Environmental Record Editing and Printing Program (EREP) User's Guide and Reference

Release 3.3

Systems

IBM

Environmental Record Editing and Printing Program (EREP) User's Guide and Reference

Release 3.3

Systems

Publication Number
GC28-1378-2

File Number
S370-37

Program Number
5654-260
5656-260
5658-260

Third Edition (July 1987)

This is a major revision of GC28-1378-1, which is now obsolete.

This edition, in support of EREP Version 3 Release 3 ***and all previous releases of EREP***, applies to the following IBM System/370 system control programs:

- DOS/VS
- VSE/Advanced Functions under VSE/System Package 1.1.0
- OS/VS2 (MVS/System Product, all releases, including MVS/XA)
- VM/370
- VM/SP
- VM/SP/HPO
- VM/XA

It applies to all subsequent releases of VSE, MVS, and VM as well, until otherwise indicated in new editions or technical newsletters.

Changes are continually made to the information herein; before using this publication in connection with the operation of IBM systems, consult the latest *IBM System/370 Bibliography*, GC20-0001, for the editions that are applicable and current.

References in this publication to IBM products, programs, or services do not imply that IBM intends to make these available in all countries in which IBM operates. A reference to an IBM program product in this publication is not intended to state or imply that only IBM's program product may be used; any functionally equivalent program may be used instead.

A form for readers' comments is provided at the back of this publication. If the form has been removed, comments may be addressed to IBM Corporation, Systems Software Products, Department A08, Building 035-3, 740 New Circle Road NE, Lexington, KY 40511. IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation whatever. You may, of course, continue to use the information you supply.

Publications are not stocked at the address given above; requests for IBM publications should be made to your IBM representative or to the IBM branch office serving your locality.

Preface

The *EREP User's Guide and Reference* applies to all versions of EREP, up to and including EREP Version 3, Release 3 and all later releases of the EREP program.

The operating systems under which EREP can run are:

- DOS/VS, DOS/VSE, and VSE/Advanced Functions — known collectively in this book as the ***VSE systems***.
- VS1, VS2, MVS/370, and MVS/XA — known collectively in this book as the ***MVS systems***.
- VM/370, VM/SP, VM/SP/HPO, and VM/XA — known collectively in this book as the ***VM systems***.

If EREP 3.3 is not installed on your system, some of the information in this book will not apply to your installation. You can find out which level of EREP your system supports by checking the release number of the EREP tape last installed; the release number is in the System Control Programming Specifications, which accompany the EREP tape. Or, for EREP 2.3 and later versions, the TOURIST output shows which release of EREP is running.

Note: New releases of EREP are always "downward compatible."

That is, the latest version of EREP will always run on your system. It will also include new functions that you can only use if you have the latest version of your operating system; but old functions, generally speaking, are not eliminated. The same is true of this book, although some very old versions of EREP (for example, IFCEREP0) are no longer supported.

Who This Book is For

This book is for the people who manage and maintain the data processing equipment in a System/370 installation:

- The system programmer, who must set up and run EREP
- The IBM customer engineer (CE), who must use the EREP reports to diagnose problems in the installation's hardware devices.
- The IBM programming service representative (PSR), who is called in when there is a problem with the running of EREP.

It is also for anyone who wants to find out what EREP is and how it works.

When reading this book, you will find a working knowledge of the operating system EREP is to run under very helpful; familiarity with its job control and entry language is also helpful, but not necessary.

Because your requirements differ according to what you are trying to do, this book contains two kinds of information:

1. Introductory and explanatory information about EREP, and detailed process information, for the person who may not know how to set up a job to run EREP. This is the user's guide.
2. Reference information in quick-look-up format — for the person who is familiar with EREP and the process of setting it up, but who wants to check out syntax or message wording or coding rules.

What and Where

The book has four parts: the user's guide information is in Part 1 and Part 2; the reference information is in Part 3 and Part 4.

Each of the four parts is made up of several chapters, which contain the following information:

- Part 1, "Planning for EREP" is the introduction to EREP.
  - Chapter 1, "Introduction to EREP" is a brief explanation of what EREP is and what it does.
  - Chapter 2, "EREP Reports" describes the EREP reports in some detail, including examples of the different types of reports. It also includes a checklist you can use when planning your EREP run.
- Part 2, "Setting Up and Running EREP" describes in detail what you have to do to run EREP under each of the S/370 operating systems.
  - Chapter 3, "Setting Up EREP" is a general introduction to the way EREP works with an operating system. It includes sections on managing your error data for EREP, on automating your EREP procedures, and on modifying an EREP job.
  - Chapter 4, "Running EREP" presents sample procedures that run EREP under each of the three system control programs, followed by descriptions of the required system controls and usage notes.
  - Chapter 5, "Correcting Coding Problems" describes the possible reasons for problems with EREP processing or output and the ways you can correct those problems.
- Part 3, "Reference Information" is the first half of the reference section of the book. It includes:
  - Chapter 6, "Introduction to EREP Controls," which explains the presentations that follow.
  - Chapter 7, "EREP Parameters," which presents the syntax and coding rules for all the EREP keyword parameters.
  - Chapter 8, "EREP Control Statements," which presents the format and coding rules for the EREP control statements.
  - Chapter 9, "CPEREP Operands - Syntax and Coding," which presents syntax and specific information about EREP controls as CPEREP operands.
  - Chapter 10, "Error Records for EREP," which introduces each type of error and operational record that EREP uses, showing its format and what it contains.

- Chapter 11, "EREP Messages," which lists all the IFC-prefixed messages as they appear in EREP output, with explanations and recommended responses.

- Chapter 12, "Summary of Tables and Charts," which contains several tables and figures to be used for quick reference. Also included are such problem determination aids as the EREP return codes, standard problem determination tables, and the DEBUG parameter.

- Part 4, "Product-Dependent Information" is the second half of the reference section of the book. It contains information specific to particular IBM machines and device types supported by EREP; information that does not apply to all users of EREP but is important to the installations that have the devices. The product-dependent information is presented by product group, as follows:

  - Chapter 13, "Card Readers and Punches"
  - Chapter 14, "Consoles and Displays"
  - Chapter 15, "Direct-Access Storage Devices (DASD)"
  - Chapter 16, "Diskette Unit"
  - Chapter 17, "Magnetic Tape Drives"
  - Chapter 18, "OCR/MICR devices"
  - Chapter 19, "Printers"
  - Chapter 20, "Processors (CPUs)"
  - Chapter 21, "Punched Tape Devices"
  - Chapter 22, "Teleprocessing (TP) Devices"
  - Chapter 23, "Other Devices"
  - Chapter 24, "System Control Programs (SCPs)"

- Finally, the book also includes a **Glossary** of terms, a list of **Abbreviations** used in the EREP reports, and a **Bibliography** of the IBM publications mentioned in the *EREP User's Guide* or associated with the use of EREP.

How To Use This Book

Here are some suggestions on how to get the most out of the information in the *EREP User's Guide and Reference*.

Before You Start to Set Up an EREP Run

- If you are a first-time user of EREP, you should read Chapter 1 and Chapter 2, to get an idea of how EREP works and what you have to do to run it, and to familiarize yourself with the EREP reports.

  Then, go on to Chapter 3 for a general introduction to running EREP under an operating system, and for information on managing error data for EREP — a very important item.

- If you are experienced as an EREP user, you may review the annotated sample EREP runs in Part 2, to see how a no-frills EREP run would be set up, and perhaps to run the sample code as is before tailoring it to your installation.

As You Set Up Your EREP Run

- Read and follow the procedures recommended for your operating system control program (SCP), and illustrated in the annotated samples in Chapter 4.
- Use the information in Chapter 5 and the messages in Chapter 11 as needed during the process of setting up and testing your EREP run.
- Use the reference information in Part 3 for quick checks on syntax or other requirements. You may also need to refer to Part 4 for information specific to one or more devices or products in your installation.

After Your EREP Run is Set Up

- Read Chapter 2 for insights into the way EREP edits the records for its reports.
- Read Chapter 3 for a general discussion of the ways to modify an EREP run or automate the EREP run.
- Use the tables and charts in Chapter 12 to refresh your memory of how the EREP parameters and controls work.
- See Chapter 10 to check on the formats and expected contents of the records used for EREP reports.

The table on the following pages gives you more specific pointers into the book according to what you need to do.

Where To Find It

WHAT YOU WANT TO DO	WHERE TO LOOK
Get a general idea about EREP	Chapter 1, "Introduction to EREP" Chapter 2, "EREP Reports"
Decide which reports to run	Chapter 2, "EREP Reports" Figure 12-3 in Chapter 12, "Summary of Tables and Charts"
Manage your error data for EREP	Chapter 3, "Setting Up EREP"
Create control statements for your hardware configuration	Chapter 8, "EREP Control Statements"
Create an EREP Run:	
General information	Chapter 3, "Setting Up EREP" on page 3-1
Under VSE:	
Sample JCS procedure	"Running EREP under VSE" on page 4-2
Required system controls (JCS)	"VSE System Controls" on page 4-14
Storage requirements	"VSE Storage Requirements" on page 4-16
Notes	"VSE Notes" on page 4-19
Under MVS:	
Sample JCL procedure	"Running EREP under MVS" on page 4-20
Required system controls (JCL)	"MVS System Controls" on page 4-32
Storage requirements	"MVS Storage Requirements" on page 4-34
Notes	"MVS Notes" on page 4-36
Under VM:	
Sample procedure	"Running EREP under VM" on page 4-38
Required System Controls	"VM System Controls" on page 4-49
Storage requirements	"Defining Files for CPEREP" on page 4-49
Notes	"VM Notes" on page 4-51

Figure 1-1 (Part 1 of 2). Where to Find Information in this Book

WHAT YOU WANT TO DO	WHERE TO LOOK
Find out what went wrong with your EREP run:	
In general	Chapter 5, "Correcting Coding Problems"
Under VSE	"VSE Notes" on page 4-19 "Information about the VSE SCP" on page 24-2
Under MVS	"MVS Notes" on page 4-36 "Information about the MVS SCP" on page 24-5
Under VM	"VM Notes" on page 4-51 "Information about the VM SCP" on page 24-8
Using EREP messages	Chapter 11, "EREP Messages"
Run EREP automatically	"Automating the Running of EREP" on page 3-8.
Review the syntax for an EREP parameter	Chapter 7, "EREP Parameters" "Syntax Rules and Conventions" on page 6-1.
Change an EREP job to get a different report	"Modifying Your EREP Run" on page 3-8.
Find out if EREP supports a device	Part 4, "Product-Dependent Information" (Look under the device type within the product group.)

Figure 1-1 (Part 2 of 2). Where to Find Information in this Book

Contents

Part 1. Planning for EREP

Part 2. Setting Up and Running EREP

Figures

Part 1. Planning for EREP

How to Use Part 1

This part of the *EREP User's Guide* tells about the EREP program — what it is, what it does, how it works.

If you are interested in learning all about EREP, you can start at the beginning and read Part 1 in its entirety.

If you only want to learn about certain aspects of EREP — what it requires of a user, for example, or what kind of output to expect — use the Section Table of Contents to locate what you want to read.

And, if you want to see a list of the questions you should ask yourself before and while you set up your EREP run, see "A Planning Checklist" on page 2-95.

Reference information is in Part 3.

Section Table of Contents

Chapter 1. Introduction to EREP

The purpose of EREP, the Environmental Record Editing and Printing program, is to help the IBM Service Representative manage and maintain your data processing installation. It is a diagnostic aid — an application program that runs under all three of the IBM System/370 operating systems.

How Does EREP Work?

EREP reads error and other environmental records generated by hardware and software, edits the records, and produces printed reports at your request. The reports show how things are — with the entire installation, with an I/O subsystem, with an individual device. The error and environmental records are the basis for the reports.

Where Do the Records Come From?

The operating system creates records from data captured by hardware or software whenever an error or other noteworthy event occurs — for example, a read error on a direct-access device or tape volume; or a machine check on a processor; or an IPL of the operating system.

Where Do the Records Go?

Each operating system includes error recovery procedures that write the records onto the system error recording data set (ERDS).

The ERDS goes by different names in the various operating systems: in the VSE systems, its symbolic name is SYSREC; in MVS, it is SYS1.LOGREC; in VM, it is not a data set but the *error recording area*. Figure 1-1 on page 1-2 represents the ERDS as initialized by the operating system.

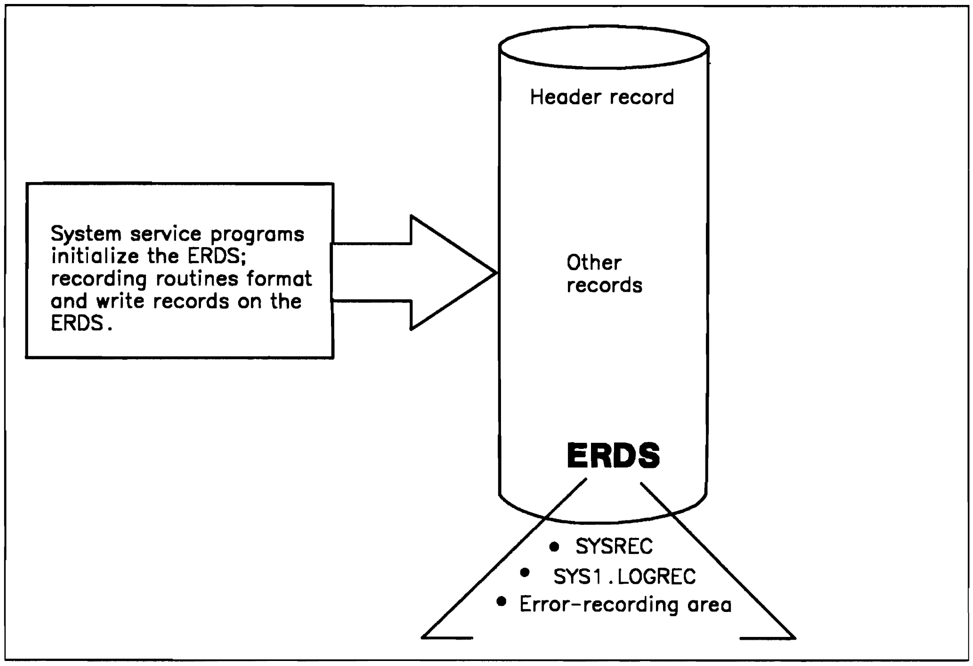


Figure 1-1. ERDS — The Error Recording Data Set, Initialized

What Does EREP Do with the Records?

When you run EREP, it reads records directly from the ERDS and processes them to produce the report you have requested.[1] EREP processing includes:

- Filtering the records through the selection parameters set up (or implied by default) for the requested report
- Sorting and formatting the records for the report
- Counting the different kinds of records used in the report
- Copying the records onto a separate output data set, if you requested it.

[1] You do not have to request a report when you run EREP; you can also use it just to move the records from the ERDS to another data set. See "The History Data Set" on page 1-6 and "Managing Error Data" on page 3-1 for more information.

How Do You Run EREP?

You run EREP by executing the program named IFCEREP1, accompanied by various parameters and controls. The parameters and controls tell EREP what you want it to do.

You execute the EREP program differently depending on which operating system you want to run it under. Following are brief summaries of how this is done; more detailed information is in Part 2, "Setting Up and Running EREP."

- Under the VSE systems, you set up a JCS job, defining input and output data sets using TLBL or DLBL statements, and the necessary logical units using ASSGN statements. You list your EREP controls as instream data to be read from the SYSIPT logical unit, and submit the job.

- Under the MVS systems, you set up a JCL job, defining input and output data sets using DD statements, and including your EREP parameters on the EXEC statement or as SYSIN instream data. Other EREP controls are SYSIN instream data.

- Under VM, you define the input and output files using FILEDEFs and then issue the CPEREP command from the CMS environment. You can enter the CPEREP and EREP operands individually or together, depending on which CMS interface you choose to use.

What Output Does the EREP Program Produce?

Each time you run EREP, you can get three things:

1. A listing of messages and other program information — the TOURIST output

2. A printed report — one of the EREP reports

3. A data set containing the records that passed filtering for the report — the history data set.

The first of these is automatic; you always get the TOURIST output. The second and third items you control with EREP parameters included when you execute the EREP program.

The TOURIST Output

While it is running, EREP puts valuable information about what it has done into a separate data set, named TOURIST.[2] The contents of the TOURIST data set are printed automatically at the end of the EREP run, unless you change the output class.

The TOURIST output shows exactly which parameters, including defaults, EREP applied to the input records to produce the report; and the number of records actually processed for the report. It also shows how EREP has interpreted any control statements you set up for the report. If the program has issued any messages during its processing, they appear in the TOURIST output. These are the same messages that are in Chapter 11, "EREP Messages," in Part 3 of this book.

You have to define the TOURIST data set to your system before running EREP. The process is described in "Defining Data Sets for EREP" on page 1-15.

An example of TOURIST output is in Chapter 12, "Summary of Tables and Charts."

The EREP Reports

The main reason to run EREP is to get one of the EREP reports. Each EREP run can produce one of several different kinds of reports to help you monitor and maintain your installation's I/O devices, controllers, channels, and processors.

Figure 1-2 on page 1-5 shows the general report types EREP produces using the records from the ERDS.

How Do You Get a Report? You request an EREP report by coding one of the *report parameters*, shown in Figure 1-5 on page 1-10. You tailor the report to contain the information you need to see by including EREP *selection parameters*, shown in Figure 1-6 on page 1-11.*:* You can use only one of the report parameters each time you execute the EREP program. However, some of the reports have several parts, so you can generate a lot of output. Unless you specifically ask for *no printed report*, EREP produces some kind of report output for each run, writing it to the EREPPT data set.

---

2 In the VSE systems, the TOURIST messages are written to the SYSLST logical unit. For the sake of simplicity, we refer to the messages as the TOURIST output and to the data set/logical unit as the TOURIST data set throughout this book.

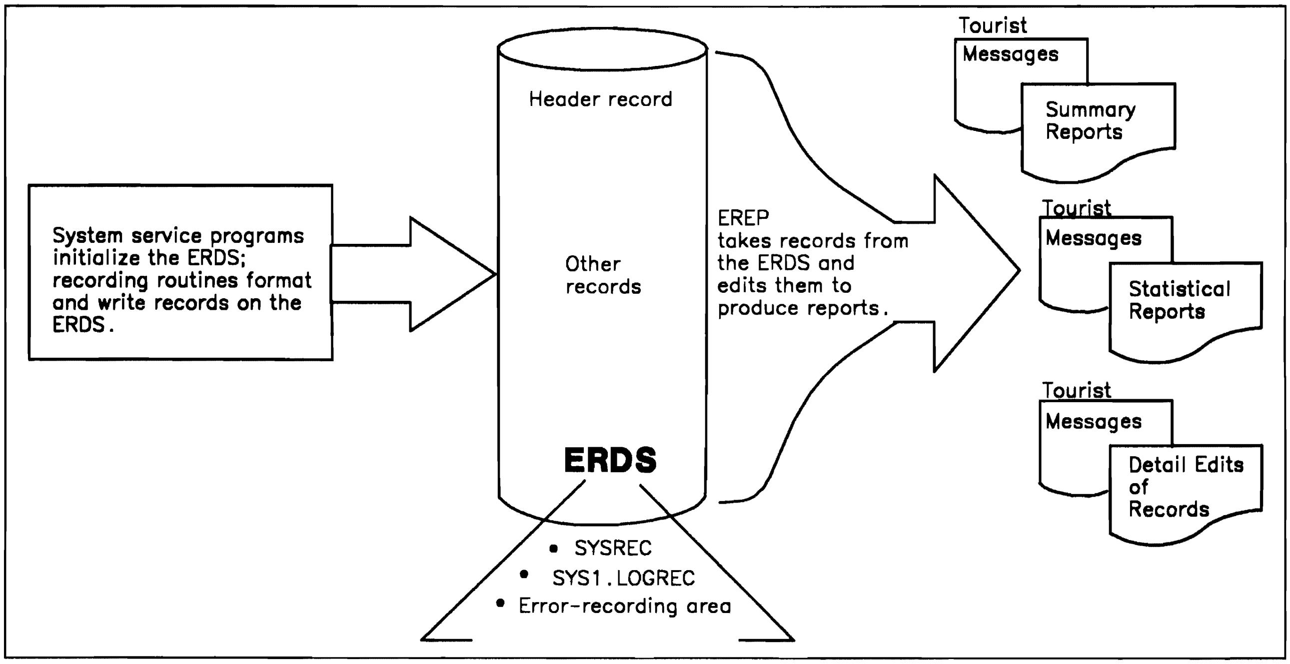


Figure 1-2. EREP Processing Produces Reports from ERDS Records

What Kinds of Reports Does EREP Produce? Most of the EREP reports summarize error-record data within various groupings. Others give you quite specific information about the error records. Listed from most general to most specific, the reports are:

- System Summary
- Trends report
- Event History
- System Exception reports, including
  - System Error Summary
  - Subsystem Exception reports for processors, channels, DASD and tape devices
  - A series of summaries of the records, analyzed according to different criteria, for DASD and tape subsystems
- Threshold Summary for some tape devices
- Detail (PRINT) Reports, including:
  - Detail Summaries of selected records
  - Data Reduction reports for specific I/O devices
  - Detail Edits of selected records

All of these report types are discussed in more detail in various places in this book; see Chapter 2, "EREP Reports" for a start.

You must define the output data set named EREPPT before running EREP to produce a report. The EREPPT data set holds the report until it is printed. See "Defining Data Sets for EREP" on page 1-15 and the sections on system controls in Part 2 for further details.

What If You Need More Than One Report? You run EREP again, requesting another report. EREP users often set up a group of jobs (or commands, if EREP is to run under VM) that execute the EREP program several times to produce a collection of reports. This group of EREP jobs is then run on a regular basis, giving a continuous picture of the performance of the installation's hardware and software systems. Part 2 of this book shows samples of such job streams.

The History Data Set

Besides requesting a report, you can also use EREP to copy (or "accumulate") the records from the ERDS to another output data set — the history data set. Figure 1-3 adds the history data set to the EREP output.

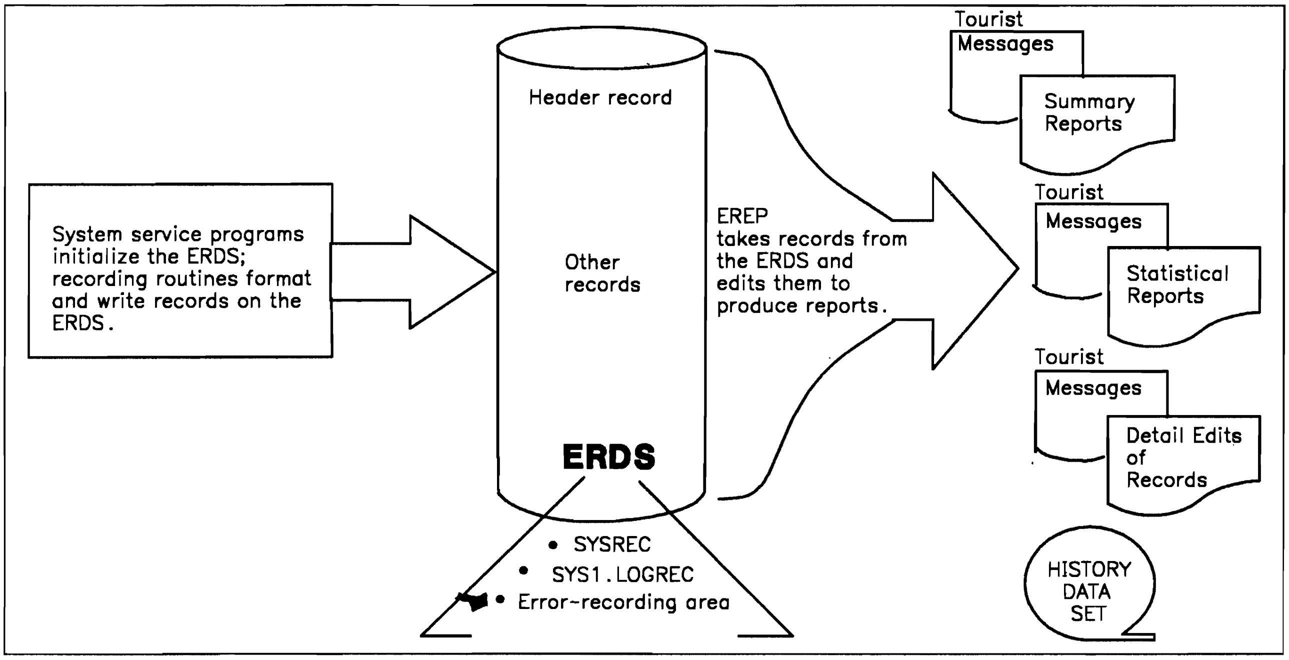


Figure 1-3. EREP Processing Produces a History Data Set

You define the output history data set as the ACCDEV output data set[3] before executing the EREP program. See "Defining Data Sets for EREP" on page 1-15.

You can have EREP accumulate the records even when it doesn't produce a report; this is the way you create a working copy of the ERDS. The process is described in "Managing Error Data" on page 3-1.

---

[3] The VSE systems use different control statements to define the input and output data sets for EREP; they are described in "VSE System Controls" on page 4-14.

Using the History Data Set as Input for EREP: After having EREP accumulate the records onto the output data set defined by the ACCDEV DD statement, you can then use the records as input for another EREP run, instead of or in addition to the ERDS. In this case, you redefine the data set in your job controls as the ACCIN input data set. It is now an input history data set. "Defining Data Sets for EREP" on page 1-15 has more information about ACCIN, and "HIST" in Chapter 7, "EREP Parameters" has more information about history input.

Other Ways You Can Use a History Data Set: You can use EREP's accumulation function to control and maintain your error data for EREP. Some of the possibilities are:

- Before running any reports at all, you can copy the records from the ERDS to another data set and clear the ERDS so it can hold new records created while the EREP reports are being run. The second data set becomes the history input for the EREP reports, and all the reports are run against the same set of input records. This is described in "Managing Error Data" on page 3-1.

- After running all the reports for a given day or week, you can copy the records from the history data set onto yet another output data set, which becomes a weekly — or monthly — history data set, holding all the records used for EREP reports for the period. If you run a Trends report against this updated history data set at regular intervals, you have a good overall picture of the frequency of errors in your installation. The sample EREP runs for your operating system in Chapter 4, "Running EREP" show this sequence of processing.

- You can merge the records from a history data set and the ERDS to get a report that uses all the records collected on the history data set plus the latest records on the ERDS. For more information, see the description of the MERGE parameter in Chapter 7, "EREP Parameters."

- If your installation runs more than one operating system, one for for each system. You can copy the records from one ERDS onto a history data set that you then merge with the ERDS from the system EREP is to run under.[4] Thus, you get EREP reports that use all the possible records for your various hardware systems, regardless of which operating system created the records.

  "Information about the MVS SCP" in Part 4 includes some suggested procedures for running EREP in a multisystem environment.

- In MVS systems, you can concatenate multiple data sets on the ACCIN DD statement, combining the records from several history data sets as input to EREP.

---

4 Note that merging history and ERDS input from different systems is not a simple task; see Chapter 24, "System Control Programs (SCPs)" in Part 4.

Figure 1-4 summarizes the possible kinds of input to EREP and output from EREP.

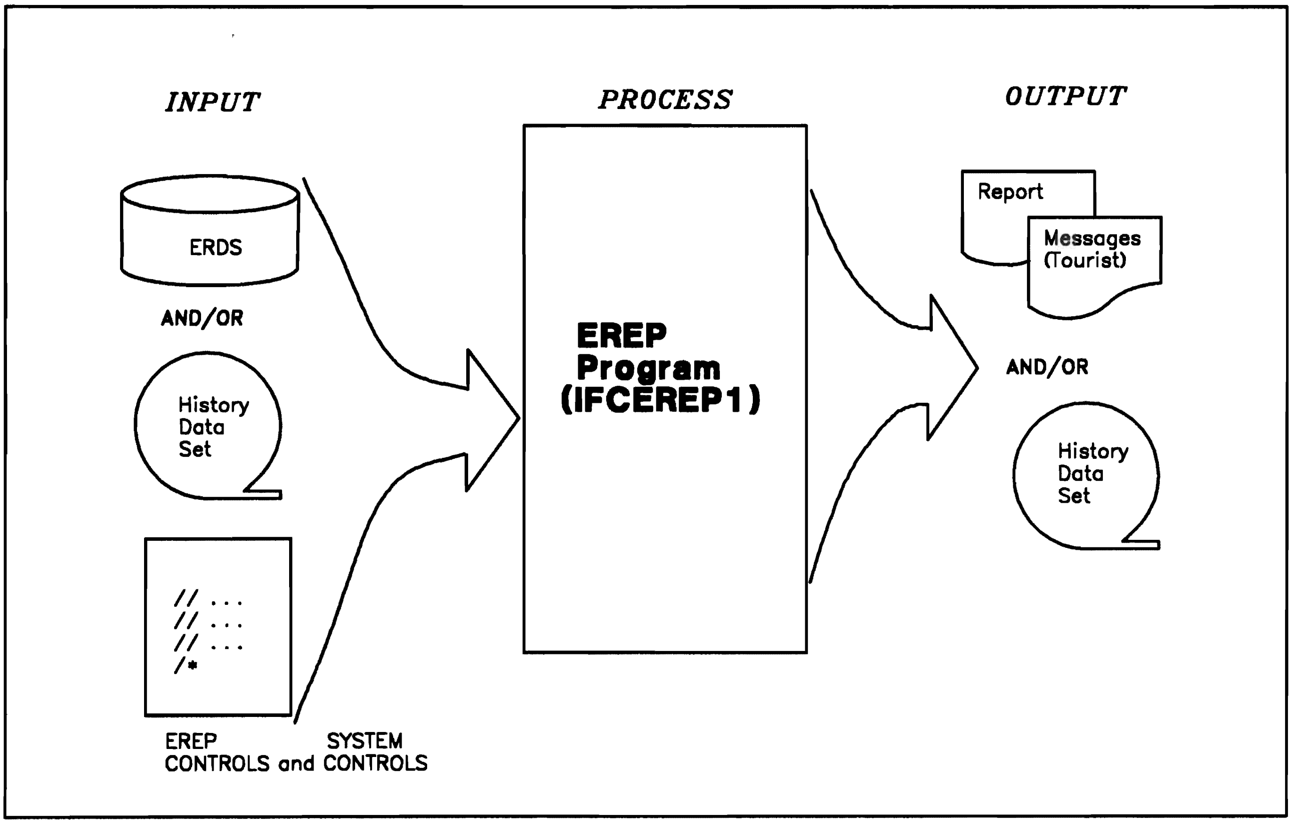


Figure 1-4. Summary of EREP's Input-Process-Output

What Are EREP Controls?

In setting up an EREP run, you are doing five things:

1. Indicating which report, if any, you want EREP to produce

2. Selecting the appropriate records for the report

3. Controlling how EREP processes the records and the report output

4. Describing your installation's configuration to EREP

5. Defining the input and output data sets and requesting the storage EREP needs from the operating system.

You use **EREP parameters** to accomplish the first two of these tasks; the third and fourth require other parameters and the **EREP control statements**. The fifth task requires **system controls**, which are different for each operating system control program (SCP).

Introducing EREP Parameters and Control Statements

The EREP controls can be grouped according to the kinds of information they convey to the EREP program.

- Some parameters tell EREP which report, if any, you want it to produce. These are the **report parameters**.

- Some parameters tell EREP which records to use for the requested report. These are the **selection parameters**.

- Some parameters control the way EREP processes the records and the report output. These are **processing parameters**.

- The EREP **control statements** convey to the program still other kinds of information — primarily about your hardware configuration.

Figure 1-5 through Figure 1-8 describe all the EREP parameters and control statements.

The actual syntax for each parameter and control statement, and more detailed usage notes, are in Part 3, "Reference Information" and Part 4, "Product-Dependent Information."

REPORT PARAMETERS	*WHAT THEY DO*
EVENT	Produces an Event History report, a chronological presentation of one-line abstracts from the selected records. It has two parts.
PRINT	Produces a series of Detail Edit and/or Summary reports for the selected record types. The number of reports depends on the input and selection parameters. *Note: PRINT is the default report parameter. The only way to run EREP without producing any report output is to code PRINT=NO.*
SYSEXN	Produces a System Exception report series, reports covering processors, channels, DASD and tape subsystems.
SYSUM	Produces a System Summary, a condensed two-part report of all errors for the principal system elements - CPU, channels, storage, SCP, I/O Subsystem.
THRESHOLD	Produces a summary of a tape subsystem, including media statistics and permanent errors that exceed the limits set on the parameter itself.
TRENDS	Produces an expanded version of the System Summary, presenting error records logged by or for the various system elements during 30 days, at most. Like the System Summary, it has two parts.

Figure 1-5. Report Parameters for EREP

EREP Report Parameters

The parameters in Figure 1-5 are the ones you use to request reports from EREP. You execute the EREP program once for every report you want, specifying the report parameter and other parameters in the job controls you use to submit the EREP program. For example:

```
//ERRUN   EXEC   PGM=IFCEREP1,PARM='EVENT,DATE=(86302),ACC=N'
```

This MVS JCL EXEC statement, if combined with the other required system controls, would produce an Event History report of all the error records logged on October 29, 1986.

DATE is one of the EREP selection parameters; see Figure 1-6 on page 1-11 and Chapter 7, "EREP Parameters."

ACC is one of the EREP processing parameters shown in Figure 1-7 on page 1-12 In this example, it indicates that the records are not to be copied to another data set.

Figure 7-1 on page 7-4 shows the selection parameters you can use for each of the EREP report parameters.

SELECTION PARAMETERS	*WHAT THEY DO*
CPU	(Processor serial and machine type numbers) Tells EREP to use only the records associated with this particular processor.
CPUCUA	(Processor serial number and device address) Tells EREP to use only the records associated with this device attached to this processor.
CUA	(Device address or number) Tells EREP to use only the records associated with this particular device address or device number.
DATE	Tells EREP to use only the records created during this date range.
DEV	(Device type) Tells EREP to use only the records associated with this particular device type; or, conversely, not to use the records associated with this device type.
DEVSER	(Device serial number) Tells EREP to use only the 34XX tape records associated with this tape device serial number.
ERRORID	(Error identifier) Tells EREP to use only the MVS software records containing this particular error identifier.
LIA/LIBADR	(Line interface [base] address) Tells EREP to use only the 3705 or 3725 communication controller records containing this line interface address.
MOD	(Processor model) Tells EREP to use only the records containing this processor machine type (number).
MODE	(370 or 370-XA) Tells EREP to use only the records created in this operating mode.
SYMCDE	(Fault symptom code) Tells EREP to use only the 33XX DASD records containing this particular fault symptom code.
TERMN	(Terminal name) Tells EREP to use only the VTAM or TCAM OBR records containing this terminal name.
TIME	Tells EREP to use only the records created during this time range.
TYPE	(Record type) Tells EREP to use only the records of the specified type(s).
VOLID	(Volume serial number) Tells EREP to use only the 33XX DASD or 34XX tape records containing this volume serial number.

Figure 1-6. Selection Parameters for EREP

EREP Selection Parameters

The parameters in Figure 1-6 are the ones you use to select the records EREP is to use in a requested report. Some of the parameters are mutually exclusive, as shown in Figure 12-1 on page 12-2. The correct syntax and other details about these parameters are in Chapter 7, "EREP Parameters."

PROCESSING PARAMETERS	*WHAT THEY DO*
ACC	(Accumulate) Tells EREP to copy the records used for the report onto an output history data set.
HIST	(History) Tells EREP that its input consists of records on a history data set.
LINECT	(Line count) Tells EREP that each page of the report output is to contain this number of lines.
MERGE	(Merge) Tells EREP that its input consists of records from both the ERDS and a history data set.
SHORT	(Short OBR) Tells EREP to print out short-form OBR records in Detail Edit report output.
TABSIZE	(Table size) Tells EREP that the sort table it uses for internal processing is to be this big.
ZERO	(Zero ERDS) When this report is complete, EREP is to clear the error-recording data set (SYSREC, or SYS1.LOGREC, or the error recording area).

Figure 1-7. Processing Parameters for EREP

EREP Processing Parameters

The parameters in Figure 1-7 are the ones you use to control the way EREP processes the records you have selected. Their functions differ widely; they are described in the individual topics in Chapter 7, "EREP Parameters."

CONTROL STATEMENTS	*WHAT THEY DO*
CONTROLLER	Tells EREP to combine the error records associated with this particular control unit and its attached devices.
DASDID	Tells EREP that this is the configuration of the 33XX DASDs within each subsystem; identifies those that do not provide physical IDs for the System Exception report series. This control statement applies only to the System Exception Report series.
ENDPARM	Tells EREP that this is the end of the instream EREP parameters; the instream data that follows consists of EREP control statements.
LIMIT	Tells EREP not to produce any output for the System Exception reports when the number of errors is below the values specified here. This control statement applies only to the System Exception Report series.
SHARE	Tells EREP to combine the records for these devices that are shared between systems. This control statement applies to all the reports that generate I/O device summaries.

Figure 1-8. Control Statements for EREP

EREP Control Statements

The statements shown in Figure 1-8 are control statements for the EREP program. They give EREP more permanent information about your configuration, and set overall criteria for the way EREP is to create a report. They are described in detail in Chapter 8, "EREP Control Statements."

Introducing System Controls

When you run EREP, you must ask the operating system to allocate storage and space on devices to hold EREP's input and output data and work spaces. You must also define the data sets and space required by the operating system for its own purposes. And you must present EREP to the operating system as a job to be executed. System controls do these things for you.

Each of the three operating systems uses different system controls for EREP: each has different names for the data sets it requires, and each has a different way for you to execute a job. See the sections headed "System Controls" in Part 2, "Setting Up and Running EREP" for detailed information about the different system job controls. What follows here is a general explanation of the differences in the way EREP works with the VSE, MVS, and VM systems.

When you run EREP to request a report, you have to submit a job to the operating system you want EREP to run under. If it is a VSE or MVS system, you supply JCS or JCL statements defining the necessary data sets, units and/or logical units and directing the system to execute the program named IFCEREP1. Having created the job or series of job steps, you submit it to the system as a batch job or interactively via TSO.

If you want to run EREP under VM, you enter the CPEREP command from the CMS environment. You have the option of being prompted for each of the CPEREP operands (EREP parameters and control statements) you want to apply to this execution of EREP, or of stacking the operands for the system to read, or putting the operands into a file named on the CPEREP command.

How EREP Runs Under an Operating System

Once EREP is executing, it works the same way regardless of the operating system it is running under:

- It obtains virtual storage from the system, in which it builds various tables needed to sort the records for the report.
- It then reads each record from the input data set(s), selecting for the report those that meet the criteria set up by record selection parameters.
- It sorts the selected records according to the way the report is to present them, and builds a line of the report in the EREPPT (or SYSLST) data set.
- When all the selected records are processed and the report is complete, it prints the EREPPT and TOURIST data sets.
- Finally, it frees the virtual storage it used for its sort tables, and returns control to the system.

The storage requirements for EREP vary widely, depending on the kind of report you request. You can adjust the amount of storage available for EREP, within system limits. The process is described in the sections headed "Storage Requirements" in Part 2, "Setting Up and Running EREP."

Defining Data Sets for EREP

Before you execute the EREP program, you must define two unique data sets for it to use:

1. The TOURIST data set (in VSE, the SYSLST logical unit), to which EREP writes messages and other processing information

2. The EREPPT data set (again, the SYSLST logical unit), in which EREP builds the report you are requesting.

In addition to these two, you must also define the other input and output data sets, depending on where the input records are and whether or not you want them copied to an output data set after processing.

Figure 1-9 shows all the system controls required for EREP by each operating system. See the sections headed "System Controls" in Part 2, "Setting Up and Running EREP" for more detailed descriptions.

INPUT AND OUTPUT FOR EREP	*VSE JOB CONTROLS (JCS)*	*MVS JOB CONTROLS (JCL)*	*VM CONTROLS*
The ERDS	SYSREC[1]	SERLOG DD statement; DSN=SYS1.LOGREC	SERLOG[2] FILEDEF
History Input	T/DLBL HISTINT/D; ASSGN SYS008	ACCIN DD statement	ACCIN[2] FILEDEF
Work Data Set for History Input	DLBL IJSYS01, EXTENT SYS001 . . . ; ASSGN SYS001	DIRECTWK DD statement	DIRECTWK[2] FILEDEF
History Output	T/DLBL HISTOT/D; ASSGN SYS009	ACCDEV DD statement	ACCDEV[2] FILEDEF
EREP Messages	SYSLST[1]	TOURIST DD statement	TOURIST[2] FILEDEF
EREP Reports	SYSLST[1]	EREPPT DD statement	EREPPT[2] FILEDEF
EREP Controls	SYSIPT[1]	SYSIN DD statement (or EXEC statement)	CPEREP operands

Figure 1-9. System Controls for EREP, by System Control Program

Notes:

1. *These logical units should already have been assigned at system/partition initialization.*

2. *CPEREP issues the FILEDEF commands for these files; you may override some of them.*

Chapter 2. EREP Reports

The EREP reports are designed to give you a variety of views of the data being processed. EREP produces:

- Overview reports, from which you can determine *if* there are problems
- Analysis reports, from which you can determine *where* there are problems
- Detail reports, from which you can determine *what* the problems are.

In order to monitor your installation — and, thus, catch problems early — you need to see certain EREP reports regularly. The sample EREP runs in Part 2, "Setting Up and Running EREP" are set up to accomplish this.

In order to decide which report to run at which time, you need to understand what each one is telling you. The following text describes the different kinds of EREP reports individually, to give you an idea of what they show and help you decide which reports to include in your EREP runs. It presents the reports in the order of most general to most specific.

IMPORTANT NOTE

The exact format of the reports depends upon the hardware devices installed and the version of EREP which you are running. However, the general examples should help as you plan your EREP run. For more detailed information about the various parts of the reports, see the maintenance documentation for the product involved.

System Summary

The System Summary report, produced when you code the SYSUM report parameter, is an overview of errors for each of your installation's principal parts, or subsystems: processors, channels, subchannels, storage, operating system control programs (SCPs), and I/O subsystems.

The report has two parts, the first part summarizing errors from all but the I/O subsystem, and the second summarizing errors recorded in the I/O subsystem.

Useful parameters for customizing your System Summary report are:

DATE
MODE
TIME

Care should be taken when specifying report parameters other than these as report results could be misleading. See Figure 12-1 on page 12-2 in Part 3 for any restrictions on using the parameters in combination.

System Summary Part 1

The first part of the System Summary report varies according to the mode of the records it summarizes. For records logged when a processor is running in 370 mode, you see counts of machine checks (MCH records) and channel checks (CCH records) by channel; for records recorded in 370-XA mode, in addition to the machine-check totals, you see counts of subchannel logouts (SLH records) by channel path ID, and channel report words (CRW records) created by both hardware and software.

The rest of the first part of the System Summary shows counts of software events that may or may not be associated with errors: IPLs and system termination. For MVS only, it also includes actual software error records.

The record counts are listed by processor (CPU); each processor is identified by a letter. See "How EREP Assigns Letters to CPUs" on page 2-92 for an explanation of the way these letter identifiers are assigned. The System Summary can only report on 10 processors; if your installation has more than 10, EREP produces the report using records from the first 9 processors it encounters. It also issues a message (IFC134I; see Chapter 11) explaining what is going on.

System Summary Part 2

The second part of the System Summary is a condensed report of every permanent and temporary error recorded for the I/O devices in your installation, listed under the processor associated with the error. If your processors share I/O devices, you must use SHARE control statements for the System Summary if you want to see I/O errors combined for all the possible paths to a device that is common to different systems. See "SHARE Control Statement."

Permanent and Temporary I/O Errors: In the System Summary, as well as in several other reports, EREP divides I/O errors into permanent errors and temporary errors. *A temporary error* is a read or write operation that failed, was retried, and eventually succeeded. *A permanent error* is a read or write operation that failed and was retried several times without success. See "Tape Subsystem Exception Report" on page 2-47 for more information about permanent errors.

The temporary errors that appear in Part 2 of the System Summary are totals of temporary read/write errors and statistical data.

Part 2 of the System Summary puts the temporary and permanent I/O errors in product or device groups. Following is a list of the product groups in the order they appear in Part 2 of the System Summary.

1. Console and unit record devices:
   a. Operator's console
   b. Card reader
   c. Card punch
   d. Printer
   e. OCR/MICR
2. Direct-access storage devices:
   a. Disk
   b. Drum/fixed-head file
   c. Mass storage system
3. Tape devices
4. Displays (channel-attached)
5. Teleprocessing (TP) communications controllers
6. Terminals
7. Other devices:
   a. Channel-to-channel adapter
   b. Cryptographic unit
   c. Dynamic pathing availability (DPA)
8. Unknown devices:
   a. Unrecognized device types

The System Summary presents the errors by control unit or device address for each device type. The device address can be the CUA, for 370-mode records, or the device number, for 370-XA-mode records. If the report includes both 370- and 370-XA-mode records, the errors are combined. If your installation includes products that have physical IDs, the errors are listed according to the entire physical ID. See "DASDID Control Statement" for the explanation of DASD physical identifiers.

EREP summarizes the I/O error data by the control unit/device address or number of the device reporting each error.

Figure 2-1, Parts 1 and 2, is an example of the System Summary, Parts 1 and 2. Following are the MVS JCL statements that produced the report.

```
//SYSUM EXEC PGM=IFCEREP1,PARM=(SYSUM,
//             'MODE=ALL','ACC=N',HIST)
//TOURIST  DD   SYSOUT=R
//EREPPT   DD   SYSOUT=R
//DIRECTWK DD   UNIT=SYSDA,SPACE=(CYL,5,,CONTIG)
//ACCIN    DD   DSN=EREP.WEEK1.HISTORY,DISP=OLD
//SYSIN    DD   DSN=D58ELM.SHARE.STMTS,DISP=OLD
[SHARE statements for appropriate I/O devices]
/*
```

Specific fields and abbreviated headings are explained in the numbered notes following each part of the figure. ***For more detailed information about hardware products included in the report, see the maintenance documentation for the product involved.***

```
S Y S T E M   S U M M A R Y                    REPORT DATE 025 84
         (PART 1)                              PERIOD FROM 329 83
 CPU/CHANNEL/STORAGE/SCP                              TO   334 83


                     TOTAL CPU-A

IPL                      0     0

MACHINE CHECK

  RECOVERABLE            0     0
  NON-RECOVERABLE        0     0

CHANNEL CHECK

  CHANNEL 0              0     0
  CHANNEL 1              0     0
  CHANNEL 2              0     0
  CHANNEL 3              0     0
  CHANNEL 4              0     0
  CHANNEL 5              0     0
  CHANNEL 6              0     0
  CHANNEL 7              0     0
  CHANNEL 8              0     0
  CHANNEL 9              0     0
  CHANNEL A              0     0
  CHANNEL B              0     0
  CHANNEL C              0     0
  CHANNEL D              0     0
  CHANNEL E              0     0
  CHANNEL F              0     0
1

PROGRAM ERROR

  PRGM INTF              0     0
  ABEND                 13    13
  RESTART                0     0

END OF DAY               0     0

TOTAL RECORDS           13    13


CPU   MODEL    SERIAL NO.

 A     3081    220864
```

Figure 2-1 (Part 1 of 2). System Summary

```
S Y S T E M   S U M M A R Y            REPORT DATE 025 84
        (PART 2)                       PERIOD FROM 329 83
      I/O SUBSYSTEM                           TO   334 83


                 TOTAL        CPU-A
              PERM  TEMP   PERM TEMP
DASD STRINGS ************************

3350   25X       0     2      0    2
3350   26X       0    22      0   22
3350   35X       0     2      0    2
3350   36X       0    40      0   40
3350   5AX       0     1      0    1
3350   71X       0     1      0    1
3350   76X       0    43      0   43
3350   77X       0     1      0    1
   B6-XX-XX      0     2      -    -
                 0    65      -    -
3400   4DX       0    24      0   24
3400   F7X       0    81      0   81
3400   FFX       0     4      0    4


TP CNTRL     ************************

3705   015
   LINES         8  1324      8 1324
3705   01C
   LINES         1    37      1   37
3705   0BE
   LINES         0   155      0  155
3705   603
   LINES         4   155      4  155


TOTALS          13  1959     13 1957

CPU   MODEL   SERIAL NO.

 A    3081    220864
```

Figure 2-1 (Part 2 of 2). System Summary

Notes:

1. **CHANNEL CHECK.***If there are 32 channels, then the channel check summary will display channels X'10' through X'1F'* ***only*** *if there is any activity on those channels.*

Trends Report

The Trends report, which you request using the TRENDS report parameter, is similar to the System Summary. The main difference between the two is that, in the Trends report, the error data is presented by the Julian day, in chronological order. Part 1 presents IPL, MCH, CCH/SLH/CRW, and Program Error (software) records for each processor. Part 2 shows permanent and temporary I/O errors for the same product/device groupings listed on page 2-3. Within product groups, the errors are presented by device address or number or physical ID within generic device or product types. The processor (CPU) associated with the record[1] appears on the line with the device address/number.

As in the System Summary, you must provide SHARE control statements if you want EREP to combine all the errors reported by a single I/O device that is common to different systems. Without SHARE statements, the Trends report shows separate entries for the device — one for each processor it is connected to. See "SHARE Control Statement" for details.

Useful parameters for customizing your Trends report are:

CUA
DATE
DEV
MODE
TIME
TYPE

Care should be taken when specifying report parameters other than these as report results could be misleading. See Figure 12-1 on page 12-2 in Part 3 for any restrictions on using the parameters in combination.

In addition, see "DEV" on page 7-11 for some restrictions on the record types you can select.

For the Trends report, you can use the DATE parameter to see records other than those created in the last 30 days, the default. However, you may not select more than 30 consecutive days of records.

1 Except in the case of devices providing physical IDs; they are associated not with a CPU, but with the control unit.

Figure 2-2, Parts 1 and 2, is an example of the Trends report, Parts 1 and 2. Following is the MVS JCL that produced the report.

```
//TRENDS EXEC PGM=IFCEREP1,PARM='CARD'
//TOURIST  DD   SYSOUT=R
//EREPPT   DD   SYSOUT=R
//DIRECTWK DD   UNIT=SYSDA,SPACE=(CYL,5,,CONTIG)
//ACCIN    DD   DSN=D24RBH1.XAFULL.HISTORY,DISP=OLD
//         DD   DSN=R24RBH1.SP211FT.FULL.HISTORY,DISP=OLD
//SYSIN    DD   *
TRENDS
HIST
ACC=N
TABSIZE=200K
DATE=(82085-82114)
CPU=(020022.3081,220022.3081,020015.3081,220015.3081)
ENDPARM
[SHARE statements for appropriate I/O devices]
/*
```

Specific fields and abbreviated headings are explained in the numbered notes following each part of the figure. ***For more detailed information about hardware products included in the report, see the maintenance documentation for the product involved.***

```
T R E N D S   R E P O R T                REPORT DATE 014 84
        (PART 1)                         PERIOD FROM 088 82
 CPU/CHANNEL/STORAGE/SCP                        TO   114 82
```

1 2

JULIAN 82 DAY	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111	112	113	114
IPL																														
CPU A	0	0	0	1	2	1	1	1	0	0	3	2	4	2	0	0	0	2	2	1	3	2	2	0	2	3	0	0	0	1
CPU B	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CPU C	0	0	0	2	2	4	1	1	1	0	2	0	2	1	0	0	0	1	2	3	1	2	0	1	1	1	1	1	2	1
CPU D	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
MACHINE CHECK																														
CPU A	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
CPU B	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CPU C	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
CPU D	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CHANNEL CHECK																														
CPU A	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3 CPU B	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CPU C	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CPU D	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
PROGRAM ERROR																														
CPU A	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CPU B	0	0	0	99	298	123	120	62	0	0	77	10	51	748	0	0	0	6	286	79	56	16	4	1	85	108	119	70	51	10
CPU C	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CPU D	0	0	0	161	192	155	999	51	0	0	73	12	44	103	0	0	0	13	371	51	29	12	9	3	91	98	151	67	38	9

CPU	MODEL	SERIAL NO.
A	3081XA	220015
B	3081	220015
C	3081XA	020015
D	3081	020015

4

```
T R E N D S   R E P O R T                REPORT DATE 014 84
        (PART 1 CONTINUED)               PERIOD FROM 088 82
 SUBCHANNEL/CHANNEL                             TO   114 82
```

5

JULIAN 82 DAY	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111	112	113	114

SUBCHANNEL

CPU A

NO ERRORS FOR THIS CPU

CPU B

NO ERRORS FOR THIS CPU

CPU C

	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111	112	113	114
CHPID 00	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	0	0	0	0	0	0	0	0	0
CHPID 05	0	0	0	0	2	1	0	0	0	0	4	0	12	4	0	0	0	2	4	3	0	1	0	0	4	2	0	0	0	4
CHPID 10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	12	0	0	0	0	0	0	0	0	0
CHPID 11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	8	0	0	0	0	0	0	0	0	0	0
CHPID 16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	0	0	0	0	0	0	0	0	0
CHPID 21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	[illegible]	0	[illegible]	0	0	0	0	0	0	0
CHPID 24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	[illegible]	27	0	[illegible]	0	0	0	0	0	0	0

CPU D

	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111	112	113	114
CHPID 15	0	0	0	26	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CHANNEL REPORT WORD

	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111	112	113	114
CPU A																														
HARDWARE	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SOFTWARE	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CPU B																														
HARDWARE	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SOFTWARE	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CPU C																														
HARDWARE	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SOFTWARE	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CPU D																														
HARDWARE	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SOFTWARE	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CPU	MODEL	SERIAL NO.
A	3081XA	220015
B	3081	220015
C	3081XA	020015
D	3081	020015

4

Figure 2-2 (Part 1 of 2). Trends Report

T R E N D S R E P O R T (PART 2) I/O SUBSYSTEM

REPORT DATE 014 84
PERIOD FROM 088 82 TO 114 82

JULIAN 82 DAY	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111	112	113	114
DASD 6						7																								
2305																														
1D0 A																														
PERM	0	0	0	3	1	2	1	0	0	0	3	0	13	3	0	0	0	3	2	0	1	0	0	0	0	2	1	0	3	0
TEMP	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1D0 C																														
PERM	0	0	0	2	3	3	4	1	0	0	5	2	8	1	0	0	0	0	4	0	2	0	0	0	6	0	0	0	1	0
TEMP	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1E0 A																														
PERM	0	0	0	1	4	1	0	1	0	0	9	0	18	1	0	0	0	1	5	1	0	0	0	0	0	0	1	1	0	0
TEMP	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1E0 C																														
PERM	0	0	0	2	2	2	2	0	0	0	8	0	4	1	0	0	0	0	3	1	0	0	0	0	2	3	2	2	4	0
TEMP	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1E8 A																														
PERM	0	0	0	2	3	3	4	0	0	0	8	2	14	2	0	0	0	0	5	1	1	0	0	0	1	4	0	1	0	0
TEMP	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7F8 C																														
PERM	0	0	0	1	0	0	0	0	0	0	2	0	0	1	0	0	0	0	2	0	0	0	0	0	0	2	2	0	1	0
TEMP	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8 3330																														
921 B																														
PERM	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
TEMP	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	0	0	0	0	0	0	0	0	0	0	0
922 A																														
PERM	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
TEMP	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
TAPE																														
3400																														
570 A																														
PERM	0	0	0	1	0	2	1	3	0	0	0	0	8	2	0	0	0	2	1	6	26	2	0	0	0	0	0	0	2	2
TEMP 9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
570 B																														
PERM	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
TEMP	0	0	0	0	1	1	0	14	0	0	0	0	0	255	0	0	0	2	0	0	62	0	0	0	0	0	0	0	2	0
570 C																														
PERM	0	0	0	0	0	2	1	3	0	0	2	0	7	1	0	0	0	0	1	4	0	4	0	0	1	4	0	0	0	1
TEMP	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
570 D																														
PERM	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
TEMP	0	0	0	5	0	0	0	3	0	0	0	0	9	0	0	0	0	147	1	46	0	35	0	0	0	4	0	0	0	0
571 A																														
PERM	0	0	0	0	0	2	0	0	0	0	4	0	5	7	0	0	0	1	3	1	0	1	0	0	1	0	1	0	3	1
TEMP	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
DISPLAY																														
3284																														
3E2 C																														
PERM	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
TEMP	0	0	0	0	2	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	1	2
3277																														
3E3 A																														
PERM	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
TEMP	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0	0	0	0	1	0	1	2
3E3 C																														
PERM	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
TEMP	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	0	0	0	0
OTHER																														
DPA																														
000 C																														
PERM	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
TEMP	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
100 C																														
PERM	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
TEMP	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
300 C																														
PERM	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
TEMP	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CPU	MODEL	SERIAL NO.
A	3081XA	220015 4
B	3081	220015
C	3081XA	020015
D	3081	020015

Figure 2-2 (Part 2 of 2). Trends Report

Notes:

1. *System error types, by processor (CPU).*
2. *Each column contains error counts for one day; unless you specify a shorter date range, the report covers 30 days.*
3. *For CCH (and SLH) records, only those channels (channel paths) with errors appear in the report.*
4. *Processors, identified from filtered data and/or share statements. XA indicates that the processor was running in 370-XA mode.*

5. *This section of the report appears only if 370-XA mode records are present.*

6. *Device group.*

7. *Each column represents one day. A field of all 9s indicates that this number was larger than the print positions allowed.*

8. *Device type.*

9. *CPU-CUA (or device number) path.*

Event History Report

The Event History report consists of one-line abstracts of selected information from each record, listed in chronological order. You request the Event History using the EVENT report parameter. Its primary value lies in showing you errors and events in context, letting you see the frequency, order, and pattern of their occurrence.

Because it is so highly condensed, however, the Event History can be difficult to read. To make things easier, the first part of the report output is a template showing the headings used for the record-dependent data from each type of record. The second part of the output is the Event History itself. The third part is a summary, by processor (CPU) identifier, of all the records presented in the report, with totals for each record type. See "How EREP Assigns Letters to CPUs" on page 2-92 for the explanation of the letter identifiers.

Useful parameters for customizing your Event History report are:

CPU
CUA
DATE
DEV
MODE
TERMN
TIME
TYPE
VOLID

Care should be taken when specifying report parameters other than these as report results could be misleading. See Figure 12-1 on page 12-2 in Part 3 for any restrictions on using the parameters in combination.

Note: Control statements (CONTROLLER and SHARE) are not appropriate for the Event History, because it is a **chronological** *presentation of errors.*

Figure 2-3 and Figure 2-4 show the heading templates for 370 and 370-XA mode records, respectively. Figure 2-5 is an example of the Event History report. Figure 2-6 is an example of the Event History summary. Following are the MVS JCL statements that produced the report.

```
//EVENT EXEC PGM=IFCEREP1,PARM=(EVENT,'DATE=(82124)',
//              'MODE=ALL','ACC=N',HIST)
//TOURIST  DD   SYSOUT=R
//EREPPT   DD   SYSOUT=R
//DIRECTWK DD   UNIT=SYSDA,SPACE=(CYL,5,,CONTIG)
//ACCIN    DD   DSN=EREP.WEEK1.HISTORY,DISP=OLD
//SYSIN    DD   DUMMY
/*
```

Note the absence of SHARE statements; the Event History does not use them.

Specific fields and abbreviated headings are explained in the numbered notes following the figures. ***For more detailed information about the various parts of the reports, see the maintenance documentation for the product involved.***

```
                                       EVENT HISTORY TEMPLATE (S/370)

FOR RECORD TYPES:   RECORD DEPENDENT DATA

MCH:                                  PSW-MCH /PROG-EC                                   ERROR-ID

CCH:              CUA DEVT        CSW

OBR:              CUA DEVT    CMD CSW

MDR:              CUA DEVT                                                             VOLUME/TERM NAME

MIH:              CUA DEVT                              CSID                           VOLUME/TERM NAME

DDR:              CUA DEVT                                                             VOLUME/TERM NAME

OBR-DMT, OBR-EOD: CUA DEVT                                                             VOLUME/TERM NAME

OBR-PRM, OBR-TMP: CUA DEVT    CMD CSW  SENSE   04  06  08  10  12  14  16  18  20  22 VOLUME SEEK SD CT

OBR-DPA:          CUA DEVT    SCUA                        SNID

SFT:                          REASON   PSW-MCH /PROG-EC  RCYRYXIT   COMP/MOD  CSECTID  ERROR-ID

IPL:                  SSYS ID REASON

MDR-DAS:          CUA DEVT             SENSE   04  06  08  10  12  14  16  18  20  22 VOLUME SEEK SD CT


EOD: * ONLY COMMON PREFIX DATA FOR THIS RECORD TYPE.

COMMON PREFIX:  (FOR ALL RECORD TYPES)
   TIME     JOBNAME  RECTYP CP
```

Figure 2-3. Event History Template for 370 Mode Records

Note: See the list of abbreviations following the Glossary for explanations of the abbreviations used in the template.

```
                                        EVENT HISTORY TEMPLATE (S/370XA)

FOR RECORD TYPES:   RECORD DEPENDENT DATA

MCH:                                      PSW-MCH /PROG-EC                                ERROR-ID

SLH:               DNO DEVT CHP     SCSW                    ESW

CRW:               DNO CRW

OBR:               DNO DEVT    CMD SCSW

MDR:               DNO DEVT CHP                                                          VOLUME/TERM NAME

MIH:               DNO DEVT CHP REASON                      CSID                         VOLUME/TERM NAME

DDR:               DNO DEVT                                                              VOLUME/TERM NAME

OBR-DMT, OBR-EOD:  DNO DEVT CHP                                                          VOLUME/TERM NAME

OBR-PRM, OBR-TMP:  DNO DEVT CHP CMD SCSW SENSE  04  06  08  10  12  14   16  18  20  22 VOLUME SEEK SD CT

OBR-DPA:           DNO DEVT CHP CMD SCSW SPID                 SNID

SFT:                            REASON   PSW-MCH /PROG-EC  RCYRYXIT   COMP/MOD  CSECTID  ERROR-ID

IPL:                   SSYS ID  REASON

MDR-DAS:           DNO DEVT CHP          SENSE  04  06  08  10  12  14   16  18  20  22 VOLUME SEEK SD CT


EOD: * ONLY COMMON PREFIX DATA FOR THIS RECORD TYPE.

COMMON PREFIX:  (FOR ALL RECORD TYPES)
   TIME     JOBNAME  RECTYP CP

AN ASTERISK(*) PRECEDING THE CPU LETTER INDICATES A S/370XA MODE RECORD
```

Figure 2-4. Event History Template for 370-XA Mode Records

Note: See the list of abbreviations following the Glossary for explanations of the abbreviations used in the template.

```
EVENT HISTORY (S/370 & S/370XA)                                                                           REPORT DATE  124 82
                                                                                                          PERIOD FROM  124 82
                                                                                                          PERIOD TO    124 82
                                                        SPID                      SNID
                                                                         BYTES     DATA CK  SEEKS  OVERRUN
   1                                                                    RD/SRCHD COR/RTRY  CNT/ERR COM/DATA             3
                                        SSYS ID  REASON   PSW-MCH /PROG-EC  RCYRYXIT   COMP/MOD  CSECTID  ERROR-ID
  TIME      JOBNAME  RECTYP CP CUA DEVT          CMD CSW  SENSE     04   06  08   10   12  14   16  18   20  22   VOLUME SEEK SD CT
                            *  DNO CRW   CHP         SCSW                      ESW

DATE  124 82                2

07 10 10 16 MSTRJCL  SFTPI   B               800C4000 070C000000099AD8  IEAVTMMT   IEAVTMMT  IEAVTMMT 00300000000010003F033
07 13 25 41 MSTRJCL  SFTPI   B               800C4000 070C000000099AD8  IEAVTMMT   IEAVTMMT  IEAVTMMT 00320000000010003F7D3
07 29 28 06          OBRTMP  C 0E2 3277                                                                RP07L0
07 34 28 48 D24RAC1  SFTABN  A               0013E000 070C000000C70846  ICVCME02   IGC0013I  ICVDSD03 00330000001E00042929
07 36 11 25 MSTRJCL  SFTPI   B               800C4000 070C000000099AD8  IEAVTMMT   IEAVTMMT  IEAVTMMT 0035000000010004202D
07 36 14 69 MSTRJCL  SFTPI   A               800C4000 070C000000099AD8  IEAVTMMT   IEAVTMMT  IEAVTMMT 003700000000100042D50
07 44 23 28          OBRTMP  D 0E5 3277                                                                RP07L0
07 44 23 28          OBRTMP  D 0E8 3277                                                                RP07L0
07 44 23 29          OBRTMP  D 0E6 3277                                                                RP07L0
07 44 23 29          OBRTMP  D 0E7 3277                                                                RP07L0
07 44 23 30          OBRTMP  D 0E0 3277                                                                RP07L0
07 44 23 30          OBRTMP  D 0E9 3277                                                                RP07L0
07 44 23 31          OBRTMP  D 0EA 3277                                                                RP07L0
07 44 23 32          OBRTMP  D 0EC 3277                                                                RP07L0
07 44 23 32          OBRTMP  D 0EB 3277                                                                RP07L0
07 44 23 34          OBRTMP  D 0ED 3277                                                                RP07L0
07 44 23 45          OBRTMP  C 0E4 3277                                                                RP07L0
07 45 04 56          OBRTMP  C 0E3 3277                                                                RP07L0
07 46 47 13          OBRTMP  D 0FE 3705                                                                0FE-S
07 47 23 71   N/A    MDR     D 0FE 3705
07 47 23 87   N/A    MDR     D 0FE 3705
07 48 24 74          OBR     D 6B6 3277       03 8200                                                  RP07L6
07 51 46 17          OBRTMP  C 60F 3800
07 53 37 50 D83WJM1  MIH    *A 7C5 3330       START PENDING                                            D92722
07 53 51 94 D15MBP1  OBRTMP  C 9CA 3330       04 0200 10000000 2A420630 2EB00000 00040000 00000000 00000000 D92494   66    6
07 54 08 98 *MASTER* MIH 4  *B 7C5 3330       START PENDING                                            D92722
07 54 40 46 *MASTER* MIH    *B 7C5 3330       START PENDING                                            D92722
07 54 56 69 D53PAH1  OBRTMP  C 9CF 3330       04 0200 10000000 07640030 2EB00000 00040000 00000000 00000000 IP0004  100    0
07 55 11 93 *MASTER* MIH    *B 7C5 3330       START PENDING                                            D92722
07 55 27 00 D60JDN1  OBRTMP  C 9C4 3330       04 0200 10000000 1C640030 2E300000 00040000 00000000 00000000 IPOL01  100    0
07 55 43 41 *MASTER* MIH    *B 7C5 3330       START PENDING                                            D92722
07 56 14 89 *MASTER* MIH    *B 7C5 3330       START PENDING                                            D92722
07 56 14 97 D71DJB1  OBRTMP *B 7C4 3330   12  31 0C00 00F72F48 06000005 10000000 1C640030 2E300000 00040000 IPOL01    0    0
07 56 46 37 *MASTER* MIH    *B 7C5 3330       START PENDING                                            D92722
07 56 46 43 FRANKXA1 OBRTMP *B 7CC 3330   12  04 0C00 00F31338 06000006 10000000 1C350930 2EB00000 00040000 D24PAK    0    0
07 57 17 85 *MASTER* MIH    *B 7C5 3330       START PENDING                                            D92722
07 57 17 87 FRANKXA1 OBRTMP *B 7CC 3330   12  04 0C00 00F6FB38 06000006 10000000 1C3A0F30 2EB00000 00040000 D24PAK    0    0
07 57 49 32 *MASTER* MIH    *A 7C5 3330       START PENDING                                            D92722
07 57 49 34 SMITHXA1 OBRTMP *B 7CC 3330   12  04 0C00 00F369B8 06000006 10000000 1C360330 2EB00000 00040000 D24PAK    0    0
07 57 59 22 D15KTK1  OBRTMP *B 4AD 3330   04  29 0C00 00F72560 0E00002C 10000000 550D001F 08000000 00000002 POK150    0    0  00
07 58 03 95 D24JAL1  OBRTMP  C 9C0 3330       04 0200 10000000 38000130 2E300000 00040000 00000000 00000000 CLR12A    0    1
07 58 20 80 *MASTER* MIH    *A 7C5 3330       START PENDING                                            D92722
07 58 30 47 NONE-FRR SFTABN  A               000AB000 070C100000B66B40  IKTIOFRR   IKTIOM02  IKTOMLU0 00390042001B0004617E
07 58 52 28 *MASTER* MIH    *B 7C5 3330       START PENDING                                            D92722
07 58 52 33 D15DES1  OBRTMP *B 7CA 3330   12  31 0C00 00F72D48 06000005 10000000 2A420630 2E300000 00040000 D92494    0    0
07 59 23 76 *MASTER* MIH    *B 7C5 3330       START PENDING                                            D92722
07 59 55 23 *MASTER* MIH    *A 7C5 3330       START PENDING                                            D92722
07 59 55 27 DAVEXA1  OBRTMP *B 7CC 3330   12  31 0C00 00F71448 06000005 10000000 1C330030 2E300000 00040000 D24PAK    0    0
08 00 10 97 D15DFY1  MIH    *A 7CA 3330       START PENDING                                            D92873
08 00 25 39 D15MBP1Z OBRTMP  C 9C2 3330       31 0200 10000000 2A020030 2E300000 00040000 00000000 00000000 D92689    2    0
08 00 26 71 *MASTER* MIH    *A 7C5 3330       START PENDING                                            D92722
08 00 41 16 D15MBP1Z OBRTMP  C 9CA 3330       04 0200 10000000 2A001230 2EB00000 00040000 00000000 00000000 D92494    0   18
08 00 42 45 *MASTER* MIH    *B 7CA 3330       START PENDING                                            D92873
08 00 58 19 *MASTER* MIH    *A 7C5 3330       START PENDING                                            D92722
08 01 31 23          OBR     C BBD 3277       03 8200                                                  RP07LB
08 02 32 77 MSTRJCL  SFTPI   B               800C4000 070C000000099AD8  IEAVTMMT   IEAVTMMT  IEAVTMMT 004500000000100046AF5
08 04 10 86 MSTRJCL  SFTPI   B               800C4000 070C000000099AD8  IEAVTMMT   IEAVTMMT  IEAVTMMT 004A00000000100046ECA
08 05 34 24 MSTRJCL  SFTPI   B               800C4000 070C000000099AD8  IEAVTMMT   IEAVTMMT  IEAVTMMT 004D0000000010004720B
08 05 40 95 MSTRJCL  SFTPI   A               800C4000 070C000000099AD8  IEAVTMMT   IEAVTMMT  IEAVTMMT 004F0000000010004724E
08 05 44 34 D10SCB1  SLH    *B 208 3330   02      0502441700FE4EB800020006 00807484
08 07 04 90          OBR     C BE1 3277       03 8200                                                  RP07LB
08 07 20 79 NONE-FRR SFTABN  A               000AB000 070C100000B66468  IKTIOFRR   IKTIOM02  IKTIMLU0 00500042004500047635
08 13 51 73 MSTRJCL  SFTPI   A               800C4000 070C000000099AD8  IEAVTMMT   IEAVTMMT  IEAVTMMT 0055000000010004857A
08 13 54 26 MSTRJCL  SFTPI   A               800C4000 070C000000099AD8  IEAVTMMT   IEAVTMMT  IEAVTMMT 00570000000100048594
08 14 15 12 MSTRJCL  SFTPI   A               800C4000 070C000000099AD8  IEAVTMMT   IEAVTMMT  IEAVTMMT 005A0000000100048664
08 15 29 85 MSTRJCL  SFTPI   A               800C4000 070C000000099AD8  IEAVTMMT   IEAVTMMT  IEAVTMMT 005C00000C0100048950
08 15 33 72 D58HBM1  SFTABN  A               0013E000 070C000000C70846  ICVCME02   IGC0013I  ICVDSD03 005D0000005400048976
08 15 41 37 MSTRJCL  SFTPI   B               800C4000 070C000000099AD8  IEAVTMMT   IEAVTMMT  IEAVTMMT 005F000000001000489C3
```

Figure 2-5. Event History Report

Notes:

1. *The header is a universal type. Headers for each record type are provided on the templates illustrated in Figure 2-3 and Figure 2-4.*

2. *CPU models and serial numbers associated with the letter identifiers are shown at the end of the report summary. The letter identifiers are internal to the Event History (high-order serial number assigned letter A) and should not be confused with external CPU machine identifiers.*

3. *Under SEEK is the DASD cylinder head or block number, under SD CT is the storage director/controller physical ID for DASD.*

4. *An asterisk (*) preceding the processor (CP) letter indicates a 370-XA mode record.*

Special Note: *For any products that record OBR records asynchronously, only the sense data reflect the origin of an error record. The other information in the record might reflect the recording device rather than the device that has the problems.*

RECORD TYPES	TOTAL	CPU-A	CPU-B	CPU-C	CPU-D
MCH	0	0	0	0	0
MCHTRM	0	0	0	0	0
MACHINE CHECK	0	0	0	0	0
OBR	3	0	0	2	1
OBRSHT	0	0	0	0	0
OBRDMT	0	0	0	0	0
OBREOD	0	0	0	0	0
OBRDPA	0	0	0	0	0
OBRTMP	28	0	7	10	11
OBRPRM	0	0	0	0	0
OUTBOARD	31	0	7	12	12
SFT	0	0	0	0	0
SFTABN	4	4	0	0	0
SFTMCH	0	0	0	0	0
SFTPI	13	6	7	0	0
SFTRST	0	0	0	0	0
SOFTWARE	17	10	7	0	0
IPL	0	0	0	0	0
SYSTEM INITIALIZATION	0	0	0	0	0
DDR	0	0	0	0	0
DDROPR	0	0	0	0	0
DDRSYS	0	0	0	0	0
SYSTEM RECONFIGURATION	0	0	0	0	0
EOD	0	0	0	0	0
SYSTEM TERMINATION	0	0	0	0	0
MDR	2	0	0	0	2
MDRDAS	0	0	0	0	0
BUFFER OFFLOAD	2	0	0	0	2
CCH	0	0	0	0	0
CCHINC	0	0	0	0	0
CCHCRH	0	0	0	0	0
CHANNEL CHECK	0	0	0	0	0

1

Figure 2-6 (Part 1 of 2). Event History Summary

RECORD TYPES	TOTAL	CPU-A	CPU-B	CPU-C	CPU-D	
MIH	0	0	0	0	0	
CHANNEL END	0	0	0	0	0	
DEVICE END	0	0	0	0	0	
MISSING INTERRUPT	0	0	0	0	0	
OVER ALL TOTALS	50	10	14	12	14	2

RECORD TYPES	TOTAL	CPU-A	CPU-B	CPU-C	CPU-D	
MISSING CSCH	0	0	0	0	0	
MISSING HSCH	0	0	0	0	0	
IDLE DEVICE	0	0	0	0	0	
START PENDING	17	7	10	0	0	
MOUNT PENDING	0	0	0	0	0	
MISSING PRI STATUS	0	0	0	0	0	
MISSING SEC STATUS	0	0	0	0	0	
MISSING INTERRUPT	17	7	10	0	0	3
HARDWARE	0	0	0	0	0	
SOFTWARE	0	0	0	0	0	
CHANNEL REPORTS	0	0	0	0	0	
CHPID-02	1	0	1	0	0	
SUBCHANNEL LOGOUTS	1	0	1	0	0	
OVER ALL TOTALS	68	17	25	12	14	4

CPU	MODEL	SERIAL NO. 5
A	3081	020015
B	3081XA	020015
C	3033	020863
D	3033	020862

Figure 2-6 (Part 2 of 2). Event History Summary

Notes:

1. **RECORD TYPES**. *If 370 and 370-XA mode records are used, the records common to both modes are combined. Exception: 370-mode MIH records are totaled separately.*

2. **OVER ALL TOTALS**. *These totals include only those records common to 370 and 370-XA modes and the MIH errors for 370-mode records.*

3. **MISSING INTERRUPT**. *These MIH errors are for 370-XA mode records.*

4. **OVER ALL TOTALS**. *These totals include all errors recorded in both processing modes.*

5. **CPU information**. *Processors, identified from filtered data. XA indicates that the processor was running in 370-XA mode.*

System Exception Reports

Depending on the types of processor and I/O subsystems your installation has, the EREP System Exception reports can help you identify problems within those subsystems. EREP produces System Exception reports for only some hardware subsystems, however; see the product groups in Part 4, "Product-Dependent Information" for product subsystems included in the System Exception reports.

When you code the SYSEXN report parameter for an EREP run, EREP analyzes hardware and software error records to produce a series of reports that provide information about how your installation's hardware subsystems are working.

You will want to set up additional controls for the System Exception reports, using the DASDID, LIMIT and SHARE control statements, before you actually request the report series. See the descriptions of the control statements in Chapter 8, "EREP Control Statements" and Part 4 for more information.

The System Exception report series comprises several separate reports: a two-part System Error Summary, Subsystem Exception reports for each category of subsystem, and more detailed summaries for DASD and tape subsystems.

For the Subsystem Exception reports and related summaries, EREP accumulates error data and, when available, usage statistics – as for a DASD or tape subsystem – and summarizes the information by component.

In order to avoid reworking the same errors, and to make sure that the probable failing unit analysis is accurate, you should run the System Exception report series every day. See Chapter 4, "Running EREP" for examples of EREP runs that include daily System Exception reports.

Care should be taken when specifying selection parameters other than DATE and TIME as report results could be misleading.

System Error Summary Format

The System Error Summary presents processor (CPU) errors and channel checks in Part 1, and permanent DASD and tape errors in Part 2. Part 1 also includes a summary of IPL, EOD, and restart records, if present. The data is presented in chronological order.

For each of the four sources of errors, the System Error Summary includes the *probable failing unit* (PFU). The PFU is the component (unit) on which EREP has determined that the error most likely occurred.

The printed output for Part 1 is one page for each supported CPU in the installation; Part 2 combines the I/O errors for all supported subsystems. EREP does not print out duplicates of records occurring together; see the example in Figure 2-7 on page 2-24.

Subsystem Exception Report Formats

EREP formats each of the Subsystem Exception reports according to the requirements of the hardware involved.

- The Processor Subsystem Exception report is organized around the *service level indicators* for termination, hard, and soft machine checks. For soft machine checks, the report shows the number of 60-minute intervals in which the LIMIT value was exceeded. See "LIMIT Control Statement" in Chapter 20, "Processors (CPUs)."

- The Channel Subsystem Exception report is organized according to the possible *source* of channel checks: the channel, the storage control unit, and the controller. It shows the number of times each of these error types exceeded the LIMIT values for specific channels or controllers, and includes the 60-minute exception count for each channel or controller.

- The DASD Subsystem Exception report is organized by the *probable failing units* (PFUs), starting with the units closest to the processor and working toward the volume. (A DASD subsystem consists of a storage control unit, if present, a controller or storage director, and the string(s) of devices attached to the controller.) The units are ordered, within each PFU grouping, from that with the largest number of permanent errors to that with the smallest number of temporary errors.

  A probable failing unit is identified through the physical ID of the device. The physical ID is the combined identifiers of storage controller, control unit, and device. In order for EREP to recognize units common to different systems and to arrive at the correct PFUs, you must code DASDID control statements to establish physical IDs for those DASD in your installation that do not provide their own physical IDs.

- The Tape Subsystem Exception report is organized by *exception type* — permanent errors and temporary errors that exceeded the LIMIT values; and by the suspected *source* of the error — either hardware or the volume and the drive it was created on. A tape subsystem consists of a control unit and its string(s) of drives.

DASD and Tape Summaries

For DASD and tape subsystems, EREP also produces more detailed summaries of the errors shown in the Subsystem Exception reports.

- For DASD subsystems, the additional summaries present errors within the following categories:

  - **Symptom Code**: lists the errors by fault symptom code within each PFU (probable failing unit) group.

  - **Data Transfer**: further broken down according to whether the PFU is the volume or something other than the volume.

  - **Storage Control Unit (SCU)**: groups overruns under each interface between channel or subchannel and SCU.

  In addition, EREP produces two other reports for DASD that can be useful in determining the source of a DASD problem:

  - The printout of **Informational Messages** could help you define a problem to IBM customer service personnel.

  - The **DASD String Summary** could help you determine if a problem is unique to a particular device, or is also occurring on other devices in the controller string.

- For tape subsystems, the additional summaries group errors and sources as follows:

  - **Permanent Errors**: lists them by CUA (or device number) and volume/creating drive.

  - **Temporary Errors**: lists *all* temporary errors, regardless of whether or not they exceeded the LIMIT values and appeared in the Subsystem Exception report. See "LIMIT Control Statement" in "34XX Tape Devices" in Part 4.

  - **DEVNO/CUA Statistics**: reports on each device number or CUA appearing on the Subsystem Exception report, showing the breakdown of all activity recorded against it.

  - **Volume Statistics**: lists all activity for all volumes that appeared on the Subsystem Exception report, by volume serial number (volid) in chronological order.

Figure 2-7 through Figure 2-21 are examples of the System Exception report output, in the order discussed here. ***Because the reports are hardware-specific, the output might not match what you see when you request the System Exception series for yourself.*** However, the general examples should help as you plan your EREP run. Following is the MVS JCL used to request the System Exception series.

```
//SYSEXN EXEC PGM=IFCEREP1,PARM=(SYSEXN,'DATE=(82347)',
//             ('ACC=N','TABSIZE=512K',HIST)
//TOURIST  DD   SYSOUT=R
//EREPPT   DD   SYSOUT=R
//DIRECTWK DD   UNIT=SYSDA,SPACE=(CYL,5,,CONTIG)
//ACCIN    DD   DSN=EREP.DAYLY.HISTORY,DISP=OLD
//SYSIN    DD   DSN=CNTRL.STMTS.SYSEXN,DISP=OLD
[DASDID, LIMIT, and SHARE statements for this report]
/*
```

Specific fields and abbreviated headings are explained in the numbered notes following each report example. ***For more detailed information about the various parts of the reports, see the maintenance documentation for the product involved.***

```
SYSTEM ERROR SUMMARY                                  REPORT DATE 348 82
      (PART 1)                                        PERIOD FROM 347 82
                                                             TO   347 82

MODEL 3033    SERIAL 020557   CPU A  1

IPL/RESTART/TERMINATION  2

               RECORD    TIME SINCE
     TIME       TYPE     LAST ACTIVE    REASON PROBABLE CAUSE

   DATE 347/82

  08:01:30:95  IPL       09:01:29:56    NM     NORMAL SYSTEM INITIALIZATION
  15:23:09:29  TERM                     MCH    FORCED TERMINATION
  15:26:30:76  IPL       00:02:29:56    NM     NORMAL SYSTEM INITIALIZATION
  19:15:56:22  RESTART                         RESTART ABEND CODE 071

PROCESSOR CHECKS  3
                                                                       PROBABLE
     TIME      JOBNAME    CUA/TYPE   ERROR DESCRIPTION               FAILING UNIT

   DATE 347/82

  08:40:55:52    N/A       N/A       BUFFER ERROR                    PROCESSOR
  15:23:03:72    N/A       N/A       REGISTER/PSW INVALID            PROCESSOR

CHANNEL CHECKS  4
                                                                       PROBABLE
     TIME      JOBNAME    CUA/TYPE   ERROR DESCRIPTION               FAILING UNIT

   DATE 347/82

  10:39:11:04  PAYROLL1  0384/3330   CHANNEL CONTROL CHECK           CHANNEL
  13:11:18:64  JOBLOADA  0233/3380   INTERFACE CONTROL CHECK         CONTROL UNIT
  *****     2 DUPLICATE LINES WITHIN THIS TIME INTERVAL HAVE NOT BEEN PRINTED
  13:14:33:09  JOBLOADA  0233/3380   INTERFACE CONTROL CHECK         CONTROL UNIT


SYSTEM ERROR SUMMARY                                  REPORT DATE 348 82
      (PART 1)                                        PERIOD FROM 347 82
                                                             TO   347 82

MODEL 3081XA SERIAL 020559   CPU B  1

IPL/RESTART/TERMINATION  2

               RECORD    TIME SINCE
     TIME       TYPE     LAST ACTIVE    REASON PROBABLE CAUSE

   DATE 347/82

  07:15:32:31  IPL-XA    12:10:52:16    NM     NORMAL SYSTEM INITIALIZATION
  10:39:49:91  TERM-XA                  EOP    IOS ERROR
  10:40:32:31  IPL-XA    00:01:10:16    NM     NORMAL SYSTEM INITIALIZATION
```

Figure 2-7 (Part 1 of 2). System Error Summary (Part 1)

```
SYSTEM ERROR SUMMARY                                   REPORT DATE 348 82
      (PART 1)                                         PERIOD FROM 347 82
                                                              TO   347 82

MODEL 0158   SERIAL 023123   CPU C  1

IPL/RESTART/TERMINATION  2

               RECORD    TIME SINCE
     TIME       TYPE     LAST ACTIVE    REASON PROBABLE CAUSE

   DATE 347/82

  08:01:30:52  IPL       19:01:32:16    NM     NORMAL SYSTEM INITIALIZATION

PROCESSOR CHECKS  3
                                                                       PROBABLE
     TIME      JOBNAME    CUA/TYPE   ERROR DESCRIPTION               FAILING UNIT

   DATE 347/82

  18:40:55:52   N/A        N/A       BUFFER ERROR                    PROCESSOR

CHANNEL CHECKS  4
                                                                       PROBABLE
     TIME      JOBNAME    CUA/TYPE   ERROR DESCRIPTION               FAILING UNIT

   DATE 347/82

  10:39:11:04  TEST12GS  0194/3330   CHANNEL CONTROL CHECK           CHANNEL
```

Figure 2-7 (Part 2 of 2). System Error Summary (Part 1)

System Error Summary, Part 1

Part 1 of the System Error Summary is a chronological listing of all machine checks and channel checks and, if applicable, IPL, restart (software), and termination records.

Notes:

1. **CPU information**. *The report is generated by CPU. This line contains the CPU model and serial numbers and a letter indicator that corresponds to the CPU letter indicators used throughout the System Exception reports. XA following the model number indicates the processor was running in 370-XA mode.*

2. **IPL/Restart/Termination section**. *This section presents records of system events. It appears only when the operating system is MVS or VSE/Advanced Function. The columns headed by REASON and PROBABLE CAUSE contain IPL reason code/restart abend code/reason code, and a brief explanation of the code. The IPL reason codes are included in Figure 10-11 in Chapter 10, "Error Records for EREP"; the possible termination reason codes are:*

EOD	*End-of-day record*
MCH	*Machine check forced termination. Non-restartable.*
EOP	*End of processing from IOS. Restartable wait state.*

3. **PROCESSOR CHECKS**. *This section appears when EREP encounters MCH records. If the JOBNAME field is blank, the failure is within an operating system task.*

   Possible ERROR DESCRIPTIONS are:

   HIR SUCCESSFUL
   POWER WARNING
   INVALID LOGOUT
   SYSTEM DAMAGE
   INSTRUCTION PROCESSOR
   HARD STORAGE ERROR
   STORAGE PROTECT KEY ERROR
   REGISTER OR PSW INVALID
   EXTERNAL DAMAGE
   BUFFER ERROR
   UNDEFINED ERROR

   Possible PROBABLE FAILING UNITS are:

   UNPROCESSED ENTRY
   PROCESSOR
   CHANNEL
   CHANNEL/DIRECTOR
   STORAGE
   CONTROL UNIT
   UNDEFINED

4. **CHANNEL CHECKS**. *This section appears if EREP encounters CCH records. If the JOBNAME field is blank, the failure is within an operating system task.*

   Possible ERROR DESCRIPTIONS are:

   INTERFACE CONTROL CHECKS
   CHANNEL CONTROL CHECKS
   CHANNEL CONTROL/INTERFACE CONTROL CHECKS
   CHANNEL DATA CHECKS
   CHANNEL DATA/INTERFACE CONTROL CHECKS
   CHANNEL DATA/CHANNEL CONTROL CHECKS
   CHANNEL DATA/CHANNEL CONTROL/INTERFACE CONTROL CHECKS

   Possible PROBABLE FAILING UNITS are the same as those appearing in the processor section of the report.

```
SYSTEM ERROR SUMMARY                                   REPORT DATE 348 82
      (PART 2)                                         PERIOD FROM 347 82
                                                              TO   347 82

                                  1                                      2                                  3
                               PHYSICAL       PHYSICAL  ERROR                                          PROBABLE
   TIME        JOBNAME CPU      ID      TYPE ADDRESS    PATH VOLUME ERROR DESCRIPTION                 FAILING UNIT

DATE 347/82

10:11:11:11 DASD0001 B   N/A      3830    0372  00-0373 663E02 PERMANENT BUS OUT PARITY CHECK    UNDETERMINED
10:31:11:12 DASD0001 B 04-XX-XX   3880    02C4  01-02C4 57ST02 PERMANENT BUS OUT PARITY CHECK    SCU
11:41:22:13 DASD0002 B   N/A      3830    0372  02-0373 TEST02 PERMANENT EQUIPMENT CHECK         UNDETERMINED
12:06:40:00 PERM101  B   N/A      3410    0685  04-0685 ABC123 LOAD POINT                        VOLUME/CD
13:16:22:10 PRIV101  B   N/A      3430    0880  13-0880 PAYROL N/A                               VOLUME/CD
16:06:40:01 PRIV221  A   N/A      3420     882      882 MASTER LO ROS/IC/BC MP2                  HARDWARE
18:10:05:49 TRENDS01 C   N/A      3480     A80      A80 TPCTRL N/A                               HARDWARE
18:25:02:49 370DXC11 B   N/A      3480    0581  06-0581 370DXA N/A                               HARDWARE
19:00:00:03 PAYROLLF A   N/A      34XX     9B6      N/A FICA01 DDR INDICATES SWAP TO PCUA   9B3  N/A
22:05:10:01 PAYROLLG A   N/A      34XX     98F      N/A FICA02 DDR INDICATES SWAP TO PCUA   98C  N/A
22:10:00:04 ATTEND12 B   N/A      34XX    0127      N/A ATTEND DDR INDICATES SWAP TO DEVNO 0128  N/A
```

Figure 2-8. System Error Summary (Part 2)

System Error Summary, Part 2

Part 2 of the System Error Summary is a chronological listing of permanent DASD and tape errors and DDR calls.

Notes:

1. **PHYSICAL ID**. *For DASD providing physical ID or DASDID statements, this field contains some combination of SCUID-CTLID-DEVID, depending on the probable failing unit. (See "Subsystem Exception Report Formats" on page 2-21.) The field contains N/A for tape or for DASD without physical ID or DASDID statements.*

2. **ERROR DESCRIPTION**. *Subsystem-dependent information. The DDR swap description appears in this field.*

3. *Possible* **PROBABLE FAILING UNITS** *are:*

   CHAN (CHANNEL)
   SCU (STORAGE CONTROL UNIT)
   CONTROLLER
   DEVICE
   VOLUME (FOR DASD) OR VOLUME/CD (FOR TAPE)
   UNDETERMINED
   UNKNOWN
   HARDWARE

   A PFU of N/A appears in the case of a DDR record.

```
7

SUBSYSTEM EXCEPTION                          REPORT DATE 180 82
     PROCESSOR                               PERIOD FROM 179 82
                                                    TO   179 82

MODEL 3033   SERIAL 066666   CPU B

   TERMINATION ERROR    1                               2              3
SERVICE LEVEL INDICATOR                             TOTAL COUNT    DATE/TIME OF LAST ERROR

    POWER WARNING    . . . . . . . . . . . . . . . .          2    179/82   11:23:45:37
    INVALID LOGOUT   . . . . . . . . . . . . . . . .          2    179/82   23:24:35:87


      HARD ERROR
SERVICE LEVEL INDICATOR                             TOTAL COUNT    DATE/TIME OF LAST ERROR

    REGISTER/PSW INVALID          . . . . . . . . .           8    179/82   13:25:46:57
    HARD STORAGE ERROR            . . . . . . . . .           4    179/82   13:35:58:77
    SYSTEM DAMAGE                 . . . . . . . . .           2    179/82   14:34:34:87
    INSTRUCTION PROCESSOR DAMAGE  . . . . . . . . .           2    179/82   11:43:45:47
    STORAGE PROTECT KEY ERROR     . . . . . . . . .           2    179/82   11:23:45:37

                                      4
  SOFT MACHINE CHECK                EXCEPTION COUNT
SERVICE LEVEL INDICATOR           60 MINUTE REFERENCE  TOTAL COUNT    DATE/TIME OF LAST ERROR

    EXTERNAL DAMAGE      . . . . . . .      2  . . .          4    179/82   13:35:58:77
    BUFFER ERROR         . . . . . . .      2  . . .          2    179/82   17:54:45:87
    HIR SUCCESSFUL       . . . . . . .      1  . . .          1    179/82   12:22:33:46


LIMITS APPLIED   EXTD=01,BUFE=01,HIRS=01   5
    0 UNIT(S) EXCLUDED DUE TO LIMITS

   20 MCH RECORD(S) PROCESSED
    1 MCH RECORD(S) UNDEFINED TO MCH ALGORITHMS     6
```

Figure 2-9. Processor Subsystem Exception Report

Processor Subsystem Exception Report

Notes:

1. **SERVICE LEVEL INDICATOR.** *May be:*

   TERMINATION ERROR
   HARD ERROR
   SOFT MACHINE CHECK

2. **TOTAL COUNT.** *The actual count of input records containing this particular error.*

3. **DATE/TIME OF LAST ERROR.** *Date and time from the last MCH record that included this error. If the date and time are the same for several service level indicators, it means that a single record included all the indicators.*

4. **EXCEPTION COUNT.** *The number of unique 60-minute intervals that had at least the LIMIT value number of this kind of soft machine check.*

5. **LIMITS APPLIED.** *The LIMIT values applied to this report.*

   Note: If the LIMIT value is zero, the EXCEPTION COUNT field is also zero.

6. *Execution-time notes. These may be:*

   nn UNITS EXCLUDED DUE TO LIMITS *(If LIMIT values are present)*

   nn MCH RECORDS UNDEFINED *(Not identifiable to EREP as valid MCH records)*

   nn MCH RECORDS IGNORED DUE TO CCH DUPLICATION *(0158 models only, from which MCH records might be double-reporting an assumed channel failure.)*

7. *This space is used for self-explanatory SCP- and device-dependent messages specific to this Subsystem Exception report. For example:*
   `**WARNING** REPORT SPANS MORE THAN 3 DAYS.`

```
[7]

SUBSYSTEM EXCEPTION                        REPORT DATE 180 82
   CHANNEL                                 PERIOD FROM 179 82
                                                  TO   179 82

MODEL 3033   SERIAL 066666   CPU B

[1]                              [2]                  [3]            [4]
SERVICE LEVEL INDICATOR           EXCEPTION COUNT     TOTAL COUNT    DATE/TIME OF LAST ERROR
                                60 MINUTE REFERENCE


  CHANNEL ERROR

      CHANNEL 6XX                        3                    4      179/82   18:47:38:67
      CHANNEL 1XX                        1                    1      179/82   21:22:23:43
      CHANNEL 2XX                        1                    1      179/82   11:34:43:65
      CHANNEL 8XX                        1                    1      179/82   11:43:32:87



  DIRECTOR ERROR

      DIRECTOR #1                        1                    1      179/82   19:32:54:89
      DIRECTOR #2                        1                    1      179/82   13:25:46:57
      DIRECTOR #3                        1                    1      179/82   11:24:36:57



  CONTROL UNIT ERROR

      CONTROL UNIT 34X                   1                    1      179/82   13:25:44:57
      CONTROL UNIT 456                   1                    1      179/82   13:32:22:37


LIMITS APPLIED   CHAN=01,DRCT=01,CTRL=01  [5]
    0 UNIT(S) EXCLUDED DUE TO LIMITS

    2 CCH RECORD(S) UNDEFINED TO CCH ALGORITHMS
    2 CCH RECORD(S) IGNORED BECAUSE OF MCH DUPLICATION  [6]
```

Figure 2-10. Channel Subsystem Exception Report

Channel Subsystem Exception Report

Notes:

1. **SERVICE LEVEL INDICATOR**. *May be:*

   CHANNEL ERROR (31XX/303X)
   CHANNEL STORAGE ERROR (31XX) OR DIRECTOR ERROR (303X)
   CONTROL UNIT ERROR

2. **EXCEPTION COUNT**. *The number of unique 60-minute intervals that had at least the LIMIT value number of this kind of channel check.*

3. **TOTAL COUNT**. *The actual count of input records containing this particular error.*

4. **DATE/TIME OF LAST ERROR**. *Date and time from the last CCH record that included this error. If the date and time are the same for several service level indicators, it means that a single record included all the indicators.*

5. **LIMITS APPLIED**. *The LIMIT values applied to this report.*

   Note: If the LIMIT value is zero, the EXCEPTION COUNT field is also zero.

6. **Execution-time notes**. *These may be:*

   nn UNITS EXCLUDED DUE TO LIMITS *(If LIMIT values are present)*

   nn INPUT RECORDS UNDEFINED *(Not identifiable to EREP as valid CCH records)*

   nn CCH RECORDS IGNORED DUE TO MCH DUPLICATION *(The number of 0158 or 0168 channel storage errors, or 303X channel errors, ignored because they might be double-reporting a processor storage error.)*

   nn CCH RECORD(S) FOUND GENERATED FOR SOFTWARE RECOVERY *(The number of sympathetic channel errors found; for 303X only.)*

7. *This space is used for self-explanatory SCP- and device-dependent messages specific to this Subsystem Exception report. For example:*
   `**WARNING** REPORT SPANS MORE THAN 3 DAYS.`

```
                                     1
SUBSYSTEM  EXCEPTION                                         REPORT DATE 179 82
       DASD                                                  PERIOD FROM 173 82
                                                                    TO   174 82

B-BUS OUT PARITY CHK   C-CHECK DATA CHK   D-DISKETTE CHK   I-INVOKED OFFSETS  2

PROBABLE                                                     ---IMPACT OF TEMPORARY ERRORS---     ---USAGE---
FAILING         FAILURE        PHYSICAL  ---TOTALS---        EQU                                  1000     MB.
UNIT            AFFECT   CPU   ADDRESS    PERM  TEMP         CHK     SKS     RD     OVRN    OTHER    SKS    READ
-------------+--------+-----+----------+------+-------+-------+------+------+------+--------+------+------+
  3    4         5        6       7           8                          9                         10
***************************************************************************************************************
CHAN  01XX     CHAN/SCU        TOTAL       1                                                      N/A    N/A
                          B    21-XX-XX    1                                                      118   3204

        01     CHAN/SCU        TOTAL       1                                                      N/A    N/A
                          A    0121        1                                                       65   4294
-------------+--------+-----+----------+------+-------+-------+------+------+------+--------+------+------+
    ** WARNING ** INVALID PHYSICAL ID ON NEXT LINE  11
SCU  01-XX-XX  CHAN/SCU        TOTAL       1                                                      N/A    N/A
       3880               B    01-XX-XX    1                                                       59    268
       12
     E0-XX-XX  SCU             TOTAL       1                                                      N/A    N/A
       3830               B    0123        1                                                       64   4026
-------------+--------+-----+----------+------+-------+-------+------+------+------+--------+------+------+
CTLR XX-03-XX  CTLR            TOTAL             1         1                                      N/A    N/A
       3375               B    21-03-04          1         1                                       58    251

     XX-E2-XX  CTLR            TOTAL             1         1                                      N/A    N/A
       3330               B    0125              1         1                                       63   3758
-------------+--------+-----+----------+------+-------+-------+------+------+------+--------+------+------+
DEV  XX-E2-07  DATAXFR         TOTAL             2                       2                        N/A    N/A
       3330               B    0127              2                       2                         62   3489

     XX-03-04  SEEK            TOTAL             1                1                               N/A    N/A
       3375               B    21-03-04          1                1                                58    251
-------------+--------+-----+----------+------+-------+-------+------+------+------+--------+------+------+
VOL   VOL002   DATAXFR         TOTAL       1                                                      N/A    N/A
       3375               B    21-14-04    1                                                       60   2952

      VOL001   DATAXFR         TOTAL             1                       1                        N/A    N/A
       3330               A    0121              1                      1                          65   4294
-------------+--------+-----+----------+------+-------+-------+------+------+------+--------+------+------+
** WARNING ** NO DASDID CARD FOUND OR INVALID PHYSICAL ID - PROBABLE UNIT NOT ASSIGNED FOR THE FOLLOWING:
               DATAXFR         TOTAL             1                       1                        N/A    N/A
       3330               B   *0223              1                       1                         57    234
-------------+--------+-----+----------+------+-------+-------+------+------+------+--------+------+------+
UNK            UNKNOWN         TOTAL                                                              N/A    N/A
       3830               B   *0139
***************************************************************************************************************
   0 UNIT(S) EXCLUDED DUE TO LIMITS  13

CPU    MODEL     SERIAL NUMBER  14
 A     3081XA    020097
 B     3033      020557

** ENTRIES WITH AN ASTERISK INDICATE THAT DASDID CARDS WERE NOT FOUND FOR THE UNIT.

NOTE: "IMPACT OF TEMPORARY ERRORS" IS THE NUMBER OF TIMES ERROR THRESHOLD HAS BEEN EXCEEDED.
NOTE: BLANK ENTRIES INDICATE ZERO VALUES OR NOT APPLICABLE. N/A = NOT AVAILABLE.
NOTE: ZERO ENTRIES INDICATE RECORDS EXIST IN EREP REPORTS BUT THRESHOLDS WERE NOT EXCEEDED.
```

Figure 2-11. DASD Subsystem Exception Report

DASD Subsystem Exception Report

This report can be used to determine if the DASD subsystem has excessive errors or is operating within acceptable limits. You can specify LIMIT controls to prevent the printing of excessive temporary errors. See "LIMIT Control Statement" under "33XX DASD" in Part 4.

The DASD exception report is organized by probable failing unit from channel to volume; within each section, the PFUs are ordered from most permanent errors to fewest temporary errors.

If this exception report indicates that corrective action is necessary, the DASD reports that follow provide the details needed for correction. See Figure 2-12 through Figure 2-16.

Notes:

1. *This space is used for self-explanatory SCP- and device-dependent messages specific to this Subsystem Exception report. For example:*
   `**WARNING** REPORT SPANS MORE THAN 3 DAYS. PFU ANALYSIS MAY BE IN ERROR.`

2. *Definitions of the suffixes for the counters that can appear in the* OTHER *column under* IMPACT OF TEMPORARY ERRORS.

3. **PROBABLE FAILING UNIT**. *The PFU is the unit most likely to be the source of the failure; the actual failure could be recorded against another unit or units. EREP identifies the PFU based on the failure affect and the units reporting errors. The accuracy of this analysis for devices without physical ID depends on DASDID control statements. The possible PFUs are:*

   CHAN — *Channel (channel, program, or CPU)*

   SCU — *Storage control unit (3830, FTA, ISC, for example)*

   CTLR — *Controller (drive string controller, or something common to more than one device on the string)*

   MULTIPLE — *Failure common to more than one device.*

   DEV — *Device (addressable unit)*

   VOL — *Volume (data on volume)*

   UNK — *Unknown (cannot be determined by report algorithms)*

   If no DASDID entry exists, or the physical ID is invalid, a warning message replaces the PFU line.

   In the line for PFU are its identifier, the failure affect, and the total errors attributed to this combination of PFU and failure affect. Usage counts are not available (N/A) because the total usage of the device is not determined in generating the report (non-failing devices are not considered).

4. **PFU identifier**. *An identifier appears for each PFU. Their formats are as follows:*

CHAN *Channel*

02XX	*02 is the channel address from the SCUAs reporting the failures.*
01	*In 370-XA mode, the channel path ID.*

SCU *Storage Control Unit*

SS-XX-XX

CTLR *Controller*

XX-CC-XX

DEV *Device (the addressable unit)*

XX-CC-DD

where:

SS	*storage control unit/director ID*
CC	*controller ID*
DD	*physical device ID.*

VOL *nnnnnn (The volume serial number from the OBR/MDR device-dependent VOLID field.)*

*When information in the DASDID is not adequate, the format is (*nnnn), where * indicates that DASDID information was inadequate and nnnn is the PCUA or device number.*

5. **FAILURE AFFECT**. *This field defines the function or machine area affected by the failure. The possible failure affects are:*

CHAN/SCU	*The channel, CPU, or program, or the channel/storage control unit interface.*
SCU	*The storage control unit.*
SCU/CTLR	*The storage control unit/controller interface.*
CTLR	*The controller.*
CTLR/DEV	*The controller/device interface.*
MULTIPLE	*Failure common to more than one device.*
DEV	*The device, including problems with a volume that must be handled by a service representative.*
SEEK	*The function of accessing the track; the failure may be in the controller, the drive, or the volume.*
DATAXFR	*Data transfer: the function of reading or writing data; the failure may be in the controller, the drive, or the volume.*
DATAXFR(HDA)	*Data transfer, where the failure is in the head disk assembly.*
UNK	*Unknown; it is possible that two failures exist, providing conflicting information.*

6. **CPU**. *The report is limited to 16 CPUs.*

7. **PHYSICAL ADDRESS**. *The physical address is the means for locating information on other EREP reports. For devices providing physical IDs, the physical ID is used. For non-physical ID devices, the address is the PCUA (its physical address) or device number.*

8. **TOTALS**.

   PERM *The count of permanent errors recorded against the unit and totaled for the PFU within the given failure affect. (A permanent error is indicated by a zero temporary error bit in the OBR record.)*

   TEMP *The sum of the counts shown for the line under* IMPACT OF TEMPORARY ERRORS.

9. **IMPACT OF TEMPORARY ERRORS.** *These fields indicate the number of temporary errors when the count exceeds a LIMIT value. Definitions of the counts of temporary errors are in the DASD maintenance manual. Types of temporary errors are:*

   EQU CHK *Temporary equipment checks*

   SKS *Temporary seek checks*

   RD *Temporary data checks during reading, corrected by retrying or by ECC (error correction code).*

   OVRN *Overruns (only applicable to a PFU of CHAN and if system retried). See "DASD Storage Control Unit Summary" on page 2-46 for total overrun count.*

   OTHER *All other temporary errors. The types are identified by the letter suffix; in the case of multiple error types, multiple letters follow the counter.*

10. **USAGE.** *The usage figures are in units of one thousand for seeks and megabytes for data read.*

11. ****WARNING** INVALID PHYSICAL ID.** *This message appears when EREP detects an SCUID of X'00' or X'01' or a CTLRID of X'00' or X'FF'. In these cases, devices might be included in the report under a PFU of CHAN, SCU, or CTLR showing only 0 or null for temporary errors.*

12. *Device type.*

13. **UNIT(S) EXCLUDED DUE TO LIMITS.** *You can limit the amount of printed output by preventing the printing of PFUs with fewer temporary errors than the limits defined on LIMIT statements. See "LIMIT Control Statement" on page 15-5. When such limits are in effect, and some PFUs would have been printed were the limits not in effect, EREP prints a message at the end of the report stating the number of PFUs not printed and the LIMIT values in effect.*

    Note: Use of the LIMIT control statement is not valid for 9332 and 9335 devices.

14. *XA following the CPU model number indicates that the processor complex was running in 370-XA mode.*

```
DASD INFORMATIONAL MESSAGES                    REPORT DATE 363 82
                                               PERIOD FROM 362 82
                                                      TO   363 82
PHYSICAL   SYMPTOM
 ID         CODE    COUNT       MESSAGE
*****************************************************************************************
02-03-01    1010      1   SECTOR RETRY THRESHOLD EXCEEDED RBN     33694
02-03-01    1313      1   THRESHOLD LOGGING COMPLETE FOR EQUIPMENT CHECKS
02-03-01    1616      1   THRESHOLD LOGGING COMPLETE FOR SEEK CHECKS
02-03-01    1919      1   THRESHOLD LOGGING COMPLETE FOR DATA CHECKS
02-03-01    2121      1   ALTERNATE BLOCKS NEARLY EXHAUSTED
06-XX-XX     N/A      1   SD/SD TEST FAILED AT IML
06-XX-XX    3A00      1   TRACE SAVED FOR STORAGE DIRECTOR
14-XX-XX     N/A      1   SD/SD TEST FAILED AT IML
22-23-01    101F      1   SECTOR RETRY THRESHOLD EXCEEDED RBN    260753
22-23-01    2072      1   CALL FOR SERVICE
40-XX-XX    000F      2   THRESHOLD LOGGING COMPLETE FOR SUBSYSTEM STORE CHECKS
40-40-XX    0002      2   THRESHOLD LOGGING COMPLETE FOR DATA CHECKS WITHOUT OFFSET
40-40-04    0001      2   THRESHOLD LOGGING COMPLETE FOR SEEK CHECKS
40-40-04    0003      2   THRESHOLD LOGGING COMPLETE FOR DATA CHECKS WITH OFFSET
```

Figure 2-12. DASD Informational Messages

DASD Informational Messages

This report provides information for the hardware service representative. The records involved are not standard sense records resulting from an error condition; rather, they may relate to a hardware failure that could degrade performance.

The DASD informational messages appear automatically following the DASD Subsystem Exception report.

Note: Information about the actions required for the various messages is in the maintenance library for the device identified in the first field.

DASD Subsystem Summaries

The next several pages present examples of the various DASD summaries included in the System Exception report series. The records and events covered in the summaries are the same ones that appear on the DASD Subsystem Exception report.

Corresponding summaries for the 34XX tape subsystem follow the "Tape Subsystem Exception Report" on page 2-47.

```
DASD STRING SUMMARY                                         REPORT DATE 188 83
                                                            PERIOD FROM 061 83
                                                                     TO 090 83
STRINGS WITH ANY DRIVE REPORTED ON THE SUBSYSTEM EXCEPTION DASD REPORT
WITH PHYSICAL ID  FROM DASDID CONTROLS OR FROM DEVICE SENSE RECORD  AND
FAILURE AFFECTS OF:  CTLR, CRLR/DEV, DEV, SEEK, OR DATAXFR

                            ERROR TYPES       SEEK                    MEGABYES
                         EQU. |SEEK |DATA   ACCESSES   MEGABYTES    WRITTEN
PHYSICAL ID  VOLUME      CHKS |     |XFER    X 1000       READ      W/VERIFY
************************************************************************
XX-1C-02     PAK181       Y
XX-1E-02     SPOOL2       Y                                 1
XX-12-00     PAK105       Y
      04     PAK109       Y
XX-16-04     PAK117       Y
XX-20-02     PAK162       Y
XX-22-00     PAK187       Y
XX-80-00     MVS120                            40         3612
      01     MVS130                            39         3605
      02     MVS150                            38          174
      03     MVS160                            45          202
      04     IBM354
      05     IBM355       Y
      06     IBM356
      07     IBM357
XX-84-00     CPDSKB       Y           Y
      01     PAK164                   Y
      04     PAK167                   Y
      06     PAK169                   Y

ALL DASD PROCESSED FOR EXCEPTION REPORT       479        10146     2
************************************************************************
CPU    MODEL    SERIAL NUMBER
 A     3081         220402
 B     3081         020402
 C     3081         020033               3
 D     3081         020631
 E     3081         020063
 F     3081         220063
 G     3081         220631
 H     3081         220033

NOTE:  THE COUNTS FOR SEEK ACCESSES x 1000, MEGABYTES READ, AND
       MEGABYTES WRITTEN W/VERIFY ARE SIX DIGIT POSITIONS.  IF THE
       SPACE IS EXCEEDED, THE COUNT IS DIVIDED BY 1000 AND A K IS
       PLACED AT THE END OF THE NUMBER.  IF THE COUNT IS EXCEEDED
       WITH A K AT THE END, 99999K WILL BE PRINTED.
```

Figure 2-13. DASD String Summary

DASD String Summary

The usage information in the DASD String Summary can help you determine whether a failure affect reported in the DASD Subsystem Exception report is unique to a particular drive or is common to more than one device in the same controller string. The String Summary is in three parts.

Notes:

1. *The first part shows failure affect and usage data for each unique combination of volume and physical ID belonging to every controller string that appeared on the Subsystem Exception Report with one of the following failure affects:*

   - CTLR
   - CTLR/DEV
   - DEV
   - SEEK
   - DATAXFR
   - DATAXFR(HDA)
   - MULTIPLE

   The first three of these failure affects are grouped under the heading of EQU.CKS; SEEK and DATA TRANSFER errors are noted under their own headings.

   The usage data for each volume/physical ID appears under three possible headings:

   - SEEK ACCESSES X 1000
   - MEGABYTES READ
   - MEGABYTES WRITTEN WITH VERIFY

   However, without valid physical IDs for the devices, or relevant failure affect data from the exception report, or usage data for the selected devices, EREP cannot generate a DASD String Summary. In these cases, the first part of the report is replaced by a message explaining the absence of report data.

2. *The second part of the String Summary shows the usage statistics for ALL DASD processed for the Subsystem Exception report, regardless of failure affect, physical ID, and whether any failures were reported.*

3. *The third part of the summary shows all CPUs processed for the exception report, by letter identifier, model number, and serial number.*

```
DASD SYMPTOM CODE SUMMARY                                  REPORT DATE 179 82
                                                           PERIOD FROM 173 82
                                                                  TO   174 82
SEQUENCE BY PROBABLE FAILING UNIT  1
2         3         5         6                         9
SYMPTOM  PHYSICAL  OCCUR-   FAILURE              DATE AND TIME OF
CODE     ID        RENCES   AFFECT     FIRST OCCURRENCE      LAST OCCURRENCE
                                             SENSE FROM FIRST OCCURRENCE
           DEVICE         0 0 0 0  0 0 0 0  0 0 1 1  1 1 1 1  1 1 1 1  2 2 2 2  11
           TYPE    PRM/TMP 0 1 2 3  4 5 6 7  8 9 0 1  2 3 4 5  6 7 8 9  0 1 2 3
           4                3        8        10
                          PHYSICAL   ERROR
                          ADDRESS    PATH       CPU(S)
*******************************************************************************
PROBABLE FAILING UNIT: CHANNEL------------------------------------------------
SEQUENCE BY SCUID, SYMPTOM CODE ----------------------------------------------
2         3         5         6                         9
0F01 *   21-XX-XX   1   0   CHAN/SCU   174/82 08:10:30:07     174/82 08:10:30:07
            3880          04010203 04050607 08091011 12131415 16171819 20210F01  11
            4               7          8         10
                            21-XX-XX   012C       A

0901 *   E0-XX-XX   1   0   CHAN/SCU   174/82 08:10:30:01     174/82 08:10:30:01
            3830          04010203 04050607 08091011 12131415 16171819 20210901
                              0121       0121       A

PROBABLE FAILING UNIT: STORAGE CONTROL UNIT-----------------------------------
SEQUENCE BY SCUID, SYMPTOM CODE ----------------------------------------------

0F01 *   01-XX-XX   1   0   CHAN/SCU   174/82 08:10:30:08     174/82 08:10:30:08
            3880          04010203 0405063B 08091011 12131415 16171819 20010F01
                            01-XX-XX   012E       A

0908 *   E0-XX-XX   1   0   SCU        174/82 08:10:30:02     174/82 08:10:30:02
            3830          00010203 04050607 08091011 12131415 16171819 20210908
                              0123       0123       A

PROBABLE FAILING UNIT: CONTROLLER---------------------------------------------
SEQUENCE BY CTLID, SYMPTOM CODE ----------------------------------------------

191D *   XX-03-XX   0   1   CTLR       174/82 08:10:30:09     174/82 08:10:30:09
            3775          04010203 04050613 08091011 12131415 16171819 2021191D
                            21-03-04   0134       A

PROBABLE FAILING UNIT: NO DASDID CARD OR UNKNOWN------------------------------
SEQUENCE BY PCUA, SYMPTOM CODE -----------------------------------------------

090B      N/A       0   0   UNKNOWN    174/82 08:10:30:13     174/82 08:10:30:13
            3830          00010203 04050603 08091011 12131415 16171819 2021090B
                              0139       0139       A

191A *    N/A       0   1   DATAXFER   174/82 08:10:30:06     174/82 08:10:30:06
            3330          04011203 04050643 08091011 12131415 16171819 2021191A
                              022B       022B       A
************************************************************************
CPU    MODEL    SERIAL NUMBER
 A     3081XA        020097
 B     3033          020558

NOTE: SYMPTOM CODES WITHOUT AN ASTERISK ARE INFORMATIONAL ONLY
NOTE: PHYSICAL ID OF N/A MEANS THERE WERE NO DASDID CARDS
```

Figure 2-14. DASD Symptom Code Summary

DASD Symptom Code Summary

This report provides information required for hardware maintenance. Each sense (OBR) record reported in the exception report is listed by PFU, fault symptom code, and physical ID.

The symptom code is to be used with the maintenance procedures for the device. The report allows the service representative to locate the failures noted in the DASD Subsystem Exception report and to note the symptom code and first sense record for each failure. Data checks (symptom codes 4XXX and 5XXX), which appear in the DASD Data Transfer Summary, also appear here, for use when hardware repair is required.

Notes:

1. *The overall sequence of this report is by probable failing unit.*

2. **SYMPTOM CODE**. *A fault symptom code recorded for this PFU. All symptom codes except those for records collected in logging mode are followed by an asterisk (*). (Records collected in logging mode do not appear on the subsystem exception report.) The symptom code that appears for the format 5 (ECC-correctable) OBR record is a dummy created by duplicating the contents of sense byte 7. (If sense byte 7=53, the symptom code is 5353.)*

   In the report, the first line for each symptom code shows the code itself, the physical ID of the device, the count of permanent and temporary errors for that symptom code, the failure affect for that symptom code, and the date/time of recording the first and last sense records received for that symptom code.

   The second line of each entry shows the device type and the first sense record received for the particular symptom code.

   The following lines show the physical and logical addresses involved with recording the particular symptom code, and the CPU on which the record was created.

3. **PHYSICAL ID**. *For DASD providing physical IDs or DASDID statements, this field contains some combination of SCUID-CTLID-DEVID,* [2] *depending on the probable failing unit. For other devices, the field contains N/A. Refer to the DASD Subsystem Exception Reports PFU Identifier for the format of the physical ID.*

4. **DEVICE TYPE**.

5. **OCCURRENCES**. *The number of permanent and temporary errors encountered for this symptom code, this physical ID, and this failure affect.*

---

2 See the DASD section of Part 4, "Product-Dependent Information" for exceptions.

6. **FAILURE AFFECT**. *This field defines the function or machine area affected by the failure. The possible failure affects are:*

CHAN/SCU	*The channel, CPU, or program, or the channel/storage control unit interface.*
SCU	*The storage control unit.*
SCU/CTLR	*The storage control unit/controller interface.*
CTLR	*The controller.*
CTLR/DEV	*The controller/device interface.*
MULTIPLE	*Failure common to more than one device.*
DEV	*The device, including problems with a volume that must be handled by a service representative.*
SEEK	*The function of accessing the track; the failure may be in the controller, the drive, or the volume.*
DATAXFR	*Data transfer: the function of reading or writing data; the failure may be in the controller, the drive, or the volume.*
DATAXFR(HDA)	*Data transfer, where the failure is in the head disk assembly.*
UNK	*Unknown; it is possible that two failures exist, providing conflicting information.*

7. **PHYSICAL ADDRESS**. *If the device provides physical ID, this is the same as the physical ID. Otherwise, it is the PCUA or device number.*

8. **ERROR PATH**. *The address from which the record was received. In 370-XA mode, the format is CHPID-device number (01-0120).*

9. **DATE AND TIME**. *The date and time of the first and last occurrences of the sense records for this symptom code.*

10. **CPU(s)**. *The EREP-assigned CPU letter indicator.*

11. **SENSE FROM FIRST OCCURRENCE**. *The first sense record received for this symptom code. There may be either 24 or 32 bytes of sense data.*

```
DASD DATA TRANSFER SUMMARY                               REPORT DATE 176 82
PROBABLE FAILING UNIT - VOLUME                           PERIOD FROM 173 82
                                                                TO   174 82

                                                                  SENSE COUNTS  7
                                                                    TEMPORARY
                                                                   OFFSET INVK THRESHOLD
                                                              PERM   NO    YES   LOGGING
*************************************************************************************
SEQUENCE BY VOLUME LABEL. PHYSICAL ADDRESS, HEAD, CYLINDER

  1                                            2
UNITADDRESS 0134  DEVTYPE 3375         VOLUME VOL002
     CPU A   PHYSICAL ADDRESS XX-14-04
      3      4

     FAILURE AT ADDRESS: CYLINDER 0005  HEAD 06            1     0     0     0  5
     LAST SENSE AT: 174/82 08:10:30:11  6
   04010203 04050643 08091011 12131415 16171819 20214941
-------------------------------------------------------------------------------------
THE FOLLOWING ENTRIES HAVE ONLY MDR RECORD TYPES.  THEREFORE, NO CYLINDER/HEAD  8
ADDRESSES ARE REPORTED.  SEE THE EXCEPTION REPORT FOR THE ERROR COUNTS.

UNITADDRESS 0121   DEVTYPE 3330         VOLUME VOL001
     CPU A   PHYSICAL ADDRESS 0121


DASD DATA TRANSFER SUMMARY                               REPORT DATE 176 82
PROBABLE FAILING UNIT - OTHER                            PERIOD FROM 173 82
                                                                TO   174 82

                                                                  SENSE COUNTS  7
                                                                    TEMPORARY
                                                                   OFFSET INVK THRESHOLD
                                                              PERM   NO    YES   LOGGING
*************************************************************************************
  1                                            2
UNITADDRESS 0126  DEVTYPE 3330         VOLUME DEV001
     CPU A   PHYSICAL ADDRESS 0127
      3      4

     FAILURE AT ADDRESS: CYLINDER 0005  HEAD 06            0     1     0     0  5
     LAST SENSE AT: 174/82 08:10:30:04  6
   00011203 04050643 08091011 12131415 16171819 2021191A

UNITADDRESS 022A   DEVTYPE 3330         VOLUME UNK001
     CPU A   PHYSICAL ADDRESS 022B

     FAILURE AT ADDRESS: CYLINDER 0005  HEAD 06            0     1     0     0
     LAST SENSE AT: 174/82 08:10:30:06
   04011203 04050643 08091011 12131415 16171819 2021191A
*************************************************************************
CPU    MODEL    SERIAL NUMBER
 A     3081XA      020097
 B     3033        020558

NOTE: CYLINDER/HEAD/BLOCK NUMBERS ARE DECIMAL VALUES
NOTE: UNITADDRESS IS THE LOGICAL ADDRESS OF THE DEVICE
```

Figure 2-15. DASD Data Transfer Summary

DASD Data Transfer Summary

This report lists each volume for which data checks appeared in the DASD Subsystem Exception report, giving the error locations for each. It can be in two parts: PFU of volume and PFU of other than volume. In general,

- A PFU of volume implies that the first attempt at correction should be data correction by a utility program such as the Device Support Facility.
- A PFU of other implies that the first attempt at correction should be performed by a service representative. See the *IBM Disk Storage Management Guide - Error Handling* for recommended actions.

Notes:

1. **UNITADDRESS**. *The keyword used by the Device Support Facility to identify the device. It is the logical address (SCUA) or device number of the volume reporting the error.*
2. **VOLUME**. *The volume serial number of the volume reporting the error.*
3. **CPU**. *The CPU identified for the last sense record.*
4. **PHYSICAL ADDRESS**. *For devices providing physical IDs, this is the physical ID; for other devices, it is the PCUA or physical device number.*
5. **FAILURE AT ADDRESS**. *The location of the data check. For count key data (CKD) devices, the "address" is expressed as cylinder and head; for fixed block (FBA) devices, it is expressed as block number. The values are in decimal. When a volume records data checks at more than one location, this report includes an entry for each location; they are in ascending order of the head/cylinder or block.*
6. **LAST SENSE AT**. *The date and time of the last sense record received for this cylinder/head or block. The sense data precedes this line. There may be either 24 or 32 sense bytes. The format of the sense record is in byte 7. For format-4 records, the symptom code is in the last two sense bytes; for format-5 records, it is the value in byte 7 repeated. The date and time follow the sense bytes.*
7. **SENSE COUNTS**. *These columns contain counts of the data checks for the particular cylinder/head or block. The permanent data checks appear in the first column. The temporary data checks are broken down as follows:*
   - OFFSET INVK *(offset invoked)* NO/YES *indicates the number of temporary data checks that were recovered, and whether it was necessary to offset the access mechanism.*
   - THRESHOLD LOGGING *indicates the number of temporary data checks recorded when the device was in logging mode because the threshold for data checks was exceeded.*

8. **THE FOLLOWING ENTRIES . . .** *In cases where the only error data is from error counters, meaning that failure addresses are not available, only the lines that define the device and volume appear.*

```
DASD STORAGE CONTROL UNIT SUMMARY                          REPORT DATE 363 82
                                                           PERIOD FROM 362 82
                                                                  TO   363 82

******************************************************************************

PHYSICAL ID 02-XX-XX   DEVTYPE 3880

  OVERRUNS  INTF-A  INTF-B  INTF-C  INTF-D  INTF-E  INTF-F  INTF-G  INTF-H  1
  CMND           5      15      25      35      45      55      65      75
  DATA          10      20      30      40      50      60      70      80

  DISKETTE READER   SEEK      DATA
  TEMPORARY CHECKS     1         1

PHYSICAL ID 04-XX-XX   DEVTYPE 3830   PHYSICAL ADDR 084X      CPU(S) ABCD

  OVERRUNS  INTF-A  INTF-B  INTF-C  INTF-D
  CMND          10      30       5     120
  DATA          20      10      10       5

PHYSICAL ID  N/A       DEVTYPE 3830   PHYSICAL ADDR 089X      CPU(S) A

  OVERRUNS  INTF-A  INTF-B  INTF-C  INTF-D
  CMND           5       0       0       0
  DATA          80       0       0       0

******************************************************************************

CPU    MODEL     SERIAL NUMBER
 A     0168          060328
 B     4341          022222
 C     3081XA        012345
 D     3032          076543
```

Figure 2-16. DASD Storage Control Unit Summary

DASD Storage Control Unit Summary

This report defines the physical channel interface over which overruns occurred for the 3830 and 3880 storage control units (SCU).

Notes:

1. **INTF-A . . . INTF-H**. *The Storage Control Unit channel interface.*

```
                              1
SUBSYSTEM EXCEPTION                                    REPORT DATE 341 82
      TAPE                                             PERIOD FROM 340 82
                                                              TO   340 82



                                     --------------------- 3420 ---------------------    --------- 3410 ---------
                                             1600 BPI                  6250 BPI                   1600 BPI
                                     TEMP WRT(CT) TEMP RD(CT)  TEMP WRT(CT) TEMP RD(CT)   TEMP WRT(CT) TEMP RD(CT)
CURRENT LIMITS    HARDWARE                4 (15)     26 ( 2)        4 (15)     26 ( 2)         4 (15)     26 ( 2)
MBYTES/ERR(CT)    VOLUME                  4 (15)     26 ( 2)        4 (15)     26 ( 2)         4 (15)     26 ( 2)
  2
                               C                                                                     10       11     12
            VOLUME DEVNO P EQU  ---MB/ERR  PERM---  ---MB/ERR  TEMP--- BUS OVR            TOTAL-MBYTES DEN-   HDR
EXCEPTION   SERIAL /CUA  U CHK  READ(CT) WRITE(CT)  WRITE(CT)  READ(CT) OUT RUN I/O CNT  READ  WRITE SITY   SER
 3                  4      5                        6                    7   8   9
  HARDWARE
    PERMANENT ERROR
                   057X  A   0  -- (  0)  -- (  0)   -- (  0) 271 (21)   0   0      2M  5701  4391*  --
                   0576  A   4   6 (  2)  -- (  0)   -- (  0)  -- ( 0)   0   0      4K    12    11* 6250

  VOLUME OR CREATING DRIVE
    PERMANENT READ OR WRITE ERRORS OR RESET KEY ON MORE THAN ONE DRIVE
            T73759 0571  A   0   0 (  2)  -- (  0)   -- (  0)  -- ( 0)   0   0      54     0     0*  -- 03433

  VOLUME
    FAILED TEMPORARY READ OR WRITE LIMITS
            74347  0570  A   0  -- (  0)  -- (  0)    1 ( 16)  -- ( 0)   0   0      3K     0    21*  --

  TOTAL NUMBER OF DRIVES ON REPORT  1 ( 10%)             TOTAL NUMBER OF VOLUMES USED    =    76
                      NOT ON REPORT  9 ( 90%)             TOTAL NUMBER OF VOLUMES LISTED =     2
                                                       13
  (*) AN AVERAGE BLOCK LENGTH WAS USED BECAUSE A ZERO BLOCK LENGTH WAS FOUND IN ONE OR MORE OBR RECORDS
  AVERAGE BLOCK LENGTH =    6324
CPU    MODEL    SERIAL NUMBER   14
 A     3081XA       020098
```

Figure 2-17. Tape Subsystem Exception Report

Tape Subsystem Exception Report

This report indicates if the tape subsystem has permanent errors or is operating within acceptable limits. You can use LIMIT controls for both hardware and volume to prevent the printing of excessive temporary errors. See "LIMIT Control Statement" in the 34XX Tape Devices section of Part 4.

Note: When you do not specify limits for temporary errors, only permanent errors appear on the Subsystem Exception report.

If the Tape Subsystem Exception report indicates that corrective action is necessary, the summary reports that follow provide the details required for correction.

The Tape Subsystem Exception report format and contents vary somewhat according to the device type involved. See Chapter 17, "Magnetic Tape Drives" on page 17-1 for more information about specific products. The notes that follow are related to the report example in Figure 2-17.

Notes:

1. *This space is used for self-explanatory SCP- and device-dependent messages specific to this Subsystem Exception report.*

2. **CURRENT LIMITS**. *These two lines show the limits set in LIMIT control statements included when the report was requested. The limits are set for temporary read and write errors occurring in the HARDWARE (device) and on the VOLUME. They prevent the printing of all temporary errors but those that occurred at the specified frequency level. These lines do not appear on the report for a 9347 device. Use of the LIMIT control statement is invalid for 9347 devices. See "LIMIT Control Statement" on page 17-4 for details on using the LIMIT statement.*

   In the tape reports, the current limits are grouped under device type (3420 and 3410, in this example), tape density (1600 BPI and 6250 BPI), and TEMP WRT and TEMP RD. They show the specified number of megabytes read or written per error (MBYTES/ERR) and the specified total number of errors recorded (CT). In this report, for example, temporary write errors recorded against a 3420 drive (hardware) at a density of either 1600 or 6250 BPI will not appear as exceptions unless there are at least 15 errors occurring at a rate of 1 or more errors in each 4 megabytes of data written. Temporary read errors on a 3420 at either density will qualify as exceptions if there are 2 or more errors occurring at a rate of 1 or more errors in each 26 megabytes of data read.

3. **EXCEPTION**. *Errors are presented in this report according to exception type; the type of exception serves as an indicator of the suspected source of the problem. There are five exception types listed on the Tape Subsystem Exception report:*

   a. ***HARDWARE — PERMANENT ERROR:*** *All CUAs with a tape permanent error are listed here or under part 3c. If the CUA (ADDR) has an X as its last digit, the problem is likely to be with the control unit rather than the tape drive. Details of the permanent errors can be found in the Tape Permanent Error Summary, Figure 2-18.*

   b. ***HARDWARE — FAILED TEMPORARY READ OR WRITE LIMITS:****All CUAs with an error rate equal to or below the specified hardware limits are listed here. For more detail, see the Tape DEVNO/CUA Statistics Summary (Figure 2-20) and the Tape Volume Statistics Summary (Figure 2-21).*

   c. ***VOLUME OR CREATING DRIVE — PERMANENT READ OR WRITE ERRORS OR RESET KEY ON MORE THAN ONE DRIVE:****The volume indicated under VOLUME SERIAL has permanent errors on more than one drive. The problem could be with the tape itself, or with the drive where the tape was created. For more detail, see the Tape DEVNO/CUA Statistics Summary (Figure 2-20) and the Tape Volume Statistics Summary (Figure 2-21).*

   d. ***VOLUME OR CREATING DRIVE — FAILED TEMPORARY READ OR WRITE LIMITS ON MORE THAN ONE DRIVE:****The volume indicated under VOLUME SERIAL has an error rate equal to or below the specified volume MBYTE/ERROR limit on more than one drive. The error count LIMIT value is not used for this category of exceptions. The problem could be with the tape itself, or with the drive where the tape was created. For more detail, see the Tape DEVNO/CUA Statistics Summary (Figure 2-20) and the Tape Volume Statistics Summary (Figure 2-21).*

   e. ***VOLUME — FAILED TEMPORARY READ OR WRITE LIMITS:****All CUAs or device numbers that had a temporary error rate equal to or below the specified volume limits are listed here. The problem could be with the device or with the volume/tape. For more details, see the Tape DEVNO/CUA Statistics Summary (Figure 2-20) and the Tape Volume Statistics Summary (Figure 2-21).*

4. **DEVNO/CUA**. *This column contains the device numbers or CUAs (channel/unit addresses) against which the permanent and temporary errors were recorded.*

5. **EQU CHK**. *Count of equipment checks.*

6. **MB/ERR PERM and MB/ERR TEMP**. *These columns list the actual numbers of errors recorded during the report period, and the actual error rate, for this CUA. For example, CUA 37E recorded 1 permanent read error in 28 megabytes of data read; CUA 9B8 recorded a total of 7 temporary write errors in 164 megabytes of data written.*

7. **BUS OUT**. *The bus out check count from the statistical data recorder (SDR) counters.*

8. **OVR RUN**. *The overrun count from the SDR counters.*

9. **I/O CNT**. *The number of START I/O instructions issued against this CUA during the report period, in thousands.*

10. **TOTAL MBYTES READ WRITE.** *Total number of megabytes read or written during the report period.*

11. **DENSITY**. *The tape density, from the mode byte in the OBR record.*

12. **HDR SER**. *The serial number of the creating drive, from the tape header.*

13. *Number of drives (hardware) and volumes appearing on this report, in comparison to the total number of drives and volumes.*

14. *The model and serial number(s) of the CPU(s) designated in the report by letter. XA following the model number indicates that the processor was running in 370-XA mode.*

```
TAPE PERMANENT ERROR SUMMARY                                REPORT DATE 341 82
                                                            PERIOD FROM 340 82
                                                                   TO   340 82

                                     2                                        7
 1             C                     R  3   4    5   6                   ...SENSE...                                        8
CHP DEVNO P                          W              SCSW64-95/                 1        1        2                          HDR
-ID /CUA  U DTE  TIME   VOLID        E CMD FLG  CNT CSW32-63  0        4        8        2        6        0     EXPLANATION  SER

**** HARDWARE ****

 05 0571  A 340 161934 73373  R  02 00000050 0E400050 08C2000C 00482D00 00020000 002DC221 0C9170D9 001A0002 START RD CHECK   08460
 05 0371  A 340 161951 73373  R  02 C000C050 0E100050 02C200CC C0122D00 CCC20000 002DC221 0C9170D9 001A0002 START RD CHECK   08160

 05 0574  A 340 161521 T71972 R  02 24001008 0E001008 08C2C000 00483D00 00030000 002DC221 059170CA 001A0012 START RD CHECK   08453
 05 0574  A 340 163123 F352JL    0C 0000AC38 0E20100B C2120000 00403D00 00020000 002DC221 039170E7 001A00C0 WORD CNT ZERO    08453
 C5 0574  A 340 171229           37 00000030 E6030001 00410000 00403D00 0002C000 C02DC221 059170EF 001A0010 NOT CAPABLE      08453

**** VOLUME OR CREATING DRIVE ****

 05 0570  A 340 115423 T73579 R  02 44001008 0E400A19 08C2FF88 00443D00 00380000 002DC220 F19170BE 001A0005 PARTIAL RECORD   08433
 05 0571  A 340 115708 T73579 R  02 44001008 0E400A19 0CC2FFC8 00443D00 00330000 002DC221 CC9170F5 001A0004 PARTIAL RECORD   08433
   NOTE: TO CONVERT 'HDR SER' TO 'CUA' USE 'TAPE UNIT SER' IN 'TAPE TEMPORARY ERROR SUMMARY' (NEXT REPORT).

CPU    MODEL    SERIAL NUMBER
 A      3081XA       020098
```

Figure 2-18. Tape Permanent Error Summary

Tape Permanent Error Summary

This report describes in more detail the permanent errors that appear on the Tape Subsystem Exception report. The errors are grouped under hardware or volume/creating drive, and listed by CUA or VOLID (volume serial number) in the order they occurred.

Notes:

1. **CHPID**. *The channel path ID from the permanent OBR record.*
2. **R W E**. *These letters indicate which kind of permanent error is involved: READ, WRITE, or EQUIPMENT CHECK.*
3. **CMD**. *The command code from the CCW in the OBR record.*
4. **FLG**. *The flag byte from the CCW in the OBR record.*
5. **CNT**. *The byte count from the CCW in the OBR record.*
6. **SCSW64-95/CSW32-63**. *Bits 64-95 of the SCSW, or bits 32-63 of the CSW, in the OBR record.*
7. **SENSE**. *The sense bytes from the OBR record.*
8. **HDR SER**. *The serial number of the creating drive.*

```
TAPE TEMPORARY ERROR SUMMARY                              REPORT DATE 341 82
                                                          PERIOD FROM 340 82
                                                                 TO   340 82


  1    2      1    3       4           5                 6                          7
      TAPE C                              WRITE              READ
DEVNO UNIT P DEN- TOTAL    TOTAL     STATISTICS         STATISTICS      ENV MTE SRC EDC VEL SKEW R/W WTM PAR/ OVER IBG
/CUA   SER  U SITY I/O CNT MOUNT MB/ERR(CT) ERSGAP MB/ERR(CT) CLNACT VRC LRC /PC CRC CHG  ERR VRC CHK TACH  RUN DET

0570   N/A A 6250  99938     20  -- (   0)      0  -- (   0)     18   1   0   0   0   0    0   1   0    1    0   0
0570   N/A A OTHR      2      2  -- (   0)      0  -- (   0)      0   0   0   0   0   0    0   0   0    0    0   0

0571 88460 A 6250 332788     42  249(   4)      4  -- (   0)    132   5   1   9   0   0    6   2   0    0    0   0
0571 88460 A 1600  94562      7   35(   8)      8  -- (   0)     92  12   6   6   0   0    4   0   0    0    0   2
0571 88460 A OTHR      1      1  -- (   0)      0  -- (   0)      0   0   0   0   0   0    0   0   0    0    0   0


6250BPI TOTALS:   432726     62     (   4)      4     (   0)    150
1600BPI TOTALS:    94562      7     (   8)      8     (   0)     92  8
OTHRBPI TOTALS:        3      3     (   0)      0     (   0)      0

    TOTALS:       527291     72     (  12)     12     (   0)    242
    AVERAGE MEGABYTES/TEMPORARY READ ERROR   =       178
    AVERAGE MEGABYTES/TEMPORARY WRITE ERROR  =       116
    AVERAGE MEGABYTES/PERMANENT READ ERROR   =       513   9
    AVERAGE MEGABYTES/PERMANENT WRITE ERROR  =    ------
    AVERAGE MEGABYTES/PERMANENT ERROR        =       919
    TOTAL MEGABYTES PROCESSED                =     10107
CPU    MODEL    SERIAL NUMBER
 A     3081XA      020098
```

Figure 2-19. Tape Temporary Error Summary

Tape Temporary Error Summary

This report presents all the temporary read/write errors recorded for tape hardware during the report period. The LIMIT control values specified when invoking EREP are ignored for this report.

Notes:

1. **DEVNO/CUA** *and* **DENSITY**. *The errors are listed by CUA or device number and density.*

2. **TAPE UNIT SER**. *The serial number of the tape unit. If this information is not available because the CUA did not have a permanent error, N/A appears in this field.*

3. **TOTAL I/O CNT**. *The number of STARTIOs issued against this device at this density.*

4. **TOTAL MOUNT**. *The number of volume mounts for this device at this density.*

5. **WRITE STATISTICS**. *The actual number of megabytes written per temporary write error, and the actual number of temporary write errors, by this device at this density. The number of erase gaps on this device at this density are totalled under ERSGAP.*

6. **READ STATISTICS**. *The actual number of megabytes read per temporary read error, and the actual number of temporary read errors, by this device at this density. The number of cleaner actions on this device at this density are listed under CLNACT.*

7. *These columns contain counts from the statistical data recorders, as follows:*

ENV VRC	*total envelope/vrc count*
MTE LRC	*total mte/lrc count*
SRC/PC	*for 3420, the total start read check count; for 3410, the total parity compare count*
EDC CRC	*total edc/crc count*
VEL CHG	*total velocity change count*
SKEW ERR	*total skew error count*
R/W VRC	*total rw/vrc count*
WTN CHK	*total write tape mark check count*
PAR/TACH	*for 3420, total partial record count; for 3410, total tach check count*
OVER RUN	*total overrun count*
IBG DET	*total ibg detected count*

8. *Totals of STARTIOs, mounts, write and read statistics for each tape density.*

9. *Averages and totals for megabytes of data processed during the report period.*

```
TAPE DEVNO/CUA STATISTICS SUMMARY FOR - 0570        REPORT DATE 341 82
                                                    PERIOD FROM 340 82
                                                           TO   340 82

                                         ----------------------- 3420 -----------------------
                                                 1600 BPI                      6250 BPI
           1 CURRENT LIMITS                 TEMP WRT(CT)  TEMP RD(CT)  TEMP WRT(CT)  TEMP RD(CT)
             MBYTES/ERR(CT)     HARDWARE        4 (15)      26 ( 2)       4 (15)      26 ( 2)
                         3                                                       6
                         R            4                5                                                                                C
VOLUME 2    TIME         W      -----MB/ERR TEMP-----        ENV  MTE  POST C/P  CRC  WTRG --SLOW-- CHAN ERLY PART  TIE   P CHP 7DEN- 8HDR
SERIAL DTE HH:MM:SS      E      WRITE(CT)       READ(CT) I/O CNT VRC  LRC  AMBL COMP III  VRC  BEG  END BUFF BRBC REC   P 07  U -ID SITY SER

T77177 340 10:29:51       -- (     0)  -- (     0)  27758   0    0    0    0    0    0    0    0    0    0    0    0 00  A      6250 08433
T73579 340 11:54:23 R     -- (     0)  -- (     0)     20   1    0    0    0    0    0    0    0    1    0    1    0 00  A   05 6250 08433
T73579 340 11:56:34       -- (     0)  -- (     0)      0   0    0    0    0    0    0    0    0    0    0    0    0 00  A      6250 08433
T73579 340 11:56:34       -- (     0)  -- (     0)      1   0    0    0    0    0    0    0    0    0    0    0    0 00  A      OTHR 08433
T73089 340 16:07:54       -- (     0)  -- (     0)    384   0    0    0    0    0    0    0    0    0    0    0    0 00  A      6250 08433
T73089 340 16:43:18       -- (     0)  -- (     0)    296   0    0    0    0    0    0    0    0    0    0    0    0 00  A      6250 08433
T74512 340 17:40:09       -- (     0)  -- (     0)    116   0    0    0    0    0    0    0    0    0    0    0    0 00  A      6250 07113

      COLUMN TOTALS:         (     0)     (     0)  28585   1    0    0    0    0    0    0    0    1    0    1

             TOTALS:    MOUNTS =        5

   9 AVERAGE MEGABYTES/TEMPORARY READ ERROR   =  --
     AVERAGE MEGABYTES/TEMPORARY WRITE ERROR  =  --

CPU     MODEL     SERIAL NUMBER
 A      3081XA       020098
```

Figure 2-20. Tape DEVNO/CUA Statistics Summary

Tape DEVNO/CUA Statistics Summary

One of these reports is generated for each device (device number or CUA) that appears as a hardware exception on the Tape Subsystem Exception report. The report presents the DEVNO/CUA's temporary errors that failed the limits set in LIMIT control statements. The errors appear by volume serial number in the order (date and time) in which they occurred.

Notes:

1. **CURRENT LIMITS**. *Because the report deals with temporary errors, the current LIMIT values for hardware appear for comparison.* ***The LIMIT control statement is invalid for 9347 devices.***

2. **DTE**. *The date and time are from the OBR record.*

3. **R W E U**. *This column contains the permanent errors against this volume. R indicates read errors, W indicates write errors, E indicates equipment checks, and U indicates unknown.*

4. **MB/ERR TEMP**. *The actual error count and error frequency for this volume mount.*

5. **I/O CNT**. *The number (from the OBR record) of START I/O instructions issued against this CUA while this volume was mounted.*

6. *These columns contain counts from the statistical data recorders, as follows:*

ENV VRC	*total envelope/vrc count*
MTE LRC	*total mte/lrc count*
POST AMBL	*post-amble count*
C/P COMP	*c/p compare count*
CRC III	*crc iii count*
WRTG VRC	*write trigger vrc count*
SLOW – BEG – END	*slow begin and end counts*
CHAN BUFF	*channel buffer count*
ERLY BRBC	*early begin read back check count*
PART REC	*partial record count*
TIE	*track in error parity bit (P) and byte (07)*
CPU	*as identified by EREP*
CHPID	*channel path ID from the permanent error OBR record*

7. **DENSITY**. *From the mode byte in the OBR record.*

8. **HDR SER**. *The serial number of the creating drive, from the tape header.*

9. *Totals and averages of the data presented in the report.*

```
TAPE VOLUME STATISTICS SUMMARY                        REPORT DATE 341 82
                                                      PERIOD FROM 340 82
                                                             TO   340 82

        VOLUMES EQUAL TO OR BELOW LIMITS OR PERMANENT ERRORS

                              ----------------------- 3420 -----------------------     --------- 3410 ----------
                                      1600 BPI                   6250 BPI                      1600 BPI
                              TEMP WRT(CT)  TEMP RD(CT)  TEMP WRT(CT)  TEMP RD(CT)     TEMP WRT(CT)  TEMP RD(CT)
CURRENT LIMITS                     4 (15)       26 ( 2)       4 (15)       26 ( 2)          4 (15)       26 ( 2)
MBYTES/ERR(CT)    VOLUME           4 (15)       26 ( 2)       4 (15)       26 ( 2)          4 (15)       26 ( 2)
  1


  2               3        4         5                   6                 7                          8
                           R                                                                       C
VOLUME                CHP DEVNO W ---MB/ERR  PERM--- ---MB/ERR  TEMP--- CLNR ERASE  I/O BLK  --JOB--- P DEN-  HDR
SERIAL DTE HH:MM:SS -ID /CUA  E READ(CT) WRITE(CT) WRITE(CT) READ(CT) ACTS  GAPS  CNT LNG  --NAME-- U SITY  SER

F36BJL 340 14:36:56      0571   -- (  0)  -- (  0)  -- (  0) -- (  0)   19     0 4285 18B4 F36BJL1T A 6250 08460
F36BJL 340 16:34:28  05  0574   -- (  0)  -- (  0)  -- (  0) -- (  0)    9     0 4082 0FEC F36BJL1K A 6250 08453

NONSHR 340 10:03:58       580 E -- (  0)  -- (  0)  -- (  0) -- (  0)    0     0    2 18B4 TESTTEST B 6250 00000
NONSHR 340 10:17:50       580   -- (  0)  -- (  0)  -- (  0) -- (  0)    0     1 1136 18B4 TESTSEST B 6250 08460

NONXA  340 11:03:58       580 E -- (  0)  -- (  0)  -- (  0) -- (  0)    0     0    2 18B4 TESXA    B 6250 00000
NONXA  340 11:17:50       580   -- (  0)  -- (  0)  -- (  0) -- (  0)    0     1 1136 18B4 TESXA    B 6250 08460


     COLUMN TOTALS:             (  0)     (  0)     (  2)    (  0)   28     2 10643

            TOTALS:     MOUNTS =         4
            TOTALS:     MEGABYTES PROCESSED =     841  9

CPU     MODEL     SERIAL NUMBER
 A      3081XA        020098
 B      3081          020098
```

Figure 2-21. Tape Volume Statistics Summary

Tape Volume Statistics Summary

This report shows all the activity for every volume listed as an exception on the Tape Subsystem Exception report. Entries are grouped by volume serial and listed in chronological order.

Notes:

1. **CURRENT LIMITS**. *Because the report includes temporary errors, the current LIMIT values for volumes appear for comparison.*

2. **VOLUME SERIAL**. *There is no volume serial number if the tape is unlabelled or cannot be read.*

3. **CHPID**. *The channel path ID from the permanent error OBR record.*

4. **R W E U**. *This column contains the permanent errors recorded against this volume. R indicates read errors, W indicates write errors, and E indicates equipment checks, and U indicates unknown.*

5. **MB/ERR PERM**. *The error count and frequency for permanent read and write errors.*

6. **MB/ERR TEMP**. *The error count and frequency for temporary read and write errors.*

7. *Except for the CPU indicator, these columns contain information taken from the OBR record:*

CLNR ACTS	*cleaner action count*
ERASE GAPS	*erase gap count*
I/O CNT	*START I/O instruction count*
BLK LNG	*block length from the OBR record; or the average block length*
JOB NAME	*the jobname*
CPU	*as identified by EREP*
DENSITY	*tape density, from the mode byte from the OBR record*

8. **HDR SER**. *The serial number of the creating drive.*

9. *Totals for the volumes on this report.*

Threshold Summary Report

If you decide not to use the System Exception report series, you can get much of the same information for your supported tape drives using the tape Threshold Summary report. In fact, for some tape device types, you will have to use the Threshold Summary, because they are not included in the System Exception series. See Chapter 17, "Magnetic Tape Drives" on page 17-1 for the device types supported for the System Exception and Threshold Summary reports.

Useful parameters for customizing your Threshold Summary report are:

CUA
DATE
DEV
DEVSER
MODE
TIME
VOLID

Care should be taken when specifying report parameters other than these as report results could be misleading. See Figure 12-1 on page 12-2 in Part 3 for any restrictions on using the parameters in combination.

The Threshold Summary uses OBR (34XX) and MDR (8809) records to show all the permanent read/write errors, temporary read/write errors, and media statistics for each volume mounted. The data is grouped within each tape subsystem, which consists of a control unit and its drives.

The report has four sections, the first three of which appear once for each processor in your installation. The first section, **DEV(ice) STATISTICS**, shows one line of statistical and error data for every demount record whose error count exceeds the read or write threshold you coded on the report parameter.

The second section, **PERMANENT ERROR SUMMARY**, shows a one-line entry for each permanent error. A permanent error can be a read error, a write error, or an equipment check.

The third section, **TEMPORARY ERROR SUMMARY**, is a summary of all temporary errors recorded for each device number/CUA, whether they exceeded your threshold or not.

The fourth section, **VOLUME STATISTICS**, takes each MDR and OBR record from the first three sections of the report, and shows the errors and usage statistics by volume serial number rather than device number/CUA. Note that the columns in this part of the report are titled differently depending on the device type involved. See the report considerations under "34XX Tape Devices" in Part 4 for how they differ.

Figure 2-22 on page 2-60, Parts 1 and 2, shows the Threshold Summary report for tape devices. Following is the MVS JCL used to generate the report.

```
//THRESHLD EXEC PGM=IFCEREP1,PARM=(THRESHOLD=(001,001)',
//              'DATE=(82117)','ACC=N',HIST)
//TOURIST  DD   SYSOUT=R
//EREPPT   DD   SYSOUT=R
//DIRECTWK DD   UNIT=SYSDA,SPACE=(CYL,5,,CONTIG)
//ACCIN    DD   DSN=EREP.DAYLY.HISTORY,DISP=OLD
//SYSIN    DD   *
[SHARE statements for appropriate I/O devices]
/*
```

For more detailed information about the various parts of the report, see the maintenance documentation for the product involved.

```
                                   3 XXXXX 34XX/3803/8809 SUBSYSTEM SUMMARY XXXXX
                                     XXXXX     PRIMARY DEV    0570-057F        XXXXX
                 1 DEV STATISTICS - DEVS EQUAL TO OR EXCEEDING 001 TEMP RDS OR 001 TEMP WRTS

          TU    DATE    VOLUME      TIME       --TEMP--   IO   DEN- NRZI  R/W WR TG  LRC  CRC  ECC  SKEW ERLY VEL   ----CPU----   HDR
DEV    SERIAL DAY YR  SERIAL   HH MM SS.TH    RDS WRTS COUNT SITY NOISE VRC  VRC   MTE  EDC  ENV  ERR  BOR  CHG    ID  SERIAL   SER

570      N/A  102 82  72770    09 29 28.21      0  147  1120  N/A  N/A     0    0    0    0    0     0    0     0 3081 020015  8433
570      N/A  102 82  72770    09 29 32.13      0    2     9  N/A  N/A     0    0    2    0    0     0    0     0 3081 020015  8433
571      N/A  102 82  ZBB12F   11 20 56.13      0    1  1239  N/A  N/A     0    0    0    0    1     0    0     0 3081 020015  8460
570      N/A  103 82  T69842   10 01 30.68      0    1    27  N/A  N/A     0    0    0    0    0     0    0     0 3081 020015  8433
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

                                        PERMANENT ERROR SUMMARY

PW  PERMANENT WRITE                         PR  PERMANENT READ                          PF  CAUSE UNKNOWN
EC  EQUIPMENT CHECK, CAUSE UNKNOWN          EE  ERASE HEAD                              EB  TAPE BOTTOM, LEFT OR RIGHT
EL  LOAD FAILURE                            EP  AIR BEARING PRESSURE                    ET  TACH START FAILURE.
EV  VELOCITY CHECK                          ER  RESET KEY                               WC  WRITE CURRENT CHECK
EM  MODE SET
                                                                              ...SENSE BYTES....
                                                                                    1  1  1  1  1  1  1  1  1  1  2  2  2  2
 DEV SERIAL ERR VOLID  LAST CCW               STATUS     0  1  2  3  4  5  6  7  8  9  0  1  2  3  4  5  6  7  8  9  0  1  2  3
                       CC CA         FL CT    US CS CT

*570 88433  PW   72770  01 C0E608    64 7FF8 0E 00 0000 08 44 E6 D8 00 40 3D 00 00 98 00 00 00 2D C2 20 F1 91 70 E6 00 1A 00 09
*571 88460  EM          D3 000000    00 0000 06 00 0001 00 22 00 02 00 40 3D 00 00 08 00 00 00 2D C2 21 0C 91 70 E6 00 00 00 00
*576 76242  PR          02 185BF8    00 0050 0E 40 0034 08 C2 FF A8 00 44 3D 00 00 88 00 00 00 2D C2 18 62 91 60 A7 00 1A 00 08
*570 88433  PW  T69842  01 898000    60 0FD9 0E 00 0000 08 44 00 40 00 40 3D 00 00 08 00 00 00 2D C2 20 F1 91 70 E6 00 1A 00 09
*571 88460  EM  T72337  D3 000000    00 0000 06 00 0001 00 22 00 06 00 40 2D 00 00 08 00 00 00 2D C2 21 0C 91 70 E7 00 01 00 00
*571 88460  EM  T72337  D3 000000    00 0000 06 00 0001 00 22 00 06 00 40 2D 00 00 08 00 00 00 2D C2 21 0C 91 70 E7 00 00 00 00
*576 76242  EM  T77659  D3 000000    00 0000 06 00 0001 00 22 00 06 00 40 2D 00 00 08 00 00 00 2D C2 18 62 91 60 C3 00 15 00 00
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

                                   34XX/3803/8809 SUBSYSTEM TEMPORARY ERROR SUMMARY
2     ERRORS/100K                                            READ              WRITE          ECC
        IOS             DATE         TOTAL     TOTAL     STATISTICS        STATISTICS         VRC    STRD   PART    OVER   VEL   IBG
DEV   READ  WRITE  -FROM---TO--      IOS     MOUNTS.  ERRORS  CLNRAC .  ERRORS   ERSGAP  .  ENV    CHK    RECK    RUN   CHG   DET

570    0.00  100+  10282  10382     20077       8 .       0       4 .     150      251 .    0      0      0       0     0     0
571    0.00  8.16  10282  10382     12254       8 .       0       0 .       1        1 .    1      0      0       0     0     0
572    0.00  0.00  10382  10382      1237       1 .       0       0 .       0        0 .    0      0      0       0     0     0
575    0.00  0.00  09882  10382     58118      16 .       0       0 .       0        0 .    0      0      0       0     0     0
577    0.00  0.00  10382  10382      1455       2 .       0       0 .       0        0 .    0      0      0       0     0     0

TOTAL  0.00  100+                   93141      35         0               151

                                   3 XXXXX 34XX/3803/8809 SUBSYSTEM SUMMARY XXXXX
                                     XXXXX     PRIMARY DEV    0570-057F        XXXXX
                   DEV STATISTICS - DEVS EQUAL TO OR EXCEEDING 001 TEMP RDS OR 001 TEMP WRTS

          TU    DATE    VOLUME      TIME       --TEMP--   IO   DEN- NRZI  R/W WR TG  LRC  CRC  ECC  SKEW ERLY VEL   ----CPU----   HDR
DEV    SERIAL DAY YR  SERIAL   HH MM SS.TH    RDS WRTS COUNT SITY NOISE VRC  VRC   MTE  EDC  ENV  ERR  BOR  CHG    ID  SERIAL   SER

570      N/A  117 82  T74262   06 04 30.01      0    1  5513  N/A  N/A     0    0    0    0    1     1    0     0 3033 020862  1433
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

                                        PERMANENT ERROR SUMMARY

PW  PERMANENT WRITE                         PR  PERMANENT READ                          PF  CAUSE UNKNOWN
EC  EQUIPMENT CHECK, CAUSE UNKNOWN          EE  ERASE HEAD                              EB  TAPE BOTTOM, LEFT OR RIGHT
EL  LOAD FAILURE                            EP  AIR BEARING PRESSURE                    ET  TACH START FAILURE
EV  VELOCITY CHECK                          ER  RESET KEY                               WC  WRITE CURRENT CHECK
EM  MODE SET
                                                                              ...SENSE BYTES....
                                                                                    1  1  1  1  1  1  1  1  1  1  2  2  2  2
 DEV SERIAL ERR VOLID  LAST CCW               STATUS     0  1  2  3  4  5  6  7  8  9  0  1  2  3  4  5  6  7  8  9  0  1  2  3
                       CC CA         FL CT    US CS CT

 576 87930  EB  T74733  02 860BF8    20 0050 0E 00 0050 50 20 00 04 62 40 2D 20 00 08 00 00 00 6B 95 1E FA 01 70 74 0F 18 00 10
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

                                   34XX/3803/8809 SUBSYSTEM TEMPORARY ERROR SUMMARY
      ERRORS/100K                                            READ              WRITE          ECC
        IOS             DATE         TOTAL     TOTAL     STATISTICS        STATISTICS.        VRC    STRD   PART    OVER   VEL   IBG
DEV   READ  WRITE  -FROM---TO--      IOS     MOUNTS.  ERRORS  CLNRAC .  ERRORS   ERSGAP  .  ENV    CHK    RECK    RUN   CHG   DET

570    0.00 15.68  11782  11782      6374       3 .       0       0 .       1        1 .    1      0      0       0     0     0
571    0.00  0.00  11782  11782       784       1 .       0       0 .       0        0 .    0      0      0       0     0     0
574    0.00  0.00  11782  11782       784       1 .       0       0 .       0        0 .    0      0      0       0     0     0
577    0.00  0.00  11782  11782       418       1 .       0       0 .       0        0 .    0      0      0       0     0     0

TOTAL  0.00 11.96                    8360       6         0                 1
```

Figure 2-22 (Part 1 of 2). Threshold Summary Report for Tape

3 XXXXX 34XX/3803/8809 SUBSYSTEM SUMMARY XXXXX
XXXXX PRIMARY DEV 0570-057F XXXXX
NO DEVS EXCEEDED THRESHOLD: 82117 TO 82117

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

PERMANENT ERROR SUMMARY

NO PERMANENT ERRORS ENCOUNTERED: 82117 TO 82117

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

34XX/3803/8809 SUBSYSTEM TEMPORARY ERROR SUMMARY

	ERRORS/100K IOS		DATE				READ STATISTICS		WRITE STATISTICS		ECC					
DEV	READ	WRITE	-FROM-	--TO--	TOTAL IOS	TOTAL MOUNTS.	ERRORS	CLNRAC .	ERRORS	ERSGAP .	VRC ENV	STRD CHK	PART RECK	OVER RUN	VEL CHG	IBG DET
570	0.00	0.00	11782	11782	2359	2 .	0	0 .	0	0 .	0	0	0	0	0	0
571	0.00	0.00	11782	11782	132	1 .	0	0 .	0	0 .	0	0	0	0	0	0
576	0.00	0.00	11782	11782	2553	3 .	0	6 .	0	0 .	1	1	0	0	0	0
577	0.00	0.00	11782	11782	784	1 .	0	0 .	0	0 .	0	0	0	0	0	0
TOTAL	0.00	0.00			5828	7	0		0							

4

VOLUME STATISTICS - VOLUMES EQUAL TO OR EXCEEDING 001 TEMPORARY READS OR 001 TEMPORARY WRITES OR PERMANENT ERRORS

VOLUME SERIAL	DATE DAY	YR	TIME HH	MM	SS TH	DEV	TU SERIAL	RD/ WRT	--PERM-- RDS	WRTS	--TEMP-- RDS	WRTS	RD CLNR	RTRY/ ACT	ERASE GAPS	IO RDS	COUNT WRTS	BLOCK LENGTH	PROGRAM ID	----CPU---- ID	SERIAL	MOD #	DEN- SITY	HDR SER
	102	82	09	29	28.59	571	88460	R	0	0	0	0		0	0	0		0		3081	020015	8	6250	0
	102	82	14	42	10.91	576	76242	R	1	0	0	0		18	0	2		0	H44DER1B	3081	020015	8	6250	0
T69842	103	82	10	01	21.27	570	88433	W	0	1	0	1		0	26	27		1		3081	020015	8	6250	8433
T69842	103	82	10	01	30.68	570	10000		0	1	0	1		0	26	27		0	N/A	3081	020015	1	800	8433
T72337	103	82	10	55	54.46	571	88460	R	0	0	0	0		0	0	1		0	XAWEEK12	3081	020015	8	1600	8433
T72337	103	82	10	56	33.24	571	88460	R	0	0	0	0		0	0	1		0	XAWEEK12	3081	020015	8	1600	8433
T74262	117	82	06	04	30.01	570	10000		0	0	0	1		0	1	5513		0	REM74262	3033	020862	1	800	1433
T74733	117	82	10	16	17.22	576	87930	E	0	0	0	0		4	0	16		21412	RMFWK17	3033	020862	8	1600	7930
T77659	103	82	11	12	07.94	576	76242	R	0	0	0	0		0	0	0		0	D33SMY1A	3081	020015	8	1600	0
ZBB12F	102	82	11	20	56.13	571	10000		0	0	0	1		0	1	1239		0	D15BLD1H	3081	020015	1	800	8460
72770	102	82	09	27	28.15	570	88433	W	0	1	0	147		0	204	1120		2995	DPD83STL	3081	020015	8	6250	8433
72770	102	82	09	29	28.21	570	10000		0	1	0	147		0	204	1120		0	N/A	3081	020015	1	800	8433
72770	102	82	09	29	32.13	570	10000		0	0	0	2		0	21	9		0	DPD83STL	3081	020015	1	800	8433

Figure 2-22 (Part 2 of 2). Threshold Summary Report for Tape

Notes:

1. *DEV is the device number; same as the CUA.*
2. *An asterisk (*) before a device number indicates a 370-XA mode record.*
3. *The first three parts of this report are produced for each processor (CPU) involved.*
4. *The volume statistics summarize all the permanent errors presented in the preceding parts of the report.*

Detail Edit and Summary (PRINT) Reports

The PRINT report parameter gives you reports that allow you to look at the error records themselves, on two levels.

- **Detail Edits** format every record you have selected on a separate page, including a hexadecimal dump of the record itself.
- **Detail Summaries** summarize selected data from the record and total the number of records that met your selection criteria; some Detail Summaries show only the total number of selected records. EREP produces one Detail Summary per CPU (processor) for each record type selected.

The format and content of the Detail Edits and Summaries vary according to the type of record and the device or product involved. See the report examples that follow, and Figure 12-2 on page 12-4, for the record types and the devices associated with them.

Chapter 10, "Error Records for EREP" has more information about the records covered by each of these examples; look under the type of record indicated in the report header.

For some devices, the PRINT parameter produces **Data Reduction** reports that format and summarize environmental data gathered by the device itself. These reports are entirely device-specific and are meant to help the IBM Customer Engineer solve problems that are causing random/intermittent errors. Figure 2-26 on page 2-67 is an example of a Data Reduction report.

The PRINT parameter allows you to request more than one kind of Detail PRINT report at a time; it is the only report parameter that works this way. See the description of the PRINT report parameter in Part 3.

Limiting PRINT Output

An EREP run in which you requested PRINT reports without also being very selective of the records to be processed could generate a great deal of printed output. For example, when you code PRINT=PT without also using the date, time and type selection parameters, EREP produces a Detail Edit of every available record, regardless of type or when they were created. Coding PRINT=PS alone produces those same Detail Edits, plus Detail Summaries of every type of record EREP found in the input file.

If you do not want to see detailed reports for every error record on your ERDS or history data set, you must limit the PRINT reports by using the selection parameters. You can use PRINT reports to see, for example, summary information about a particular class of devices on a particular control unit; or detail information from a particular type of record as associated with a particular device number.

Every selection parameter except DEVSER is valid for use with the PRINT report parameter.

Examples

Figure 2-23 through Figure 2-52 are examples of Detail Edit and Summary reports for various kinds of error and operational records. Although it is unlikely that you would request all of these reports at once, it is possible to do so. In this case, the MVS JCL would look like this:

```
//DETAIL   EXEC  PGM=IFCEREP1,PARM='PRINT=AL,ACC=N,HIST'
//TOURIST  DD    SYSOUT=R
//EREPPT   DD    SYSOUT=R
//DIRECTWK DD    UNIT=SYSDA,SPACE=(CYL,5,,CONTIG)
//ACCIN    DD    DSN=EREP.DAYLY.HISTORY,DISP=OLD
//SYSIN    DD    *
[SHARE statements for appropriate I/O devices]
/*
```

The output would include many Detail Edit reports for each record type. More realistic examples of the code needed to produce the various Detail PRINT reports are in the sample EREP runs in Chapter 4, "Running EREP."

Not all the possible PRINT report combinations for each record type are shown in the following examples; the maintenance documentation for a particular device should include sample Detail Edit reports for the relevant records.

```
MODEL   9373                        SERIAL NO.   034762

 VS 2 REL.    02

--- RECORD SOURCE - CCH             TYPE - INBOARD

JOB NAME            TESTCCH

       DAY  YEAR                                HH MM SS.TH
DATE - 264   86                          TIME - 07 19 53 73

CHANNEL/UNIT ADDRESS  000700

             CC   DA     FL     CT
FAILING CCW  00 000000  00 00 0000

             K    CA     US CS  CT
CSW          00 000000  00 04 0000

 ---UNIT STATUS---                     ---CHANNEL STATUS---
  ATTENTION          0                  PRGM-CTLD IRPT    0
  STATUS MODIFIER    0                  INCORRECT LENGTH  0
  CONTROL UNIT END   0                  PROGRAM CHECK     0
  BUSY               0                  PROTECTION CHECK  0
  CHANNEL END        0                  CHAN DATA CHECK   0
  DEVICE END         0                  CHAN CTRL CHECK   1
  UNIT CHECK         0                  I/F CTRL CHECK    0
  UNIT EXCEPTION     0                  CHAINING CHECK    0

 *****************************************************************************************************

---LIMITED CHANNEL LOGOUT DATA EDITING---
 ---FIELD VALIDITY FLAGS---            ---TERMINATION CODE---
  SEQUENCE CODE STORED IS VALID     0   INTERFACE DISCONNECT       0
  UNIT STATUS STORED IS VALID       0   STOP, STACK OR NORMAL      0
  CCW ADDR AND KEY IN CSW ARE VALID 0   SELECTIVE RESET            0
  CHANNEL ADDRESS STORED IS VALID   0   INTERFACE INOPERATIVE      0
  DEVICE ADDRESS STORED IS VALID    0   ERROR ALERT                0

 ---SEQUENCE CODE---
  ERROR DETECTED DURING TEST I/O OR CLEAR I/O  1
  COMMAND WENT OUT, DEVICE STATUS NOT IN       0
  COMMAND ACCEPTED, NO DATA TRANSFERRED        0
  AT LEAST ONE DATA BYTE TRANSFERRED           0
  COMMAND EITHER NOT SENT OR NOT ACCEPTED      0
  COMMAND ACCEPTED BUT DATA XFER UNPREDICTABLE 0

 ---MEASUREMENT BYTE---

  BYTE:  00000000            NUMBER OF PENDING OPERATIONS (NPO):    000


 ---DELAY CODE---
  CHANNEL BUSY       0
  CONTROL UNIT BUSY  0
  DEVICE BUSY        0
```

Figure 2-23 (Part 1 of 2). CCH Detail Edit Report

```
I/O UNIT FOUND BUSY
 CHANNEL/UNIT ADDR     0740 0741 0742 0746

--- CHANNEL TYPE ---
  INTGTD MPX

*******************************************************************************
                           CHANNEL  ERROR  ANALYSIS
  CSW STORED BY  INTERRUPT
  TERMINATION BY  -- SYSTEM RESET- CODE 3
  TIME CHANNEL DETECTED ERROR -  COULD NOT BE ASSESSED
  VALIDITY OF RECORDED DATA
           COUNT              = NOT VALID
           SENSE DATA         = STORED
           UNIT STATUS        = NOT VALID
           COMMAND ADDRESS    = NOT VALID
           CHANNEL ADDRESS    = VALID
           DEVICE ADDRESS     = NOT VALID
  PROBABLE SOURCE OF ERROR-  CHANNEL
*******************************************************************************

   HEX DUMP OF RECORD
   HEADER    20660800  00000000  0086264F  07195373   00234567  93730000


       0000  E3C5E2E3   C3C3C800   07400741   07420746    00000000   00000000   00000000   00000000
       0020  00000000   00040000   444002C0   00000000    01000700   00000000   40000700   0004EC94
```

Figure 2-23 (Part 2 of 2). CCH Detail Edit Report

```
             MODEL 9373 CHANNEL CHECK RECORDS

                                                 DAY YEAR     DAY YEAR
                              DATE RANGE -  FROM 264  86  TO  268  86

                                                 SERIAL NO.          234567

                                                 NO.OF RECORDS        00005

                         --- SUMMARY OF MODEL 9373 CHANNEL CHECK RECORDS ---

                                                 ERROR SOURCE

                                                   CPU    0000
                                                   CHAN   0005
                                                   SCU    0000
                                                   SU     0000
                                                   CU     0000

              --- UNIT STATUS ---                                     --- CHANNEL STATUS ---
ATTENTION          0000   CHANNEL END       0000   PRGM-CTLD IRPT     0000   CHAN DATA CHECK    0000
STATUS MODIFIER    0000   DEVICE END        0000   INCORRECT LENGTH   0000   CHAN CTL CHECK     0005
CONTROL UNIT END   0000   UNIT CHECK        0000   PROGRAM CHECK      0000   I/F CTL CHECK      0000
BUSY               0000   UNIT EXCEPTION    0000   PROTECTION CHECK   0000   CHAINING CHECK     0000
```

Figure 2-24. CCH Detail Summary Report

```
DEVICE NUMBER:    0480    REPORT: CRW EDIT          DAY  YEAR  RECORDING MODULE: CRWEXMPL
                                              DATE: 169   81                        C3D9E6C5E7D4D7D3
DEVICE TYPE:      N/A     SCP: VS 2 REL.   3
                          CPU MODEL: 3081           HH MM SS.TH
CHANNEL PATH ID: 02       CPU SERIAL: 020024  TIME: 12 28 41.54
CHANNEL REPORT WORD INFORMATION
  CRW VALIDITY:     VALID, OVERFLOW INDICATED  1
  CRW:              F738 2214
  RECORDING CODE:   01
  ORIGIN:           SYSTEM DAMAGE MACHINE CHECK
  STORED BY:        HARDWARE
  CREATED BY:       HARDWARE
  PROCESSOR ADDR:   0002
CRW SEQUENCE NUMBER:                   63519980
ASSOCIATED CRW SEQUENCE NUMBER:        9A6F1187
INTERRUPT SUBCLASS DEFINITION TABLE:   73981674 F1EA8716
PATH MANAGEMENT CONTROL WORD  2
  SUBCHANNEL ENABLED            0
  PROG CHECK ADDR >= LIMIT      0
  PROG CHECKADDR <= LIMIT       1
  STORE MEASUREMENTS IN CMB     1
  STORE DCTI IN EXT STAT WORD   0
  DYNAM PATH MULTI-PATH STATE   1
  TIMING FACILITY AVAILABLE     0
  VALID DEVICE NUMBER ASSIGNED  0
UCB INFORMATION                                            CHANNEL PATH INFORMATION
  UCB LEVEL VALUE:             F8                            CHANNEL PATH RECOVERY COUNT:   F0  2
  UCB LEVEL BIT MASK:          1673F0F0
  SUBCHANNEL RECOVERY ANCHOR:  62751387                          3
------------------UCB DEVICE STATUS FLAGS---------------   ----CHPID ICHPT FLAGS-----
UCB TEMPORARILY UNUSABLE 0   INTRCEPT CNDITION EXISTS 0    CHP VALID FOR INSTLATION 1
DEVICE NOT READY         0   DVICE HAS NO USABLE PATH 0    CHP OWNED BY THIS SYSTEM 1
DEVICE SUBCHAN UNUSABLE  0   DEVICE HAS NO SUBCHANNEL 1    CHP IS ONLINE            1
PENDING SENSE OPERATION  1   ABNORMAL UCBLEVEL VALUE  1    CHP UNDRGOING CHP RCOVRY 1
START SUBCHANNEL ISSUED  0   RESERVED                 0    VARY OFF IN PROG FOR CHP 0
HALT SUBCHANNEL ISSUED   0   RESERVED                 1    FORCE CHP OFFLINE FAILED 0
CLEAR SUBCHANNEL ISSUED  1   RESERVED                 0    RECOVRY IN LAST UCB SCAN 0
DVICE OFFLN DUE TO ERROR 0   RESERVED                 0    RESERVED                 0
HEX DUMP OF RECORD
HEADER    25831000  00000000  0081169F  12284154  01020024  30810000
    0018  C3D9E6C5  E7D4D7D3  01800002  00020000  F7382214  0480FF00  63519980  9A6F1187
    0038  12341234  F0F000F8  1673F0F0  62751387  F0F07398  1674F1EA  8716
```

Figure 2-25. CRW Detail Edit Report (370-XA only)

Notes:

1. *These words appear if any CRW records are lost because they are being produced on the hardware queue faster than the recording service can retrieve them.*

2. *All zeros indicates that the UCB was not available.*

3. *The channel path table flags appear only if the CRW indicates a CHPID.*

```
*******************************************************************************

           MAINTENANCE DEVICE CODE FOR DEVICE TYPE = 3370
       DEVICE ADDRESS = 0701  SHARED      SERIAL =  700006
        1
     MD CODE TYPE =  DC1  MDC=0008 SAMPLES=   1
     MD CODE TYPE =  FC1  MDC=0200 SAMPLES=   1   2
     MD CODE TYPE =  SV   MDC=0130 SAMPLES=   2
MODIFER(S):  3
  EXPECTED      ACTUAL        ACCESS     EVEN   OVER/  DIFFERENCE CT
  DESTINATION   DESTINATION   DIRECTION  TRACK  UNDER  REMAINDER
  CCC-HH-M/F-SM CCC-HH-M/F-SM    F/R      E/O   OS/US  DIFF
    7  0  M   0    0 15  F   3    R        0    OS-  7    0
    7  0  M   0    0 15  F   3    R        0    OS-  7    0
       4
     MD CODE TYPE =  SVE  MDC=8130 SAMPLES=   1
MODIFER(S):
  EXPECTED      ACTUAL        ACCESS     EVEN   OVER/  DIFFERENCE CT
  DESTINATION   DESTINATION   DIRECTION  TRACK  UNDER  REMAINDER
  CCC-HH-M/F-SM CCC-HH-M/F-SM    F/R      E/O   OS/US  DIFF
    7  0  M   0    0 15  F   3    R        0    OS-  7    0
     MD CODE TYPE =  SC   MDC=0001 SAMPLES=   2
     MD CODE TYPE =  SCE  MDC=8001 SAMPLES=   1
     MD CODE TYPE =  RW   MDC=0132 SAMPLES=   6
     MD CODE TYPE =  DC   MDC=0300 SAMPLES=   1

 ECC CORRECTABLE            UNCORRECTABLE            NO SYNC BYTE FOUND
ALTERNATE DATA BLOCK            N/A                       N/A
CCC = 999 HH =  2 BB =  2

IFC1691      6 RECORDS NOT USED BY IFCNFPDR FOR THIS CUX 070X   5
```

Figure 2-26. Data Reduction Report

Notes:

1. *There are six different MDCs, each using a particular subset of fault symptom codes.*

2. *The number of records used to build this MDC.*

3. *Four of the MDCs have additional information printed.*

4. *An additional MDC is printed for records with only the environmental data bit on.*

5. *To build the MDC, only selected OBR (by fault code) records from a 3370 are used.*

```
--- RECORD ENTRY TYPE - DDR   SOURCE - DDR   MODEL -  3033   SERIAL NO. 025791

    VS 2 REL.    03
                              DAY YEAR   HH.MM.SS.TH   JOB IDENTITY DUMMYDDR

                              061  80    12 18 47 09             C4E4D4D4E8C4C4D9

 FROM UCB DEVICE TYPE        00001008                     TO UCB DEVICE TYPE         00001008
 FROM CHANNEL UNIT ADDRESS     000234                     TO CHANNEL UNIT ADDRESS      000236  1
 FROM VOLUME SERIAL NUMBER     234567                     TO VOLUME SERIAL NUMBER      765432
 FROM PHYSICAL ID                  01                     TO PHYSICAL ID                   02

 RECORD DEPENDENT SWITCH           10

 RECONFIGURATION PERFORMED AS A RESULT OF A PERMANENT ERROR


   HEX DUMP OF RECORD
   HEADER    60830810  00001100  0080061F  12184709   00025791  303302A0

       0000  C4E4D4D4  E8C4C4D9  F2F3F4F5  F6F7F7F6   F5F4F3F2  01000234  00001008  02000236
       0020  00001008
```

Figure 2-27. DDR Detail Edit Report

Notes:

1. *For records created in 370-XA mode, the device number (DEV) replaces CUA.*

```
SUMMARY OF DDR RECORDS                CUA 000234

                      DAY YEAR   DAY YEAR
RECORD DATE RANGE     061  80    062  80

MODEL - 3033     SERIAL NO - 025791

TOTAL NUMBER OF RECORDS=0002
```

Figure 2-28. DDR Detail Summary Report

```
                                  --- MACHINE CHECK DATA EDITING ---
***************************************************************************************************
MODEL=9373                      SERIAL NO=       010620

 VS 2 REL.    03

                                                                      DAY YEAR            HH MM SS
                                                               DATE - 211  86      TIME - 16 40 51
                             SM KS CM UA        IA
OLD MACHINE CHECK PSW        00 0C 00 0000      01C90A

JOB NAME=             TEG

PROGRAM NAME=         RMSTEG

                       *************************************************************************

NOTE: THE PRODUCT FUNCTIONAL CHARACTERISTICS PUBLICATION DESCRIBES THE MACHINE CHECK INTERRUPT CODE SUPPORT.

                         --- MACHINE CHECK INTERRUPT CODE ---

         --- SUB CLASS ---

SYSTEM DAMAGE (SD)      0                                CLOCK DAMAGE (CD)       0
PROC.DAMAGE (PD)        0                                WARNING (W)             0
SYSTEM RECOVERY (SR)    0                                DEGRADATION (DG)        0

                           --- INTERRUPT TENSE CODES ---

                       --- STORAGE AND PROTECTION ERROR CODES ---

UNCORRECTED STORAGE ERRORS (SE)     0                    KEY IN STOR ERR(KE)                    0
CORRECTED STORAGE ERRORS (SC)       0                    STOR DEGRADATION (DS)                  0

                                   --- PSW VALIDITY CODES ---

EMWP BITS OF M.C. OLD ARE VALID (WP)    1                SYSTEM MASK OF M.C. OLD IS VALID (MS)    1
PROGRAM MASK OF M.C. OLD IS VALID (PM)  1                INSTR ADDR OF M.C. OLD IS VALID (IA)     1

                                 --- MISC VALIDITY CODES ---

FAILING STORAGE ADDR IS VALID (FA)      0                INSTR MODIFIED STORAGE IS VALID (ST)     1
FP REGS STORED ARE VALID (FP)           1                GP REGS STORED ARE VALID (GP)            1
CONTROL REGS STORED ARE VALID (CR)      1                CLOCK COMPARATOR STORED IS VALID(CC)     1
REGION CODE IS VALID (RC)               0
EXTERNAL LOGOUT AREA IS VALID(CC)       0                EXTERNAL DAMAGE CODE IS VALID (EC)       1

EXTENDED LOGOUT LENGTH           0000                        FAILING STORAGE ADDRESS         00000000

                                 --- EXTERNAL DAMAGE CODE ---
EXTERNAL SECONDARY REPORT               1                CHANNEL NOT OPERATIONAL                  0
I/O INTERRUPT TIMEOUT                   1                I/O INSTRUCTION TIMEOUT                  0

                                    --- REGION CODE ---
DAMAGE DURING I/O INSTRUCTION           DEVICE           0000
                       *************************************************************************
```

Figure 2-29 (Part 1 of 2). MCH Detail Edit Report

```
--- FLOATING POINT REGISTERS ---
FP REGS 0,2    00 00 00 00   00 00 00 00   00 00 00 00   00 00 00 00
FP REGS 4,6    00 00 00 00   00 00 00 00   00 00 00 00   00 00 00 00

--- GENERAL PURPOSE REGISTERS ---
GP REGS 0-3    00 00 00 00   00 01 CE 10   00 00 00 00   00 01 D2 EC
GP REGS 4-7    40 01 C8 F4   80 01 D2 F0   00 3D 03 80   00 00 00 34
GP REGS 8-B    00 3D 03 80   00 3D 04 C8   07 0D 00 00   00 3C DC 98
GP REGS C-F    00 01 C3 A8   00 1F A1 A8   00 39 54 80   00 03 35 84

--- CONTROL REGISTERS ---
GP REGS 0-3    80 80 0E C0   09 3C E5 C0   FF FF FF FF   00 00 00 00
GP REGS 4-7    00 00 00 00   00 00 00 00   A3 3A 1F 48   00 00 00 00
GP REGS 8-B    00 00 00 00   00 00 00 00   00 00 00 00   00 00 00 00
GP REGS C-F    00 00 00 00   00 00 00 00   00 00 00 00   00 00 00 00

--- MACHINE CHECK LOGOUT BYTES ---
0000 04010F3D 00030000 00000000 22000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0030 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0060 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
0090 00000000 00000000 00000000 0001CE10 00000000 0001D2EC 4001C8F4 8001D2F0 003D0380 00000034 003D0380 003D04C8
00C0 070D0000 003CDC98 0001C3A8 001FA1A8 00395480 00033584 80800EC0 093CE5C0 FFFFFFFF 00000000 00000000 00000000
00F0 A33A1F48 00000000 00000000 00000000 00000000 00000000 00000000 00000000 CFC00000 00000000


   HEX DUMP OF RECORD
   HEADER     10660800 00000000 0086211F 16405155   00010620 93730000
       0018   D9D4E2E3 C5C74040 E3C5C740 40404040   000C0000 0001C90A 04010F3D 00030000
       0038   00000000 22000000 00000000 00000000   00000000 00000000 00000000 00000000
       0058   00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000
       0078   00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000
       0098   00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000
       00B8   00000000 00000000 00000000 00000000   00000000 0001CE10 00000000 0001D2EC
       00D8   4001C8F4 8001D2F0 003D0380 00000034   003D0380 003D04C8 070D0000 003CDC98
       00F8   0001C3A8 001FA1A8 00395480 00033584   80800EC0 093CE5C0 FFFFFFFF 00000000
       0118   00000000 00000000 A33A1F48 00000000   00000000 00000000 00000000 00000000
       0138   00000000 00000000 CFC00000 00000000   00580000 00000000 80000000 00000004
       0158   00000000 00000000 00000000 00000000   00000000 00000000 00000000 00000000
       0178   00000000 00000000 00000000 00000002   00000000 00000000 00000000 00000000
       0198   003D0380 0039BAB0
```

Figure 2-29 (Part 2 of 2). MCH Detail Edit Report

```
          MODEL 9373 MACHINE CHECK RECORDS     DAY YEAR     DAY YEAR
                           DATE RANGE -  FROM  211  86  TO  219  86
                                               SERIAL NO.         010620
                                               NO.OF RECORDS       00011

                        --- SUMMARY OF MODEL 9373 MACHINE CHECK RECORDS ---

                        --- MACHINE CHECK INTERRUPT CODE ---
                                  --- SUB CLASS ---
SYSTEM DAMAGE (SD)      0000                            CLOCK DAMAGE (CD)      0000
PROC.DAMAGE (PD)        0000                            EXTERNAL DAMAGE (ED)   0011
SYSTEM RECOVERY (SR)    0000                            AUTO-CONFIG (AC)       0000
TIMER DAMAGE (TD)       0000                            WARNING (W)            0000

                          --- INTERRUPT TENSE CODES ---
BACK-UP (B)             0000                            DELAYED (D)            0005

                       --- STORAGE AND PROTECTION ERROR CODES ---
UNCORRECTED STORAGE ERRORS (SE)    0000                 UNCORRECTED PROTECTION ERRORS (PE)    0000
CORRECTED STORAGE ERRORS (SC)      0000                 STORAGE DEGRADATION (DS)              0000
```

Figure 2-30. MCH Detail Summary Report

```
---RECORD ENTRY TYPE - NCP MDR     SOURCE - OUTBOARD                  MODEL- 3031     SERIAL NO.   037961
    VS 2 REL.    06
                      DAY YEAR                      HH MM SS.TH
                DATE- 063   80            TIME 08 47 22 10

 DEVICE TYPE          3705
 CHANNEL UNIT ADDRESS 051D
 RESOURCE I.D.        F850

RECORD TYPE - BSC/SS PERMANENT LINE ERROR
 LIB ADDR.            002C
 TERMINAL NAME        D004

  BASIC TRANSMISSION UNIT
   BTU COMMAND  02              IOB COMMAND        AC        IOB INITIAL ERROR STATUS   0680
   BTU MODIFIER 0A              IOB MODIFIERS      0000      IOB INITIAL ERR EXT STAT   00
   BTU FLAGS    1000            IOB IMMED CTL CMMND 00       IOB STATUS                 0680
                                                             IOB EXTENDED STATUS        00

   INITIAL ERROR STATUS 06    INITIAL ERR EXT STAT 00    LAST ERROR STATUS    06    LAST ERR EXT STAT    00
      FIRST BYTE                                            FIRST BYTE
   EXTENDED ERR STAT FLG 0    OVERRUN/UNDERRUN FLAG 0    EXTENDED ERR STAT FLG 0    OVERRUN/UNDERRUN FLAG 0
   FORMAT EXCEPTION FLAG 0    LINE QUIET TIMEOUT FG 0    FORMAT EXCEPTION FLAG 0    LINE QUIET TIMEOUT FG 0
   SYNC CHECK FLAG       0    LEADING DLE FORMAT CH 0    SYNC CHECK FLAG       0    LEADING DLE FORMAT CH 0
   DATA CHECK FLAG       0    SUB BLOCK ERROR FLAG  0    DATA CHECK FLAG       0    SUB BLOCK ERROR FLAG  0
   PH    ER              0    UNUSED                0    PH    ER              0    UNUSED                0
     AS    RO            1    UNUSED                0      AS    RO            1    UNUSED                0
       E     R           1    UNUSED                0        E     R           1    UNUSED                0
   LENGTH CHECK FLAG     0    UNUSED                0    LENGTH CHECK FLAG     0    UNUSED                0

  SIO COUNTER 0000  TEMPORARY ERROR COUNTER 00
  2770            00
   HEX DUMP OF RECORD
   HEADER    91460800    058A0000    0080063F    08472210       00037961    30310588
       C018  051DC4F0    F0F44040    4040F850    002C0005       020A1000    AC000000    06800006    80000000
       0038  00005087    4C
```

Figure 2-31. MDR Detail Edit Report

```
---SUMMARY OF ENTRY TYPE - 3705 MDR          DEVICE TYPE 3705          MODEL- 3031          SERIAL NO. 037961
               DAY YEAR     DAY YEAR
   DATE RANGE- 063   80 TO  063   80
    CHANNEL UNIT ADDRESS 00051D                    TOTAL NUMBER OF RECORDS  0021

                                                             - - - - - - - - PERMANENT ERROR TYPES - - - - - - - -
                 LIB                TEMP    PERM                                                                MODEM/
TERM NAME  RID   ADDR  # I/O OPS   ERRORS  ERRORS      HDWR   TM OUT DATA CK   RCV   ITV RQD   MISC   INTFC
A025       F854  002C  00000000    000000  000003  %%  00000   00003   00000   00000   00000   00000  00000
B020       F801  002C  00000000    000000  000002  %%  00000   00002   00000   00000   00000   00000  00000
B021       F80B  002C  00000000    000000  000001  %%  00000   00001   00000   00000   00000   00000  00000
B022       F819  002C  00000000    000000  000001  %%  00000   00001   00000   00000   00000   00000  00000
B023       F827  002C  00000000    000000  000001  %%  00000   00001   00000   00000   00000   00000  00000
B024       F835  002C  00000000    000000  000001  %%  00000   00001   00000   00000   00000   00000  00000
B025       F843  002C  00000000    000000  000001  %%  00000   00001   00000   00000   00000   00000  00000
B101       F84E  002C  00000000    000000  000005  %%  00000   00005   00000   00000   00000   00000  00000
D003       F84F  002C  00000000    000000  000003  %%  00000   00003   00000   00000   00000   00000  00000
D004       F850  002C  00000000    000000  000003  %%  00000   00003   00000   00000   00000   00000  00000
```

Figure 2-32. MDR Detail Summary Report

Note: *As of EREP 2.3, there is no Detail Edit or Summary of DASD MDR records. See "33XX DASD" in Part 4.*

```
--- RECORD ENTRY TYPE - MIH   SOURCE - MIH   MODEL -  3033   SERIAL NO. 025791

    VS 2 REL.    03
                              DAY YEAR   HH.MM.SS.TH   JOB IDENTITY DUMMYMIH

                              061  80    18 34 07 25               C4E4D4D4E8D4C9C8

 UCB DEVICE TYPE                  00001008
 PRIMARY   CHANNEL UNIT ADDRESS     000234
 ALTERNATE CHANNEL UNIT ADDRESS     000234
 CHANNEL SET ID                         01
 MISSING INTERRUPT                      40

 TIME INTERVAL                    00012345

 VOLUME SERIAL NUMBER               111111

   HEX DUMP OF RECORD
   HEADER    70830800  40011100  0080061F  18340725    00025791  30330000

       0000  C4E4D4D4  E8D4C9C8  0002340D  0234F1F1    F1F1F1F1  0000080D  F0F0F0F1  F2F3F4F5
```

Figure 2-33. MIH Detail Edit Report for 370-Mode Records

```
     SUMMARY OF MIH RECORDS               CUA 0004A1

                           DAY YEAR   DAY YEAR
     RECORD DATE RANGE     061  80    061  80

     MODEL - 3033     SERIAL NO - 021929

     TOTAL NUMBER OF RECORDS=0004
```

Figure 2-34. MIH Detail Summary Report for 370-Mode Records

```
DEVICE NUMBER:    04AC    REPORT: MIH EDIT              DAY YEAR   JOB IDENTITY: E17JAD1A
                          SCP:    VS 2 REL.  3    DATE: 112  82                  C5F1F7D1C1C4F1C1
DEVICE TYPE:      3330
                          CPU MODEL:  3081              HH MM SS.TH
CHANNEL PATH ID: N/A      CPU SERIAL: 220015      TIME: 10 03 03.98

MISSING INTERRUPT: 10 - START PENDING IN SUBCHANNEL      SUBCHANNEL ID NUMBER: 0000101DC
                                                         VOLUME SERIAL:        POK087
                    HH MM SS.TH                          UCB LEVEL BYTE:       01
TIME INTERVAL:      00 00 15.00

RECOVERY ACTIONS PERFORMED BYTE: AC   1

   HALT OR CLEAR SUBCHANNEL  1
   SIMULATED INTERRUPT       0
   REDRIVE DEVICE            1
   REQUEUE I/O REQUEST       0
   ISSUE MESSAGE             1
   LOG THE CONDITION         1
   BIT 6                     0
   BIT 7                     0

   HEX DUMP OF SUBCHANNEL INFORMATION BLOCK
   OFFSET         00F96710  289B04AC  A00020F0  01C5BFF0
           0010   04241116  FFFFFFFF  00000000  00004400
           0020   00D005B0  10000000  6B1F3001  292711F7
           0030   00000935

   HEX DUMP OF RECORD
   HEADER    71831800  00000000  0082112F  10030398  03220015  30810000
       0018  C5F1F7D1  C1C4F1C1  00F96710  289B04AC  A00020F0  01C5BFF0  04241116  FFFFFFFF
       0038  00000000  00004400  00D005B0  10000000  6B1F3001  292711F7  00000935  F0F0F0F0
       0058  F1F5F0F0  10BCBCAC  000101DC  2898A020  F0042411  16FFFFFF  FF010000  00000100
       0078  800004AC  08003070  200DD7D6  D2F0F8F7  00000000
```

Figure 2-35. MIH Detail Edit Report for 370-XA Records

Notes:

1. *The hexadecimal value in the byte is shown here; the bit settings are detailed below it.*

```
DEVICE NUMBER:    07CA       REPORT:  MIH SUMMARY             REPORT DATE:  118  82
                                                              PERIOD FROM:  112  82
DEVICE TYPE:      3330       CPU MODEL:  3081                         TO:   112  82
                             CPU SERIAL: 020015

MISSING INTERRUPT

  MISSING CSCH                 00000000
  MISSING HSCH                 00000000
  IDLE DEVICE WITH WORK QUEUED 00000000
  START PENDING IN SUBCHANNEL  00000018
  MOUNT PENDING                00000000
  MISSING PRIMARY STATUS       00000000
  MISSING SECONDARY STATUS     00000000
```

Figure 2-36. MIH Detail Summary Report for 370-XA Records

```
PRIMARY CUA:      0C70              REPORT:  OUTBOARD (LONG)          DAY YEAR    JOB IDENTITY: MAINT
                                      SCP:     VS 2 REL.  3      DATE: 289  86                  D4C1C9D5E3404040
DEVICE TYPE:      9347
                                      MODEL:    9375                  HH MM SS.TH
ERROR PATH:       0C70                SERIAL:  234567            TIME: 05 35 04.82

RECORD IS:        PERMANENT

MODE IS:          370

                  CC   CA    FL     CT
FAILING CCW:      01 304F0B 20 80 00A5

                  K    CA    US CS  CT
CSW:              00 6A8590 0E 00 00A5

---UNIT STATUS----  CHANNEL STATUS
ATTENTION         0  PGM-CTLD IRPT    0
STATUS MODIFIER   0  INCORRECT LENGTH 0
CONTROL UNIT END 0   PROGRAM CHECK    0
BUSY              0  PROTECTION CHECK 0
CHANNEL END       1  CHAN DATA CHECK  0
DEVICE END        1  CHAN CTL CHECK   0
UNIT CHECK        1  I/F CTL CHECK    0
UNIT EXCEPTION    0  CHAINING CHECK   0


DEVICE DEPENDENT DATA

SYMPTOM CODE    2002
VOLUME SERIAL   HISTOT


SENSE BYTE DATA

------BYTE00----08 ------BYTE01----44 ------BYTE02----0C ------BYTE03----07 ------BYTE04----10 -----BYTE05----00
COMMAND REJECT   0 NOISE            0 --------------   0 --------------   0 NOT USED         0 NOT USED        0
INTRVN REQUIRED  0 DEVICE STATUS A  1 ]              ] 0 ]              ] 0 NOT USED         0 NOT USED        0
NOT USED         0 DEVICE STATUS B  0 ]              ] 0 ]  ERROR       ] 0 TAPE INDICATE    0 NOT USED        0
EQUIPMENT CHECK  0 NOT USED         0 ] TRACK IN     ] 0 ]  RECOVERY    ] 0 PERMANENT ERROR  1 PE-ID CHECK     0
DATA CHECK       1 AT LOAD POINT    0 ] ERR (0-7)    ] 1 ]  PROCEDURE ] 0 HOST DETECT ERR  0 NOT USED        0
OVERRUN          0 WRITE STATUS     1 ]              ] 1 ]  (0-7)       ] 1 LOOP WRITE-READ  0 NOT USED        0
NOT USED         0 FILE PROTECT     0 ]              ] 0 ]              ] 1 NOT USED         0 NOT USED        0
NOT USED         0 NOT CAPABLE      0 --------------   0 --------------   1 NOT USED         0 NOT USED        0

------BYTE06----00 ------BYTE07----10 ------BYTE08----38 ------BYTE09----38 ------BYTE10----00 -----BYTE11----00
NOT USED         0 FORMAT CODE 8    0 BUFFER FULL LOW  0 BOT              0 DFCI SEQ CHECK   0 DOOR OPENED     0
NOT USED         0 FORMAT CODE 4    0 BUFFER FULL HIGH 0 EOT              0 DFCI PARITY CHK  0 REEL MISSING    0
NOT USED         0 FORMAT CODE 2    0 DRIVE ONLINE     1 TAPE-IN PATH SNR 1 SYNCH INS/OUTS   0 REEL INVERTED   0
NOT USED         0 FORMAT CODE 1    1 DRIVE READY      1 WRITE ENABLED    1 XFER FAILURE     0 NO BOT          0
NOT USED         0 DATA SECUR ERASE 0 POS. TO MOVE FWD 1 PE 1600 ID BURST 1 NOT USED         0 LOAD FAILURE    0
NOT USED         0 NOT USED         0 NOT USED         0 PE 3200 ID BURST 0 NOT USED         0 REEL NOT CNTRED 0
NOT USED         0 NOT USED         0 NOT USED         0 NOT USED         0 NOT USED         0 NOT USED        0
NOT USED         0 NOT USED         0 NOT USED         0 NOT USED         0 NOT USED         0 P.O.S.T         0
```

Figure 2-37 (Part 1 of 2). OBR Detail Edit Report for 370 Mode Records

```
------BYTE12----00 ------BYTE13----C0 ------BYTE14----00 ------BYTE15----80 ------BYTE16----80 -----BYTE17----
TENSION ARM      0 WRITE CHECK      1 READ SKEW        0 TIE PARITY       1 1600 CPI/25IPS   1 NOT USED
TAPE SPEED       0 WRITE IBG NOISE  1 READ UNCORRECT P 0 NOT USED         0 1600 CPI/100IPS  0 NOT USED
3700 FT OF TAPE  0 WRITE ID CHECK   0 READ MCHNL DROP  0 NOT USED         0 3200 CPI/50IPS   0 IBG 32
TENSION ARM VOL  0 WRT POSTAMBLE CK 0 READ ID          0 NOT USED         0 NOT USED         0 IBG 16
TACHOMETER       0 ERASE GAP SIZE   0 NOT USED         0 NOT USED         0 NOT USED         0 IBG  8
SUPPLY HUB LOCK  0 PIC ERROR        0 NOT USED         0 NOT USED         0 NOT USED         0 IBG  4
TAKE-UP HUB SLIP 0 NOT USED         0 NOT USED         0 NOT USED         0 NOT USED         0 IBG  2
SUPPLY HUB SLIP  0 NOT USED         0 NOT USED         0 NOT USED         0 NOT USED         0 IBG  1

------BYTE18----04 ------BYTE19----00 ------BYTE20----00 ------BYTE21----00 ------BYTE22----00 -----BYTE23----
BLK LENGTH 32768 0 BLK LENGTH 128   0 1ST LVL IND BIT1 0 2ND LVL IND BIT1 0 3RD LVL IND BIT1 0 4TH LVL IND BT1
BLK LENGTH 16384 0 BLK LENGTH 64    0 1ST LVL IND BIT2 0 2ND LVL IND BIT2 0 3RD LVL IND BIT2 0 4TH LVL IND BT2
BLK LENGTH 8192  0 BLK LENGTH 32    0 1ST LVL IND BIT3 0 2ND LVL IND BIT3 0 3RD LVL IND BIT3 0 4TH LVL IND BT3
BLK LENGTH 4096  0 BLK LENGTH 16    0 1ST LVL IND BIT4 0 2ND LVL IND BIT4 0 3RD LVL IND BIT4 0 4TH LVL IND BT4
BLK LENGTH 2048  0 BLK LENGTH 8     0 1ST LVL IND BIT5 0 2ND LVL IND BIT5 0 3RD LVL IND BIT5 0 4TH LVL IND BT5
BLK LENGTH 1024  1 BLK LENGTH 4     0 NOT USED         0 NOT USED         0 NOT USED         0 NOT USED
BLK LENGTH 512   0 BLK LENGTH 2     0 NOT USED         0 NOT USED         0 NOT USED         0 NOT USED
BLK LENGTH 256   0 BLK LENGTH 1     0 NOT USED         0 NOT USED         0 NOT USED         0 NOT USED

------BYTE24----00 ------BYTE25----00 ------BYTE26----00 ------BYTE27----00 ------BYTE28----00 -----BYTE29----
5TH LVL IND BIT1 0 6TH LVL IND BIT1 0 NOT USED         0 --------------   0 --------------   0 NOT USED
5TH LVL IND BIT2 0 6TH LVL IND BIT2 0 NOT USED         0 ]            ]   0 ]            ]   0 NOT USED
5TH LVL IND BIT3 0 6TH LVL IND BIT3 0 NOT USED         0 ]            ]   0 ]            ]   0 NOT USED
5TH LVL IND BIT4 0 6TH LVL IND BIT4 0 NOT USED         0 ] CONDITION  ]   0 ] DIAGNOSTIC ]   0 NOT USED
5TH LVL IND BIT5 0 6TH LVL IND BIT5 0 NOT USED         0 ]   FLAG     ]   0 ]    LED     ]   0 NOT USED
NOT USED         0 NOT USED         0 NOT USED         0 ]            ]   0 ]            ]   0 NOT USED
NOT USED         0 NOT USED         0 NOT USED         0 ]            ]   0 ]            ]   0 NOT USED
NOT USED         0 NOT USED         0 NOT USED         0 --------------   0 --------------   0 NOT USED

------BYTE30----20 ------BYTE31----02
--------------   0 --------------   0
]            ]   0 ]            ]   0
]  FAULT     ]   1 ]  FAULT     ]   0
]  SYMPTOM   ]   0 ]  SYMPTOM   ]   0
]  CODE      ]   0 ]  CODE      ]   0
]  (MSB)     ]   0 ]  (LSB)     ]   0
]            ]   0 ]            ]   1
--------------   0 --------------   0

HEX DUMP OF RECORD

HEADER    30660800  00000000  0086289F  05350482  10234567  93750000
    0018  D4C1C9D5  E3404040  01304F0B  208000A5  006A8590  0E0000A5  01000C70  00008009
    0038  00000C70  00000020  C8C9E2E3  D6E30000  08440C07  10000010  38380000  00C00080
    0058  80000400  00000000  00000000  00002002  00000000  00000000  00000000  00000000
    0078  00000000  00000000  00000000
```

Figure 2-37 (Part 2 of 2). OBR Detail Edit Report for 370 Mode Records

```
DEVICE NUMBER:   000C70              REPORT:  OUTBOARD (LONG)        DAY YEAR     JOB IDENTITY: MAINT
                                     SCP:     VS 2 REL.  3     DATE: 289  86                   D4C1C9D5E3404040
DEVICE TYPE:     9347
                                     MODEL:   9375                   HH MM SS.TH
ERROR PATH:      00-0C70             SERIAL:  234567           TIME: 05 35 04.82
RECORD IS:       PERMANENT
MODE IS:         370XA
                   CC   CA    FL      CT
FAILING CCW:       01 304F0B 20 80 00A5
                   K  FLAGS  CA       US CS  CT
SCSW:              00 000000 00000000 00 00 0000

---UNIT STATUS----  SUB-CHANNEL STATUS  --------------------------SCSW FLAGS-----------------------
ATTENTION         0  PGM-CTLD IRPT    0  CCW FORMAT       0  RESERVED          0  SUBCHANNEL ACTIV 0
STATUS MODIFIER   0  INCORRECT LENGTH 0  PRE-FETCH CCW    0  SSCH FUNCTION     0  DEVICE ACTIVE    0
CONTROL UNIT END 0  PROGRAM CHECK    0  INIT STATUS      0  HSCH FUNCTION     0  SUSPENDED        0
BUSY              0  PROTECTION CHECK 0  ADDR LIMIT       0  CSCH FUNCTION     0  ALERT STATUS     0
CHANNEL END       0  CHAN DATA CHECK  0  SUPP SUSPEND INT 0  RESUME PENDING    0  INTERMED STATUS  0
DEVICE END        0  CHAN CTL CHECK   0  ZERO COND CODE   0  START PENDING     0  PRIMARY STATUS   0
UNIT CHECK        0  I/F CTL CHECK    0  EXTENDED CONTROL 0  HALT PENDING      0  SECONDARY STATUS 0
UNIT EXCEPTION    0  CHAINING CHECK   0  PATH NOT OPER    0  CLEAR PENDING     0  STATUS PENDING   0

DEVICE DEPENDENT DATA
SYMPTOM CODE   2002
VOLUME SERIAL  HISTOT


SENSE BYTE DATA

------BYTE00----08 ------BYTE01----44 ------BYTE02----0C ------BYTE03----07 ------BYTE04----10 -----BYTE05----00
COMMAND REJECT   0 NOISE            0 --------------   0 --------------   0 NOT USED         0 NOT USED         0
INTRVN REQUIRED  0 DEVICE STATUS A  1 ]             ]  0 ]             ]  0 NOT USED         0 NOT USED         0
NOT USED         0 DEVICE STATUS B  0 ]             ]  0 ]  ERROR      ]  0 TAPE INDICATE    0 NOT USED         0
EQUIPMENT CHECK  0 NOT USED         0 ] TRACK IN    ]  0 ]  RECOVERY   ]  0 PERMANENT ERROR  1 PE-ID CHECK      0
DATA CHECK       1 AT LOAD POINT    0 ] ERR (0-7)   ]  1 ]  PROCEDURE  ]  0 HOST DETECT ERR  0 NOT USED         0
OVERRUN          0 WRITE STATUS     1 ]             ]  1 ]  (0-7)      ]  1 LOOP WRITE-READ  0 NOT USED         0
NOT USED         0 FILE PROTECT     0 ]             ]  0 ]             ]  1 NOT USED         0 NOT USED         0
NOT USED         0 NOT CAPABLE      0 --------------   0 --------------   1 NOT USED         0 NOT USED         0

------BYTE06----00 ------BYTE07----10 ------BYTE08----38 ------BYTE09----38 ------BYTE10----00 -----BYTE11----00
NOT USED         0 FORMAT CODE 8    0 BUFFER FULL LOW  0 BOT              0 DFCI SEQ CHECK   0 DOOR OPENED      0
NOT USED         0 FORMAT CODE 4    0 BUFFER FULL HIGH 0 EOT              0 DFCI PARITY CHK  0 REEL MISSING     0
NOT USED         0 FORMAT CODE 2    0 DRIVE ONLINE     1 TAPE-IN PATH SNR 1 SYNCH INS/OUTS   0 REEL INVERTED    0
NOT USED         0 FORMAT CODE 1    1 DRIVE READY      1 WRITE ENABLED    1 XFER FAILURE     0 NO BOT           0
NOT USED         0 DATA SECUR ERASE 0 POS. TO MOVE FWD 1 PE 1600 ID BURST 1 NOT USED         0 LOAD FAILURE     0
NOT USED         0 NOT USED         0 NOT USED         0 PE 3200 ID BURST 0 NOT USED         0 REEL NOT CNTRED 0
NOT USED         0 NOT USED         0 NOT USED         0 NOT USED         0 NOT USED         0 NOT USED         0
NOT USED         0 NOT USED         0 NOT USED         0 NOT USED         0 NOT USED         0 P.O.S.T          0
```

Figure 2-38 (Part 1 of 2). OBR Detail Edit Report for 370-XA Mode Records

```
------BYTE12----00 ------BYTE13----C0 ------BYTE14----00 ------BYTE15----80 ------BYTE16----80 -----BYTE17----00
TENSION ARM      0 WRITE CHECK      1 READ SKEW        0 TIE PARITY       1 1600 CPI/25IPS   1 NOT USED        0
TAPE SPEED       0 WRITE IBG NOISE  1 READ UNCORRECT P 0 NOT USED         0 1600 CPI/100IPS  0 NOT USED        0
3700 FT OF TAPE  0 WRITE ID CHECK   0 READ MCHNL DROP  0 NOT USED         0 3200 CPI/50IPS   0 IBG 32          0
TENSION ARM VOL  0 WRT POSTAMBLE CK 0 READ ID          0 NOT USED         0 NOT USED         0 IBG 16          0
TACHOMETER       0 ERASE GAP SIZE   0 NOT USED         0 NOT USED         0 NOT USED         0 IBG  8          0
SUPPLY HUB LOCK  0 PIC ERROR        0 NOT USED         0 NOT USED         0 NOT USED         0 IBG  4          0
TAKE-UP HUB SLIP 0 NOT USED         0 NOT USED         0 NOT USED         0 NOT USED         0 IBG  2          0
SUPPLY HUB SLIP  0 NOT USED         0 NOT USED         0 NOT USED         0 NOT USED         0 IBG  1          0

------BYTE18----04 ------BYTE19----00 ------BYTE20----00 ------BYTE21----00 ------BYTE22----00 -----BYTE23----00
BLK LENGTH 32768 0 BLK LENGTH 128   0 1ST LVL IND BIT1 0 2ND LVL IND BIT1 0 3RD LVL IND BIT1 0 4TH LVL IND BT1 0
BLK LENGTH 16384 0 BLK LENGTH 64    0 1ST LVL IND BIT2 0 2ND LVL IND BIT2 0 3RD LVL IND BIT2 0 4TH LVL IND BT2 0
BLK LENGTH 8192  0 BLK LENGTH 32    0 1ST LVL IND BIT3 0 2ND LVL IND BIT3 0 3RD LVL IND BIT3 0 4TH LVL IND BT3 0
BLK LENGTH 4096  0 BLK LENGTH 16    0 1ST LVL IND BIT4 0 2ND LVL IND BIT4 0 3RD LVL IND BIT4 0 4TH LVL IND BT4 0
BLK LENGTH 2048  0 BLK LENGTH 8     0 1ST LVL IND BIT5 0 2ND LVL IND BIT5 0 3RD LVL IND BIT5 0 4TH LVL IND BT5 0
BLK LENGTH 1024  1 BLK LENGTH 4     0 NOT USED         0 NOT USED         0 NOT USED         0 NOT USED        0
BLK LENGTH 512   0 BLK LENGTH 2     0 NOT USED         0 NOT USED         0 NOT USED         0 NOT USED        0
BLK LENGTH 256   0 BLK LENGTH 1     0 NOT USED         0 NOT USED         0 NOT USED         0 NOT USED        0

------BYTE24----00 ------BYTE25----00 ------BYTE26----00 ------BYTE27----00 ------BYTE28----00 -----BYTE29----00
5TH LVL IND BIT1 0 6TH LVL IND BIT1 0 NOT USED         0 --------------   0 --------------   0 NOT USED        0
5TH LVL IND BIT2 0 6TH LVL IND BIT2 0 NOT USED         0 ]            ]   0 ]            ]   0 NOT USED        0
5TH LVL IND BIT3 0 6TH LVL IND BIT3 0 NOT USED         0 ]            ]   0 ]            ]   0 NOT USED        0
5TH LVL IND BIT4 0 6TH LVL IND BIT4 0 NOT USED         0 ] CONDITION  ]   0 ] DIAGNOSTIC ]   0 NOT USED        0
5TH LVL IND BIT5 0 6TH LVL IND BIT5 0 NOT USED         0 ]   FLAG     ]   0 ]    LED     ]   0 NOT USED        0
NOT USED         0 NOT USED         0 NOT USED         0 ]            ]   0 ]            ]   0 NOT USED        0
NOT USED         0 NOT USED         0 NOT USED         0 ]            ]   0 ]            ]   0 NOT USED        0
NOT USED         0 NOT USED         0 NOT USED         0 --------------   0 --------------   0 NOT USED        0

------BYTE30----20 ------BYTE31----02
--------------   0 --------------   0
]            ]   0 ]            ]   0
]  FAULT     ]   1 ]  FAULT     ]   0
]  SYMPTOM   ]   0 ]  SYMPTOM   ]   0
]  CODE      ]   0 ]  CODE      ]   0
]  (MSB)     ]   0 ]  (LSB)     ]   0
]            ]   0 ]            ]   1
--------------   0 --------------   0

HEX DUMP OF RECORD

HEADER    30661800  00000000  0086289F  05350482  10234567  93750000
    0018  D4C1C9D5  E3404040  01304F0B  208000A5  006A8590  0E0000A5  01000C70  00008009
    0038  00000C70  00000020  C8C9E2E3  D6E30000  08440C07  10000010  38380000  00C00080
    0058  80000400  00000000  00000000  00002002  00000000  00000000  00000000  00000000
    0078  00000000  00000000  00000000
```

Figure 2-38 (Part 2 of 2). OBR Detail Edit Report for 370-XA Mode Records

```
DEVICE NUMBER:       000002        REPORT:  OUTBOARD SUMMARY          REPORT DATE:  042 87
                                   MODEL:   3081                      PERIOD FROM:  087 84
DEVICE TYPE:         3211          SERIAL:  020447                            TO:   159 84
TOTAL NUMBER OF RECORDS      041   TOTAL OF OVERFLOW RECORDS 000
CCW COMMAND CODES ENCOUNTERED(MAXIMUM OF 24)
CMND  TOTAL
0B     041

STATISTICAL DATA SUMMARY
TEMPORARY READS  000  TEMPORARY WRITES 000
NOT USED         000  BUS OUT PAR CHK  000
EQUIPMENT CHECK  000  BUFR PARITY CHK  013
LOAD CHECK       045  NOT USED         000
COMMAND RETRY    000  PRINT CHECK      000
PRINT QUALITY    000  LINE POSITION    000
NOT USED         000  CMD SUPPRESS     045
NOT USED         000  CHAN DATA CHECK  000
NOT USED         000  NOT USED         000
NOT USED         000  NOT USED         000

SENSE BIT DATA SUMMARY
------BYTE 00----- ------BYTE 01---- ------BYTE 02----- ------BYTE 03----- -----BYTE 04----- -----BYTE 05-----
COMMAND REJECT  00 COMMAND RETRY  00 CARRIAGE F MOV  00 UCB PARITY      00 EXTRA SCAN      00 BIT 0 NOT USED 00
INTRVNTN REQ'D  00 PRINT CHECK    00 CARRIAGE SQ CHK 00 PLB PARITY      00 PRINT TRIG      00 BIT 1 NOT USED 00
BUS OUT PAR CHK 00 PRINT QUALITY  00 CARRIAGE SP CHK 00 FCB PARITY      00 BLK PRINT       00 BIT 2 NOT USED 00
EQUIPMENT CHECK 00 LINE POSITION  00 PLT F ADVANCE   04 COIL PROTECTION 00 PLAT INTERLOCK 00 BIT 3 NOT USED 00
DATA CHECK      00 FORMS CHECK    41 PLT F RETRY     00 H. FIRE CHK     00 CARRIAGE GO     00 BIT 4 NOT USED 00
BUFR PARITY CHK 09 CMD SUPPRESS   41 FORMS JAM       00 SERVICE AID     00 CARRIAGE SETTL 00 BIT 5 NOT USED 00
LOAD CHECK      41 MECH MOTION    14 RIBBON MOTION   00 UCSAR S C       00 CARRIAGE COMPR 00 BIT 6 NOT USED 00
CHANNEL NINE(9) 41 BIT 7 NOT USED 00 TRAIN OVERLOAD  00 PSE S CHECK     00 FORWARD DRIVE   00 BIT 7 NOT USED 00
```

Figure 2-39. OBR Detail Summary for 370 Mode Records

```
SUMMARY OF I/O RECORDS  TYPE-OBR/SDR/MDR  SOURCE-OUTBOARD/MISC  DEVICE 3800 MOD 3,8  CPU MODEL- 0158  SERIAL NO. 024706
            DAY YEAR    DAY YEAR
DATE RANGE  001  82 TO  032   2
                                                  NO. OBR SHORT RECORDS    0004
                                                  NO. OBR LONG RECORDS     0024
                                                  NO. MDR RECORDS          0003
                                                                          ------
CHANNEL UNIT ADDRESS/DEVICE NUMBER 0190           TOTAL NUMBER OF RECORDS  0031

--SUMMARY BY ERROR TYPE (COUNTS IN DECIMAL) --

                 SDR COUNTERS 1                          LONG OBR DATA 2                          MDR RECORDS 3
TEMP CHL DATA CK  0065  CFS MISFOLD (32)  0065  PERM CHL DATA CK  0000  INTVN RQD CK       0024  NO. INT LOG ENTRY 0024
TEMP CHL CTL CK   0065  BUR/TRIM JAM(40)  0065  PERM CHL CTL CK   0000  EQUIP CK           0024
TEMP INTF CTL CK  0065  NO BURST CK (41)  0065  PERM INTF CTL CK  0000  TEMP BOPAR          0000
                        BUR/STKR JAM(42)  0065                          PERM BOPAR         0001

--SUMMARY OF PERMANENT ERRORS FROM OBR RECORDS BY STATUS CODE (SENSE BYTE 4) - COUNTS IN DECIMAL

11 XFR UNDETENTED 0001  33 DATA WIDTH CK  0000  73 RPG SHIFTER CK 0000  95 POST XFR CRNA  0000  B3 PLTN OVTEMP     0000
14 XFR ST/SP CK   0001  34 FSR OUTPUT CK  0000  74 STRIP BUFFR CK 0000  96 LSR PWR SUPPLY 0000  B4 PLN THRM OPEN   0000
15 XFR MISREG/JAM 0000  3B X/F PG CT CK   0000  75 CG XEQ CS CK   0000  97 MIRROR DR CK   0000  B5 HR THRMSTR OPN 0000
16 XFR ENCODER CK 0000  43 EARLY BURST CK 0001  76 LASER POWER CK 0000  98 DVM CHECK      0000  B6 FSR CURRENT CK 0000
17 XFR MTR OVRLD  0000  4B BTS LOOP CK    0001  77 MIRROR SPD CK  0000                        B7 THERMAL- NO CK 0000
18 XFR PRT POS CK 0000                          78 SERIALIZER CK  0000  A0 PRNT PWR NRDY  0001  B8 NVS CHECK       0000
1C XFR TRACTOR CK 0000  51 MISSING FO FLH 0001  79 SRLZR INTRF CK 0000                        B9 PROC PAPER CK  0000
1E X/F LOOP CK    0000  52 EXTRA FO FLASK 0001  7A MIRR ROTATE CK 0000  A2 PROC VOLT CP   0000  BA SYS CHNL CK    0000

                        62 PRINT CONTRAST 0001  7C SER SYNCH CK   0000  A3 LOGIC VOLT CP  0000  BB DISK FILE CK   0000
21 FSR TEMP CK    0001  63 VACUUM SYS CK  0001  7D S/B OVER-RUN   0000  A4 MIRROR MTR TH  0000  BC FILE READ CK   0000
22 PLATN TEMP CK  0001  64 OPT STP LMT CK 0000  7E CG CS START CK 0000  A5 DR COOLR CHECK 0000  BD FILE WRITE CK  0000
23 FSR BUR NCLOSD 0000  65 CLNR BRUSH CK  0000  7F CG CS CMPLT CK 0000  A6 CYC BLWR MT TH 0000  BF DISK DAMAGED   0000
24 FSR BUR NOPEN  0000  66 ERASE LAMP CK  0000  80 RETRY G LOG FL 0001  A7 DEV MOTOR THRM 0000  D0 EXGRF-CGEN CK  0001
25 FSR PRT ALGNMT 0000  67 MARK SENSOR CK 0000  89 PERM IEU PE    0001  A8 CLNR BR MTR TH 0000  D1 X/G CPS CHECK  0001
26 FSR WIDTH CK   0000  68 DRUM SLOW      0000  8B SUBSYS CLK CK  0000  A9 CTL ASM GT TH  0000  D2 X/G RD WR CK   0000
27 FSR MTR OVRLD  0000  69 DRUM FAST      0000  8D SUC PRT RSTART 0000                        D3 EXGRF-DECOMP CK0000
28 FSR PAPER SKEW 0000  6A DRUM MTR OVLD  0000  8E SUBSYS RUN RST 0000  AB FSR SCFF MT TH 0000  D4 CPS ER-DECOMP  0000
2A X/F SHORT LOOP 0000  6C TONER OVRFEED  0000  8F CHAN SEL RESET 0000  AC CFS ELEV MT TH 0000  D8 ACCUMULATOR CK 0000
2B X/F LONG LOOP  0000  6D TONER LOOP OPN 0000  90 PROC CLOCK CK  0001  AD CFS MOTOR THRM 0000  D9 ACCUM STRG CK  0000
2E FSR ROLL WRAP  0000                          91 CHARGE CRNA CK 0001  AE MUL THRM SW CK 0000  DA ACCUM/SB CK    0000
                        70 CGEN INTRF CK  0001  92 XFR CORONA CK  0000  AF FSR THERM CHK  0000  DB NO RSP TIMEOUT 0000
30 OUTPT LNGTH CK 0001  71 CGEN CNTRL CK  0001  93 PRCLN CRNA CK  0000  B1 BTS CAM MT TH  0001  DD CPS RD/WR CK   0000
31 DRDY LNGTH CK  0001  72 RPG CHECK      0000  94 MAG BR BIAS CK 0000  B2 FSR ROLL OVR T 0000  DF EXGRF DEC CK   0000

--SUMMARY OF RECOVERED ERRORS FROM MDR RECORDS BY STATUS CODE (INTERNAL LOG ENTRY BYTE 0) - COUNTS IN DECIMAL--

51 MISSING FO FLH 0002  72 RPG CHECK      0000  79 SRLZR INTRF CK 0000  93 PRCLN CRNA CK  0000  D1 X/G CPS CHECK  0002
52 EXTRA FO FLASH 0000  73 RPG SHIFTER CK 0000  7C SER SYNCH CK   0000  94 MAG BR BIAS CK 0000  D2 X/G RD WR CK   0000
63 VACUUM SYS CK  0002  74 STRIP BUFFR CK 0000  7D S/B OVER-RUN   0000  95 POST XFR CORONA0000  D8 ACCUMULATOR CK 0000
65 CLNR BRUSH CK  0000  75 CG XEQ CS CK   0000  7E CG CS START CK 0000  96 LSR PWR SUPPLY 0000  D9 ACCUM STRG CK  0000
66 ERASE LAMP CK  0000  76 LASER POWER CK 0000  7F CG CS CMPLT CK 0000  97 MORROR DR CK   0000  DA ACCUM/SB CK    0001
6A DRUM MTR OVLD  0000  77 MIRROR SPD CK  0000                        BA SYS CHNL CK    0002
70 CGEN INTRF CK  0002  78 SERIALIZER CK  0000  91 CHSRGE CRNA CK 0002  BC FILE READ CK   0009
71 CGEN CNTRL CK  0000                                                BD FILE WRITE CK  0000

--SUMMARY OF STATISTICAL USAGE DATA FROM INTERNAL LOG FOR DATE RANGE INDICATED ABOVE - COUNTS IN DECIMAL--

 BTS COUNT                000000015900
 CFS FEET COUNT           000001524300
 PAPER COUNT              000000102900
```

Figure 2-40. OBR/MDR Summary

Notes:

1. *The statistical data counters keep track of the number of temporary data and equipment checks experienced by the device.*

2. *OBR records reflect permanent (that is, uncorrectable) data and equipment checks. In this report, the data is from long OBR records only. See "Outboard (OBR) Records" on page 10-39.*

3. *Error information kept on the device's internal log becomes MDR records. This column shows the number of entries in the log; the data is summarized below as recovered errors.*

```
SUMMARY OF I/O OUTBOARD ENVIRONMENT RECORDS FOR DEVICE 000571  DEVICE TYPE 3420     TU SERIAL  N/A
                                                                                   CPU MODEL   3084
                                                                                   CPU SERIAL  221128
                      DAY YEAR      DAY YEAR
OUTBOARD DATE RANGE - 130  84   TO  217  85
TOTAL NUMBER OF RECORDS   003
VOLUME LABELS ENCOUNTERED(MAXIMUM OF 10 ENTRIES)
VOL. LABEL  64528      001
VOL. LABEL  T77618     001
VOL. LABEL  T69375     001

CCW COMMAND CODES ENCOUNTERED(MAXIMUM OF 24 ENTRIES)
CMND   TOTAL
SENSE BYTES SUMMARY
BYTE 0              BYTE 1              BYTE 2              BYTE 3              BYTE 4              BYTE 5
CMND REJECT    000  NOISE          000  TRK ERR 0      000  R/W VRC        000  ALU CK/MP ERR 000  NEW SUBSY      XXX
INTV REQUIRED  000  TU STAT A      000  TRK ERR 1      000  MTE/LRCR       000  REJECT TU     000  NEW SUBSY      XXX
BUS OUT CHK    000  TU STAT B      000  TRK ERR 2      000  SKEW           000  TAPE INDICATE 000  WRT TM CHK     000
EQUIP CHK      000  7 TRACK        000  TRK ERR 3      000  EDC/CRCR       000  WRT TR VRC    000  ID BURST CHK   000
DATA CHK       000  LOAD POINT     000  TRK ERR 4      000  ENV/ECC        000  U-PGM DET/RES 000  ST RD CHK      000
OVERRUN        000  WRITE STATUS   000  TRK ERR 5      000  1600 BPI TU    000  LOOP W/R      000  PARTIAL REC    000
WORD COUNT     000  FILE PROTECT   000  TRK ERR 6      000  BACKWARD       000  TU CHECK      000  EXC POST OR TM 000
DT CNVTT       000  NOT CAPABLE    000  TRK ERR 7      000  C/P COMPARE    000  RES FOR RPQ   000  RES FOR RPQ    000
BYTE 6              BYTE 7              BYTE 8              BYTE 9              BYTE 10             BYTE 11
7 TRACK        XXX  LAMP FAIL      000  IBG DROP       000  6250 CORRECTN 000   CMD ST REJ    000  B BUS LSR/MP1  000
WRT HD CR      000  TP BOTTOM LF   000  FEED THRU/RES 000   VEL CHG        000  RESERVED      000  SPARE          000
DUAL DEN       XXX  TP BOTTOM RH   000  RESERVED       000  CBC            000  CTL ST REJ    000  XFER/LOIC      000
NOT 1600       XXX  RES-DOOR       000  EARLY BOR      000  CRC III        000  NO BOR/WTM    000  INST/HIIC      000
                    DSE            000  E END/SAGC CK 000   RLC/3803-2     000  WTM/DRC       000  U-PGM ERR      000
                    ERASE HD       000  SLOW BOR       000  RESERVED       000  TACH FAIL     000  D BUS PTY      000
                    AIR PRES       000  SLOW END       000  RESERVED       000  RESERVED      000  RESERVED       000
                    LOAD FAIL      000  VEL RETRY      000  CNTL UNIT RES 000   VEL CHK       000  BOC ALU1/MP1   000
BYTE 12             BYTE 18             BYTE 19             BYTE 20             BYTE 21
B BUS LSR/MP2 000   PWRCHK/AIRFLW 000   DE DR 7        000  DE DR F        000  LD BUTTON     000
SPARE          000  RESERVED       000  DE DR 6        000  DE DR E        000  LFT REEL      000
XFER/LOIC      000  RESERVED       000  DE DR 5        000  DE DR D        000  RHT REEL      000
INST/HIIC      000  RESERVED       000  DE DR 4        000  DE DR C        000  TAPE PRES     000
U-PGM ERR      000  EC OF DRV      000  DE DR 3        000  DE DR B        000  REELS LDD     000
D BUS PTY      000  EC OF DRV      000  DE DR 2        000  DE DR A        000  LD RWD        000
RESERVED       000  EC OF DRV      000  DE DR 1        000  DE DR 9        000  LD COMPL      000
BOC ALU2/MP2   000  EC OF DRV      000  DE DR 0        000  DE DR 8        000  LD CHK        000
SDR AREA                                                                        DEVICE DEPENDENT COUNTERS
NOISE          0001 WRT TR VRC     0000 EARLY BOR      0000 CRC III        0000 TEMP RDS      00000
R/W VRC        0001 WRT TM CHK     0000 E END/SAGC CK  0000 BACKWARD       0000 TEMP WRTS     00000
MTE/LRCR       0001 ST RD CHK      0000 SLOW BOR       0000 BUS OUT CHK    0000 SIO COUNT     00001358
EDC/CRCR       0000 PARTIAL REC    0001 SLOW END       0000 ALU CK/MP ERR  0000 ERASE GAPS    00000
ENV/ECC        0001 EXC POST OR TM0000 VEL RETRY      0000 ID BURST CHK   0000 CLEAN ACTS    00009
OVERRUN        0000 IBG DROP       0000 6250 CORRECTN  0001                     NOISE RCD     00000
SKEW           0000 FEED THRU/RES  0000 VEL CHG        0000
C/P COMPARE    0000 RESERVED       0000 CBC            0000
```

Figure 2-41. I/O Outboard Environment Summary

```
DEVICE NUMBER:    0576    REPORT: SLH EDIT            DAY YEAR   JOB IDENTITY: D40BAE1
                          SCP:    VS 2 REL.  3   DATE: 109  82                 C4F4F0C2C1C5F140
DEVICE TYPE:      3410
                          CPU MODEL:  3081XA            HH MM SS.TH
CHANNEL PATH ID: 05       CPU SERIAL: X20015     TIME: 11 22 53.30
                            1
                  CC   CA    FL     CT
FAILING CCW       02 BBF108 44 00 1008                   VOLUME SERIAL            T81784
                                                         SUBCHANNEL ID NUMBER     00010259
                  K  FLAGS    CA     US SS  CT           ERROR TYPE               OTHER  2
SCSW              84 024017 00EEFD40 00 02 1008

---UNIT STATUS----  SUB-CHANNEL STATUS  -----------------------SCSW FLAGS-----------------------
                                               FLAG 0                FLAG 1            FLAG 2
                                          3
ATTENTION         0  PGM-CTLD IRPT    0  CCW FORMAT        0  RESERVE         0  SUBCHANNEL ACTIV 0
STATUS MODIFIER   0  INCORRECT LENGTH 0  PRE-FETCH CCW     0  SSCH FUNCTION   1  DEVICE ACTIVE    0
CONTROL UNIT END  0  PROGRAM CHECK    0  INIT STATUS       0  HSCH FUNCTION   0  SUSPENDED        0
BUSY              0  PROTECTION CHECK 0  ADDR LIMIT        0  CSCH FUNCTION   0  ALERT STATUS     0
CHANNEL END       0  CHAN DATA CHECK  0  SUPP SUSPEND INT  0  RESUME PENDING  0  INTERMED STATUS  0
DEVICE END        0  CHAN CTL CHECK   0  ZERO COND CODE    0  START PENDING   0  PRIMARY STATUS   0
UNIT CHECK        0  I/F CTL CHECK    1  EXTENDED CONTROL  1  HALT PENDING    0  SECONDARY STATUS 0
UNIT EXCEPTION    0  CHAINING CHECK   0  PATH NOT OPER     0  CLEAR PENDING   0  STATUS PENDING   0


CHANNEL ERROR ANALYSIS       0

  IRB STORED BY INTERRUPT    1

  TERMINATION  BY --  SELECTIVE RESET  --  CODE   2

  SEQ CODE 2 - COMMAND ACCEPTED BY DEVICE BUT NO DATA TRANSFERRED

  VALIDITY OF RECORDED DATA
    COUNT                INVALID
    TERMINATION CODE     VALID
    SEQUENCE CODE        VALID
    DEVICE STATUS        INVALID
    CCW ADDRESS          VALID
    DEVICE NUMBER        VALID
    SENSE DATA           NOT STORED
```

Figure 2-42 (Part 1 of 2). SLH Detail Edit Report (370-XA only)

```
CHANNEL EXTENDED LOGOUT

  INTERFACE CONTROL CHECK REASON CODE

    BUS IN PARITY              0
    TAG SEQUENCE ERROR         0
    IN TAG ERROR               0
    INTERFACE HANG CONDITION   0
    DISCONNECT IN              0
    DSE TIMEOUT                0
    IAE TIMEOUT                1

  UNIT ADDRESS

    TRANSMITTED          76    INVALID
    RECEIVED             76    VALID
    UA COMPARE ERROR     NO

  CHANNEL PATH ID:       05

  I/O INTERFACE

    TAG LINES
      OP OUT         1      OP IN          1
      SEL OUT        1      SEL IN         0
      SUPPRESS OUT   0      REQUEST IN     0
      HOLD OUT       1      DISCONNECT IN  0
      ADDRESS OUT    0      ADDRESS IN     0
      COMMAND OUT    0      STATUS IN      0
      SERVICE OUT    0      SERVICE IN     0
                            DATA IN        0

   CHANNEL DATA REGISTER:  00   PARITY:     YES

HEX DUMP OF RECORD
HEADER     23831800  00000000  0082109F  11225330  03020015  30810000
    0018  C4F4F0C2  C1C5F140  02BBF108  44001008  32108003  00000000  40807782  82024017
    0038  00EEFD40  00021008  00807482  00000000  00000000  00000000  00000000  00720576
    0058  42207640  00000000  00000000  00000000  00000000  00000000  00000000  00F9B610
    0078  0576E3F8  F1F7F8F4  01000000  01000005  00010259  00000000  00000000  00000000
```

Figure 2-42 (Part 2 of 2). SLH Detail Edit Report (370-XA only)

Notes:

1. *The SLH record is logged in 370-XA mode only. It is identified by CPU complex, not individual CPU serial number. Only the last 5 digits are significant.*

2. *The error type may be storage, key or other. If the error type is storage or key, a line containing the absolute address of the error is printed.*

3. *CCW format is 0 in 24-bit addressing mode, 1 in 31-bit addressing mode.*

```
CHANNEL PATH ID:    24          REPORT:  SLH SUMMARY             REPORT DATE:  113  82
                                                                 PERIOD FROM:  113  82
NUMBER OF RECORDS:  0004        CPU MODEL:    3081XA                     TO:   113  82
                                CPU SERIAL:  X20015

ERROR TYPE: STORAGE    0000
            KEY        0000
            OTHER      0004

  ----UNIT STATUS----          --SUB-CHANNEL STATUS--
ATTENTION          0000        PGM-CTLD IRPT       0000
STATUS MODIFIER    0000        INCORRECT LENGTH    0000
CONTROL UNIT END   0000        PROGRAM CHECK       0000
BUSY               0000        PROTECTION CHECK    0000
CHANNEL END        0000        CHAN DATA CHECK     0000
DEVICE END         0000        CHAN CTN CHECK      0000
UNIT CHECK         0000        I/F CTL CHECK       0004
UNIT EXCEPTION     0000        CHAINING CHECK      0000
```

Figure 2-43. SLH Detail Summary Report (370-XA only)

```
                                                                           DATE       TIME       CPU      CPU
                                                                          DAY YR   HH MM SS.TH   SERIAL   ID
--- RECORD ENTRY SOURCE - SOFTWARE ---  TYPE   SOFTWARE(SVC 13)  065 85   10 31 23 24   270044   3090

    VS 2 REL.    03

               ERRORID=SEQ00001 CPU0042 ASID0001 TIME10.31.23.0

JOBNAME                       *MASTER
ABENDING PROGRAM NAME         N/A          BC MODE PSW AT TIME OF ERROR             BC MODE PSW OF LAST RB
NAME OF MODULE INVOLVED       IEAVFREE
NAME OF CSECT INVOLVED        IEAVFREE          00000000 00000000                     00000000 00000000
FUNCTIONAL RECOVERY ROUTINE   IEAVRCV

REGS AT TIME OF ERROR

REGS 0-7      00FFF000   00C0D000   C0024680   0004C320   00A27170   00000000   00000D08   0000E890
REGS 8-15     0010B520   400B3EE6   00000000   0079FC90   0004C320   00000000   500B3FB0   00000004

EC PSW AT TIME OF ABEND    040C1000 000B3FD2           EC PSW FROM ESTAE RB(0 FOR ESTAI)  040C0000 0009B5C8
ADDITIONAL INFO:                                       ADDITIONAL INFO:
INST LENGTH CODE           02                          INST LENGTH CODE           02
INTERRUPT CODE             000D                        INTERRUPT CODE             000D
VIRT ADDR OF TRANS EXCEP   00CD5006                    VIRT ADDR OF TRANS EXCEP   00CD5006

REGS OF RB LEVEL OF ESTAE EXIT OR ZERO FOR ESTAI

REGS 0-7      00FFF000   00C0D000   C0024680   0004C320   00A27170   00000000   00000D08   0000E890
REGS 8-15     0010B520   400B3EE6   00000000   0079FC90   0004C320   00000000   500B3FB0   00000004

   MCH FLAG BYTE               MCH INPUT INFO          FRAME ERROR INDICATORS   STORAGE ERROR INDICATORS
STORAGE ADDRS ARE VALID    0 STORAGE KEY FAILURE     0 STORAGE ERROR ALREADY SET 0 FRAME OFFLINE(OR SCHED)   0
MCK RECORD NOT RECORDED    0 REGISTERS UNPREDICTABLE 0 CHANGE INDICATOR ON       0 INTERCEPT                 0
TIME STAMP IS VALID        0 PSW UNPREDICTABLE       0                             STORAGE ERROR PERMANENT   0
STORAGE IS RECONFIGURED    0 STORAGE DATA CHECK      0                             PERMANENT RES. STORAGE    0
RECONFIGURE STATUS AVAIL   0 ACR REQUEST             0                             FRAME IN SQA              0
RECONFIGURE NOT ATTEMPTED  0 INSTRUCTION FAILURE     0                             FRAME IN LSQA             0
                             SOFT ERROR              0                             FRAME IS PAGE FIXED       0
                             TIMER ERROR             0                             FRAME IS V=R              0
                                                       TIME STAMP OF ASSOCIATED MACHINE CHECK RECORD
BEGINNING VIRT ADDR OF STORAGE CHECK   00000000                     DATE      TIME
ENDING VIRT ADDR OF STORAGE CHECK      00000000                    DAY YR   HH MM SS.TH
REAL STORAGE FAILING ADDRESS           00000000                    000 00   00 00 00 00

MACHINE CHECK              0 TYPE 1 SVC IN CONTROL   1 PREV ESTA OR FRR FAILED   0 EXIT TO CLEANUP ONLY      0
PROGRAM CHECK              0 ENABLED RB IN CONTROL   0 (E)STAI PREV IN CONTROL   0 RB OF ESTA NOT IN CONTROL 0
RESTART KEY DEPRESSED      0 DISABLED RTN IN CONTROL 1 IRB PRECEDED RB           0 ESTA EXIT FOR PREV ABEND  0
TASK ISSUED SVC 13         0 SYSTEM IN SRB MODE      0 THIS RTN PERCOLATED TO    0 STEP ABEND REQUESTED      0
SYSTEM FORCED SVC 13       0                           LOWER LEVEL EXIT INFO     0 TASK ANCESTOR ABENDED     0
SVC BY LOCKED OR SRB RTN   1                                                       REGS AND PSW UNAVAILABLE  0
TRANSLATION FAILURE        0                                                       MCK INFO UNAVAILABLE      0
PAGE I/O ERROR             0
```

Figure 2-44 (Part 1 of 2). Software Detail Edit Report

```
                                    CURRENT I/O STATUS
MEMORY ASID      .     0000  I/O IS RESTORABLE        0
RECOVERY RETURN CODE    04   I/O IS NOT RESTORABLE    0
                               NO I/O OUTSTANDING       0
                               NO I/O PROCESSINGG       0

ADDITIONAL PROCESSING        GLOBAL LOCKS TO BE FREED   LOCKWORDS
RECORDING REQUESTED      1   DISPATCHER LOCK          0
VALID SPIN               0   SRM LOCK                 0
UPDATED REGS FOR RETRY   1   IOSCAT LOCK              0  IOSCAT LOCKWORD            00000000
FREE RTCA BEFORE RETRY   0   IOSUCB LOCK              0  IOSUCB LOCKWORD            00000000
                             IOSLCH LOCK              0  IOSLCH LOCKWORD            00000000
                             IOSYNCH LOCK             0  IOSYNCH LOCKWORD           00000000
                             NCB LOCK                 0  NCB LOCKWORD               00000000
                             DNCB LOCK                0  DNCB LOCKWORD              00000000
                             ACBDEBS LOCK             0  ACBDEBS LOCKWORD           00000000
                             ASMPAT LOCK              0  ASMPAT LOCKWORD            00000000
                             SALLOC LOCK              0  ASID LOCKWORD              00000000
                             CMS LOCK                 0
                             LOCAL LOCK               0
DUMP CHARACTERISTICS
                                                                                       DUMP RANGES AREA
ADDITIONAL PROCESSING        GLOBAL LOCKS TO BE FREED   LOCKWORDS
DUMP FLAGS                   SDATA OPTIONS              PDATA OPTIONS                        FROM      TO
SNAP DUMP REQUEST        0   DISPLAY NUCLEUS          0  DISPLAY SAVE AREAS         0  RANGE 1  00000000  00000000
PARM LIST SUPPLIED       0   DISPLAY SQA              0  DISPLAY SAVE AREA HEADER 0  RANGE 2  00000000  00000000
STORAGE LIST SUPPLIED    0   DISPLAY LSQA             0  DISPLAY REGISTERS          0  RANGE 3  00000000  00000000
                             DISPLAY SWA              0  DISPLAY TASK LPA MODULES 0  RANGE 4  00000000  00000000
                             DISPLAY GTF TRACE TABLE 0   DISPLAY TASK JPA MODULES 0
                             DISPLAY CONTROL BLOCKS  0   DISPLAY PSW                0
                             DISPLAY QCB/QELS        0   DISPLAY USER SUBPOOLS      0
   HEX DUMP OF RECORD
   HEADER    40830820  00000000  0085065F  10312324   00270044  30900000  5CD4C1E2  E3C5D95C
       0000  00000D08  00C0D000  00000000  00000000   00000000  00000000  00FFF000  00C0D000
       0020  C0024680  0004C320  00A27170  00000000   00000D08  0000E890  0010B520  400B3EE6
       0040  00000000  0079FC90  0004C320  00000000   500B3FB0  00000004  00000000  00000000
       0060  00000000  00000000  040C1000  000B3FD2   0002000D  00CD5006  04000000  0009B5C8
       0080  0002000D  00CD5006  00FFF000  00C0D000   C0024680  0004C320  00A27170  00000000
       00A0  00000D08  0000E890  0010B520  400B3EE6   00000000  0079FC90  0004C320  00000000
       00C0  500B3FB0  00000004  00000000  00000000   00000000  00000000  00000000  00000000
       00E0  0000000F  00000000  040A0001  00000042   000B3FD2  00FE8DC4  00000000  04980000
       0100  00000000  00000000  00000000  00000000   00000000  00000000  00000000  00000000
       0120  00010001  C9C5C1E5  C6D9C5C5  C9C5C1E5   C6D9C5C5  C9C5C1E5  D9C3E540  00FE8D70
       0140  00000000  00000000  00000000  00000000   00000000  00000000  00000000  00000000
       0160  00000000  00000000  00000000  FFFF0001   00FE8F08  80000001  00010001  00000000
       0180  00000000  00000000  00000000  0005C7CE   00FF8064  00000000  00000000  00508010
       01A0  00000000  01800000  00000000  00000000   00000000  00000000  00000000  00000000
       01C0  00000000  00000000  00000000  00000000   00000000  00000000  00000000  00000000
       01E0  00000000  00000000  00000000  00000000   00000000  00000000  E2C3F1C3  D9000000
       0200  00000000  00000000  00000000  00000000   00000000  00000000  00000000  00000000
       0220  00000000  00000000  00000000  00000000   00000000  00000000  00010042  00010005
       0240  C7CE
```

Figure 2-44 (Part 2 of 2). Software Detail Edit Report

```
                     DAY YEAR   DAY YEAR       MODEL-  0168              SERIAL NO.     060702

SOFTWARE DATE RANGE - 051  80 TO 051  80


                                       SUMMARY OF SOFTWARE ENVIRONMENT RECORDS
TOTAL NUMBER OF RECORDS     0009


  ROUTINE  CSECT  NUMBER    ROUTINE  CSECT  NUMBER    ROUTINE  CSECT  NUMBER    ROUTINE  CSECT  NUMBER
   NAME    NAME  ENTRIES     NAME    NAME  ENTRIES     NAME    NAME  ENTRIES     NAME    NAME  ENTRIES

  IEBGENER IEBGENER 005     IGXF00BB IFCER02  003     IGXF0003 IFCER01  001
```

Figure 2-45. Software Detail Summary Report

```
                                                              DATE       TIME          CPU      CPU
                                                             DAY YR   HH MM SS.TH    SERIAL    ID
VS 2 REL.    03    - SOFTWARE ---  TYPE   LOST REC SUMMARY  119 82   03 28 50 08    020015    30

NO ERRORID ASSOCIATED WITH THIS RECORD

MISSING RECORD COUNT         095
   HEX DUMP OF RECORD
   HEADER    4F831880  00000000  0082119F  03285008    03020015  30810000

       0000  5F
```

Figure 2-46. Lost Record Summary (MVS only)

```
IPL RECORD EDIT AND PRINTING SECTION
      DAY  YEAR                             HH MM SS TH
DATE -287   86                        TIME -16 51 52 17
  MODEL - 9375      CPU SERIAL NO. -  234567
    VS 2 REL.    02

--CHANNEL TYPE--
   CHANNELS 0-15
SEL       SEL       BLKMPX    BLKMPX    BLKMPX    BLKMPX    SEL       MPX
BLKMPX    MPX       UNATT     UNATT     UNATT     UNATT     UNATT     UNATT
          IPL REASON CODE - DF   DEFAULT -U-
          SUBSYSTEM ID - 00                       SUBSYSTEM NAME - NULL
          HIGHEST STORAGE ADDRESS  00FFFFFF


   HEX DUMP OF RECORD
   HEADER    50820000    00001100    0086287F    16515217       10234567    93750000

       0000  00000000    C4C6BFC0    22333321    31000000       00FFFFFF    00000000
```

Figure 2-47. System Initialization (IPL) Detail Edit Report

```
SUMMARY OF IPL RECORDS                                                                MODEL 9375
                                              DAY YEAR     DAY YEAR             CPU SERIAL  234567
                    DATE RANGE  FROM     287   86    TO 290   86
                    NO. OF RECORDS  008


  XXXX  SUBSYSTEM NAME AND NUMBER OF OCCURRENCES XXXX
  NULL             008          PROCESSOR                     000
  TAPE             000          TELEPROCESSING                000
  MICR/OCR         000          GRAPHIX/DISPLAY/AUDIO         000
  CARD/PRINT       000          IBM SYSTEM CONTROL PROGRAM    000
  DIRECT ACCESS    000          IBM PROGRAMMING PRODUCT       000
  OTHER            000
  XXXX IPL REASON CODE AND NUMBER OF OCCURRENCES XXX
  NORMAL           000          MEDIA                         000
  UNKNOWN          000          OPERATIONAL                   000
  USER PROGRAM     000          ENVIRONMENTAL                 000
  IBM HARDWARE PROGRAMMING PROBLEM-CE/SE NOT REQUIRED  000
  IBM HARDWARE PROGRAMMING PROBLEM-CE/SE REQUIRED      000
  CE/SE HAS THE SYSTEM      000
  DEFAULT -U-               008
  INVALID IPL REASON CODE  000
         XXXXXXX END OF IPL SUMMARY XXXXXXX
```

Figure 2-48. System Initialization (IPL) Detail Summary Report

```
EOD RECORD EDIT AND PRINTING SECTION

      DAY   YEAR                              HH MM SS TH
DATE -234   86                          TIME -05 31 41 68

  MODEL - 3090       CPU SERIAL NO. -  270150
    VS 2 REL.     03


   HEX DUMP OF RECORD
   HEADER    80831800    00400000    0086234F    05314168        40270150    30900000

       0000
```

Figure 2-49. System Termination (EOD) Detail Edit Report

```
SUMMARY OF EOD RECORDS                                                            MODEL 3090
                                        DAY YEAR     DAY YEAR                CPU SERIAL  270150
                     DATE RANGE  FROM   219  86    TO 234  86

                     NO. OF RECORDS  005

                     XXXXXXX END OF EOD SUMMARY XXXXXX
```

Figure 2-50. System Termination (EOD) Detail Summary Report

```
RECORD TYPE UNKNOWN OR UNSUPPORTED

MODEL-9373          SERIAL NO-  010620

--- RECORD ENTRY SOURCE - MCH  1

    V370 REL.    06

       DAY  YEAR                              HH MM SS TH
DATE-  211   86                        TIME-  16 45 08 59

PROGRAM IDENTITY-   CP/370

JOB     IDENTITY-

OLD MACHINE CHECK PSW - 00 0C 00 00 00 01 09 0A

   HEX DUMP OF RECORD
   HEADER     10660800   00000000   0086211F   16450859     00010620   93730000   4040C3D7   61F3F7F0
              00000000   00000000   000C0000   0001C90A

       0030   04010F3D   00030000   00000000   22000000     00000000   00000000   00000000   00000000
       0050   00000000   00000000   00000000   00000000     00000000   00000000   00000000   00000000
       0070   00000000   00000000   00000000   00000000     00000000   00000000   00000000   00000000
       0090   00000000   00000000   00000000   00000000     00000000   00000000   00000000   00000000
       00B0   00000000   00000000   00000000   00000000     00000000   00000000   00000000   0001CE10
       00D0   00000000   0001D2EC   4001C8F4   8001D2F0     003D0380   00000034   003D0380   003D04C8
       00F0   070D0000   00065678   0001C3A8   001FA1A8     00395480   00033584   80800EC0   063D0940
       0110   FFFFFFFF   00000000   00000000   00000000     A33A33B8   00000000   00000000   00000000
       0130   00000000   00000000   00000000   00000000     CFC00000   00000000   00580000   00000000
       0150   80000000   00000004   00000000   00000000     00000000   00000000   00000000   00000000
       0170   00000000   00000000   00000000   00000000     00000000   00000002   00000000   00000000
       0190   00000000   00000000   003D0380   0039BAB0
```

Figure 2-51. Unknown or Unsupported Record Detail Edit Report

Notes:

1. *Depends upon the record type. It will be either the hex representation of the first byte in the record or a 3 character description of the record type. See Figure 10-6 on page 10-12.*

```
SUMMARY OF UNKNOWN OR UNSUPPORTED RECORDS

                      DAY YEAR   DAY YEAR
RECORD DATE RANGE     264  86    264  86

MODEL-9373        SERIAL NO - 234567

TOTAL NUMBER OF RECORDS=0002

CLASSES ENCOUNTERED(MAXIMUM OF 10)

RECORD CLASS -10   0002
```

Figure 2-52. Unknown or Unsupported Record Detail Summary Report

Interpreting EREP Reports

After you have succeeded in generating a set of EREP reports, you want to look at them to see what they indicate about your data processing system. In many installations, someone looks the reports over for indications of obvious problems and calls the IBM Customer Engineer if a problem seems to be hardware-connected.

You should become familiar with the general appearance of the reports, so you can tell if you are getting the right output from the EREP run; you may want to do some of your own problem determination, as well.

The examples of EREP reports earlier in this chapter are meant to give you a general idea of what each one is supposed to look like, but there is no way to show you all the possible variations caused by different devices and different parameter combinations.

For that kind of information, you need the maintenance documentation for the product. Some IBM products provide detailed directions on how to interpret the information in EREP reports; see the bibliographies included in some of the product sections in Part 4.

This book can also help you distinguish among the various record types and the devices or products each type is associated with. Figure 12-2 on page 12-4 shows the records generated by the different product groups, and the reports EREP produces for those records. Chapter 10, "Error Records for EREP" includes brief explanations of why the various records are generated, along with their formats.

For detailed information about the significance of particular fields in the EREP reports, you must look in the maintenance documentation for the devices in your installation.

Some observations about how EREP arrives at some of the information in the reports follow.

How EREP Assigns Letters to CPUs

In nearly all its reports, EREP identifies each processor by a single letter of the alphabet. It assigns the letter identifiers separately for each report, based on the processors' model and serial numbers and when it encounters them. If your EREP controls include SHARE or CONTROLLER statements, EREP assigns letters to the processors you specify on those statements before reverting to a default method. You can take advantage of this sequence to force EREP to assign specific letters to specific processors, and to use the same letter for each processor in all the EREP reports.

Sequence of Letter Assignments

As it processes records for a requested report, EREP builds a table of CPU letter assignments, starting with those CPUs that appear on SHARE or CONTROLLER statements.

EREP assigns letters to processors (CPUs) appearing on SHARE or CONTROLLER statements as follows:

1. It examines the *first entry on every statement*, assigning the next letter of the alphabet to each new CPU model or serial number it encounters.

2. After assigning letters to the CPUs in all the first entries, it examines *the rest of the entries on each statement in turn*, assigning the next letter to each new CPU serial number it finds.

3. After completing these assignments, EREP assigns letters to *any processors it encounters in the input data that are not specified on SHARE or CONTROLLER statements*, using its default method.

The default method is to assign letters to processors in the order in which they occur in the input data. These letter assignments can change from one report to the next, if the reports use different error records.

The following example illustrates EREP's letter assignments for CPUs that appear on SHARE or CONTROLLER statements.

```
SHARE=(000001.120,000002.120,000006.120)
SHARE=(000003.130,000004.130)
SHARE=(000005.140,000003.140)
```

Assume that EREP also encounters CPU serial number 000007 in the input data. EREP assigns letter identifiers to all of these processors as follows:

Letter Identifier	*CPU Serial Number*
A	000001
B	000003
C	000005
D	000002
E	000006
F	000004
G	000007

Forcing CPU Letter Assignments

All of these letter assignments could be different for the next report, because EREP reassigns letters to CPUs for each report. You can, however, force EREP to assign the same letter to the same CPU for every report[3] by using SHARE statements as follows:[4]

- Set up SHARE statements for each of your processors, in which the first entry is for the actual processor and the second entry contains a dummy CPU serial number.
- Make one of these SHARE statements the first one for each group associated with the real processor serial number.
- When EREP encounters the SHARE statements while assigning CPU identifiers, it will assign the letter A to the processor represented by the first entry on the first dummy SHARE statement, B to the first entry on the next dummy SHARE statement, and so on. It will eventually assign a letter to the dummy serial number, too, but the identifier will not appear on the report because there will be no records associated with the dummy processor.
- If you include these SHARE statements for every EREP run, every report except the Event History will identify the same processors the same way.

---

3 Except for the Event History, which cannot use SHARE or CONTROLLER statements because it must order the records chronologically.

4 Even if you don't need SHARE statements to identify shared devices.

A Planning Checklist

Having read the first two chapters of the *EREP User's Guide*, you are now ready to start setting up an EREP run. The following checklist contains the questions you need to ask yourself as you prepare to set up EREP for your installation. Where relevant, the questions are followed by pointers to information elsewhere in this book that can help you answer "Yes."

__ 1. Report content and sequence agreed upon with CE?

__ 2. Working copy of LOGREC/SYSREC on disk? See "Managing Error Data" on page 3-1.

__ 3. History data set up to date and usable? See "Managing Error Data" on page 3-1.

__ 4. DASDID statements ready for System Exception reports? See "DASDID Control Statement" on page 8-7.

  __ a. Trial run for TOURIST output?
  __ b. Configuration chart on TOURIST agree with the installation's actual configuration?

__ 5. LIMIT statements ready for System Exception reports? See "LIMIT Control Statement" in Part 3, and the LIMIT Control statement subsections under Chapter 20, "Processors (CPUs)," "34XX Tape Devices," and "33XX DASD" in Part 4.

  __ a. Trial run for syntax check?

__ 6. SHARE/CONTROLLER statements ready? See "SHARE Control Statement" on page 8-18 and "CONTROLLER Control Statement" on page 8-4.

__ 7. Should the region/partition size be increased? See "VSE Storage Requirements" on page 4-16 or "MVS Storage Requirements" on page 4-34.

__ 8. Are you setting up EREP to run automatically?

  a. For a VSE EREP run, set up a job to be submitted via ICCF
  b. For an MVS EREP run, set up a job to be submitted via TSO
  c. For a VM EREP run, set up CMS EXECs

  __ a. EREP run schedule ready for operator?
  __ b. Running EREP automatically off timer?
  __ c. Which reports?
  __ d. Which other functions?
  __ e. In what order?
  __ f. How often?

  See "Automating the Running of EREP" on page 3-7.

__ 9. Have you set up a procedure to get an EREP report on demand? See "Modifying Your EREP Run" on page 3-8.

__ 10. Report distribution set up?

- __ a. CE
- __ b. PSR
- __ c. System programmer
- __ d. Other

You must, of course, review your EREP plan whenever your configuration changes, and ask these questions:

- Does the new configuration require new SHARE, DASDID, or LIMIT controls?
- Does it mean getting an additional report or set of reports?
- Does it require a new history tape because of changed device addresses?

The specifics of EREP support for new products are in Part 4, "Product-Dependent Information."

Part 2. Setting Up and Running EREP

How to Use Part 2

This part of the *User's Guide* contains the information you need to set up an EREP run for your installation. Most of the information is grouped according to operating system control program (SCP) and includes the following:

- An annotated master example of the EREP and system controls that make up a typical EREP run
- A description of the system data sets and controls required for running EREP under the SCP
- Other specific coding and storage considerations for running EREP.

Before and after this SCP-specific information, Part 2 presents several more general topics that are important to the process of running EREP:

- Suggestions for managing and maintaining your error-record data
- Suggestions for automating the running of EREP
- Suggested ways to modify EREP controls for a particular run
- Suggestions for correcting coding problems.

Use the Section Table of Contents to find things in Part 2.

If you need more detailed information about the syntax and coding of EREP parameters and control statements, it is in Part 3.

If you want further information about your SCP or specific devices and products, see Part 4.

For examples of the EREP reports, see Chapter 2, "EREP Reports"; there are also a few in Part 4.

Part 2 - Section Table of Contents

Chapter 3. Setting Up EREP

Before you start setting up an EREP run for your installation, there are a few general topics relating to the running of EREP that you might find helpful. Managing the error data, automating your EREP run, and modifying an EREP run, are all things you should consider as you prepare the system and EREP controls for your EREP run.

Managing Error Data

If EREP is to be useful and helpful, it must have "good" data to edit and print. You must manage your ERDS and other error data sets so that the records they contain truly reflect the state of your installation's hardware and operating system.

Managing error data for EREP, like managing any data base, has several aspects. Two of particular importance to EREP are:

- The integrity of the data base: making sure the input to EREP includes all the records produced, but no duplicates.
- The consistency of input to a given EREP run: making sure all the reports in a run use the same records.

In addition to these is the matter of saving the records in case of emergency. EREP includes a special routine that takes care of this situation; see "Clearing the ERDS in an Emergency" on page 3-4.

Maintaining ERDS Data Integrity

When your system's ERDS is almost full, the recording routines sense it and issue a message to the operator. The operator is supposed to run EREP at that point, to clear the ERDS. Usually, this EREP run requests a System Summary; it always includes ACC=Y and ZERO=Y, to write the contents of the ERDS to another data set and to clear the ERDS so the recording routines can write more records to it. Although the procedure is quite straightforward, there are several potential problems:

1. In an MVS system when you request a System Summary,[1] EREP forces the system to dump the statistical counters from storage and from buffered-log devices to the ERDS in the form of OBR and MDR records.[2] If the operator delays the running of EREP until the ERDS is completely full, there is no room for the additional records; the statistical data is lost. In such a situation, it is better to use IFCOFFLD; "Clearing the ERDS in an Emergency" on page 3-4 tells you how to do this.

2. If one or more I/O devices are not able to complete the transfer of data from buffered-log when requested by EREP, then the EREP job may not complete, and reports may not be produced. In such a situation IFCOFFLD may be used to transfer the contents of the ERDS to the history data set and then the history data set may be used to produce EREP reports.

3. If the step or job fails for any reason and you subsequently rerun it, you could duplicate some of the records already on the output data set.

Merely knowing about these problems should help you avoid them; the ways to avoid them differ slightly from one operating system to another. The following sections include some suggestions to help ensure the integrity of EREP's input data.

Maintaining Consistency of Input across Reports

Using EREP properly gives you the ability to monitor and track potential failures in your installation's hardware and software, and thus to catch them before they cause serious damage.

Part of using EREP properly is running it regularly. In a large MVS installation, this means daily; in a smaller VSE installation, weekly. The main thing is to set EREP up so it can be run easily. "Automating the Running of EREP" on page 3-8 addresses this task in a general way. Your installation's policies and the operating system you want to run EREP under determine how automatic the process of running EREP can be.

A secondary consideration is the kind of data each run processes. The ERDS is a dynamic data set; it is constantly being updated by the system's recording routines, even while you are running EREP against it. When the ERDS is the source of input records for EREP, especially in a multi-step run, the data on the ERDS can change from one report to another.

If you want to compare the contents of several reports, you need consistent input for each report. The only way to get it is to create, at the beginning of your EREP run, a static working data set that can be used as input for all the subse-

---

[1] And some other reports as well; see "Statistical Data on the ERDS" on page 3-7.

[2] A VSE system writes such statistical records only when the operator issues the ROD (record on demand) command.

quent reports. Then, every report draws records from the same data base. See "Creating a Working Copy of the ERDS" on page 3-5.

Clearing the ERDS in an Emergency

If the ERDS is filling up too quickly, so that it could overflow, you can use the special EREP procedure named IFCOFFLD to offload the records to another data set as quickly as possible.

IFCOFFLD does essentially the same thing a normal ERDS offload procedure does: it produces a System Summary using the records on the ERDS, then copies the records to an output data set and clears the ERDS. The only difference between the two procedures is that IFCOFFLD does not cause the dumping of statistical data to the ERDS prior to reading records for the System Summary. This difference is significant, because it prevents the loss of statistical data. It also saves time.

IFCOFFLD preserves the data on the ERDS and gives you a summary report that can help you find the problem that caused the ERDS to fill up.

Note: *Because IFCOFFLD does not dump any statistical data to the ERDS, you should not use it to offload the ERDS under normal conditions; it is meant for emergency use only.*

IFCOFFLD is for VSE and MVS systems; in a VM system, the records in the error-recording area can only be offloaded by a normal EREP run that includes ACC=Y and ZERO=Y. It is wise to set up a separate EXEC for the operator to run when the page-full message appears on the console. The EXEC should duplicate the processing that IFCOFFLD does. See "Entering CPEREP Operands" on page 9-8 for more about running EREP under VM.

IFCOFFLD under VSE

If you have to offload SYSREC in an emergency, use the following JCS:

```
// TLBL   HISTOT
// ASSGN  SYS009,TAPE
// EXEC   IFCOFFLD
/*
/&
```

This job will do the following:

- Generate a System Summary report without dumping the buffered and in-storage statistical data to SYSREC
- Copy the records from SYSREC to the output data set (SYS009)
- Zero out SYSREC.

It would be an good idea to assign your regular permanent EREP history data set to SYS009 for IFCOFFLD. It would be an excellent idea to set up an IFCOFFLD procedure to be started from the console as needed. See "Automating the Running of EREP" on page 3-7.

IFCOFFLD under MVS

If you have to clear LOGREC in an emergency, use the following JCL:

```
//OFFLD    JOB   accounting/programmer information
//STEP1    EXEC   PGM=IFCOFFLD
//SERLOG   DD  DSN=SYS1.LOGREC,DISP=OLD
//ACCDEV   DD  UNIT=unit,DSN=name,DISP=(status,disposition),
//             DCB=(RECFM=VB,BLKSIZE=size)
//TOURIST DD  SYSOUT=A,DCB=BLKSIZE=133
//EREPPT   DD  SYSOUT=A,DCB=BLKSIZE=133
//SYSIN    DD  DUMMY
/*
```

This job will do the following:

- Generate a System Summary without dumping the buffered and in-storage statistical data to LOGREC
- Copy the records from LOGREC to the ACCDEV data set
- Zero out LOGREC.

Where to Put the Offloaded Records

No values appear in the preceding example for the parameters on the ACCDEV DD statement because you have some choices about which data set to use. You can set up the emergency offload job so it writes the LOGREC records to a new data set or to an existing data set, depending on how you want to handle the input to your regular EREP run.

If you have to offload LOGREC in an emergency, you certainly want to include those records in the next regular EREP run, so you can see what caused LOGREC to fill up so quickly. You can include the offloaded records one of at least two ways:

1. Concatenate the data set from IFCOFFLD to the ACCIN DD statement in all but the first and last steps of the EREP run.
2. Offload LOGREC directly to the working data set itself, and then add any more records from LOGREC to that data set in the first step of the EREP run.

Each of these alternatives has advantages and disadvantages.

Concatenating the IFCOFFLD data set to the ACCIN DD statements is straightforward, but not ideal for performance, as EREP would be opening and trying to read from an empty data set most of the time.

Using the same data set for both functions is the most efficient, but it involves an additional step in the regular EREP run: you must, after deleting the working data set, immediately create it again so it is available for IFCOFFLD. The first step in the regular EREP run would modify the existing data set when offloading LOGREC to it, instead of creating it (DISP=(MOD,PASS) instead of

DISP=(NEW,PASS)). You would probably want to use a utility, such as IEFBR14, to recreate the working data set after deleting it.

Either one of these alternatives lends itself well to the automated EREP; in both cases, if IFCOFFLD runs, the records are included in the next EREP run without anyone having to do anything about it.

When you have decided what you want to do about this question, you can set up IFCOFFLD in a cataloged procedure that can be started from the operator's console as needed. See "Automating the Running of EREP" on page 3-7.

Creating a Working Copy of the ERDS

The easiest way to ensure the integrity of the data on your ERDS is to make a working copy of the data set either before running any EREP reports or as a byproduct of the first report.

If you copy the records from the ERDS to another data set and then run your EREP reports using the working data set as the input source, you accomplish three things:

1. Protect the record data from being changed or damaged while you are running the EREP reports.
2. Avoid repeated offloading of statistical counters, something that is unnecessary and can be costly in terms of both time and storage.
3. Guarantee that all the reports in a given run use the same data base, making them useful for comparison.

If you run EREP regularly and frequently, making a working copy of the ERDS also makes sure that the ERDS is cleared regularly and frequently — a good practice.

The three sample EREP runs later in this section all copy the records on the ERDS to a working data set in their first step.

See "Transferring Error Data to Another Data Set" for some data-integrity considerations. In addition, you should consider the possibility of IFCOFFLD having copied the contents of the ERDS prior to the regular EREP run. See "Clearing the ERDS in an Emergency" on page 3-3.

Transferring Error Data to Another Data Set

In several situations described here, you are copying records from one data set to another — with or without requesting EREP reports at the same time.

Whenever you offload data from a data set, you run the risk of losing or duplicating or otherwise ruining the data in the process. In the sample procedures in the next chapter, the error records are copied twice: in the first step, from the ERDS to a working data set; and in the next-to-last step, from the working data set to the "permanent" history data set.

In both steps, the potential for ruining the data exists. If the step fails for some reason and then is rerun, and the output data set is being added to, the result is duplicate records on the output data set. If the step does not fail, but no records are copied to the output data set, subsequent steps could be using an empty input data set for the EREP reports. There are many possibilities, all of which you want to anticipate and prevent.

You can prevent even the most remote of these problems by including safeguards in your EREP setup. The sample procedures in the next chapter do not show such safeguards because there are many to choose from. As the most obvious instances:

- Use the appropriate parameters (COND, for MVS) or control statements (IF . . . GOTO, for newer VSE systems) to prevent execution of subsequent steps if the offloading step fails.

- Add control statements (SYSUDUMP, for MVS) to request a dump in case the offloading step fails.

- Separate the generation of reports like System Exception from the copying of records.

Statistical Data on the ERDS

The MVS systems write statistical and usage data to LOGREC whenever you request certain reports using the LOGREC data as input, so as to give you the very latest information about your I/O devices. VSE systems do the same thing when the operator issues the ROD (record on demand) command before running an EREP report. The data comes from counters associated with the devices and located either in buffers or in storage, depending on the device; the operating system dumps it to the ERDS in the form of MDR and OBR records.

These are the reports for which EREP forces the MVS systems to dump the counter data:

- System Summary
- System Exception series
- Threshold Summary
- Trends report
- Any report for which you have specified a device type (DEV) or ZERO=Y.

Depending on your situation and needs, you may or may not want to have statistical data records added to your ERDS whenever you run these reports.

If you are running EREP as a series of steps that request one report after another, the repeated dumping of the statistical data is costly of I/O overhead and processing time. And the data is of little practical value because it is collected over so short a time. A System Exception report step, for example, can fail for lack of storage because EREP is trying to process so many statistical records.

The addition of statistical data can also affect the consistency of the data used in a series of reports — one of the main arguments for first creating a working copy of the ERDS.

If you need to see the latest usage statistics for an I/O device, you can do so by running a separate detail PRINT report and specifying the device type (DEV parameter).

The situation is not so hard to manage in a VSE installation, because the operator can directly control the dumping of statistical counters to SYSREC. However, semi-automatic use of the ROD command can lead to the same inefficiencies and inconsistencies described for the other systems.

Automating the Running of EREP

For EREP to be most helpful, it should be run regularly and frequently. To save time and avoid mistakes, you can set up a procedure or series of jobs or EXECs, depending on the operating system, that can be started by the operator or automatically off a timer.

The sample EREP runs in the next chapter lend themselves to such automation because they cover, in general, all the kinds of reports you would want to see from a normal EREP run. Simply add the system controls needed to make one into a cataloged procedure (VSE and MVS) or a series of EXECs (VM), and it can be your basic EREP setup.

If yours is a VSE installation, you can set up two procedures to be run weekly — one for the first run of the month, and the other for subsequent weeks — that will create and update a monthly history tape. The operator can choose and run the appropriate procedure once a week.

For an MVS installation, the EREP runs can be in cataloged procedures, to be started by the operator or by a timer at set intervals. Ideally, you would create several procedures to cover various situations.

Also for MVS installations, the reports themselves can be sent to an output device other than a printer: to a TSO terminal, for example, for online viewing.

Under VM, you can create the series of EXECs illustrated in the sample run and make the whole series into a single EXEC to be run from the console or off a timer; or you can name each of the EXECs (EXEC SYSUM, EXEC 37XX, and so on) and run each report separately.

Modifying Your EREP Run

One of the benefits of setting up a procedure or EXEC to run EREP from the console is the fact that such online procedures are easy to modify if necessary. However, it would be even better to set up separate, smaller procedures to deal with the various situations that might call for a single EREP report at a time.

- The emergency of the overflowing ERDS is an extreme example of such a situation, and IFCOFFLD is the specific solution to that particular problem. Remember that IFCOFFLD should be ready to start from the console at any time.

- If one of the general-purpose reports reveals a problem with a particular product or group of devices, you might need to get detailed information about the device type or the particular error type, in order to isolate the problem. It is easy to select the right report if you have named your series of procedures for the reports or devices involved.

  In a variation of this, you could set up your usual EREP run to focus on summary data, including only Detail Summaries of specific device and record types besides the more general reports. If a problem shows up, you can then run Detail Edit reports for the suspect devices to find the exact cause. For faster access to the detail edit information, spool the EREPPT data set to a display device instead of the printer.

- Similarly, if any report is accompanied by an EREP message about the input record(s), you might want to rerun EREP using the special parameter that prints out the records in hexadecimal. A procedure that requests an Event History and includes a symbolic name for the DEBUG=(17) option would let you see the record in question almost immediately. (See "The EREP DEBUG Parameter" in Part 3.)

In any of these situations, you could modify one of the steps in your procedure or EXEC and run the whole procedure again. Or you can separate out the steps you might need, set them up as individual procedures or EXECs, and have them ready to run as needed. Either way, you will be using EREP to the best advantage.

Chapter 4. Running EREP

On the following pages are examples of EREP runs for each of the S/370 operating systems. Each example has been tested using actual records; if you fill in the correct information for your installation, the example for your operating system should run as coded.

All three examples include the same series of reports: the reports considered to be generally most useful for a typical S/370 installation. Furthermore, all three examples request the reports in a particular order: the order that allows for creation of a working copy of the ERDS at the beginning and for the updating of a permanent history tape at the end.

The examples include notes that explain what the various data sets and controls are for, and point to more detailed information elsewhere in the book. ***These notes should not be included if you run the examples; they are not correct job control language comments.***

- The master example for VSE systems is under "Running EREP under VSE" on page 4-2.
- The master example for MVS systems is under "Running EREP under MVS" on page 4-20.
- The master example for VM systems is under "Running EREP under VM" on page 4-38.

Note: Make sure you know which level of EREP is installed on your operating system; sometimes new EREP support and SCP support do not match, so you might not be able to use all the EREP features.

Following each example of EREP code is a detailed explanation of the system controls needed to run EREP under the operating system, and notes specific to the system.

Running EREP under VSE

All of the EREP parameters are described in detail in Chapter 7, "EREP Parameters"; the EREP control statements in Chapter 8, "EREP Control Statements."

The History Dataset

You should always create a history data set before running your EREP reports. This insures consistency across the reports.

The operating system adds records to the ERDS continually. If you run your reports against the ERDS (HIST=N) then the set of records used by the first report is not the same as the set used by the next report since records could have been added while the report was running. By the time the last report is run the set of records may be quite different.

By creating a history data set and then running all of the reports against that data set you insure that all of the reports are using the same set of records.

The JCS on pages 4-3 through 4-12 shows a VSE job with several steps. The first step creates a history data set which is used in the remainder of the steps. ***This is only an example.*** You must decide which reports are relevent to your installation, what order they should be generated in and how often they should be run.

Create a History Dataset

Copy SYSREC to a tape data set

Note: The operator must mount a scratch tape for use in creating the working history tape, and must issue the ROD command to force the recording of statistical data on SYSREC before EREP begins copying the records.

`// JOB EREPJOB`	
`// TLBL HISTOT`	Creates the tape label for the output (ACCDEV) data set.
`// PAUSE   ASSIGN SYS009 TO A SCRATCH TAPE` `// PAUSE   ISSUE ROD COMMAND`	Instructions to the operator.
`// ASSGN SYS001,cuu`	Defines the DIRECTWK data set. Logical units SYSLST (for report and message output) and SYSREC, the ERDS, are assigned at IPL.
`// DLBL IJSYS01` `// EXTENT SYS001,xxx ...`	Creates the disk label for DIRECTWK. Required for disk data sets. SYS001 must be a single extent.
`// EXEC IFCEREP1,SIZE=AUTO`	Partition size must be large enough to accommodate EREP and a sort table as indicated by TABSIZE. EREP parameters and control statements are entered as instream data.
`PRINT=NO`	Want no printed reports yet.
`ACC=Y`	Copying the records from SYSREC to SYS009.
`ZERO=N`	Want to preserve SYSREC so it can be merged with the monthly history tape in the job on page 4-13.
`/*`	End of instream data.
`// MTC REW,SYS009`	Rewind the ACCDEV (now history) tape.

System Summary Report

Generate a System Summary report from the records on the working history data set.

`// TLBL HISTINT`	Creates a tape label for the input history data set (ACCIN).
`// ASSGN SYS008,SYS009`	Assigns the ACCDEV data set to the input (ACCIN) logical unit.
`// ASSGN SYS009,UA`	Releases the logical unit used for ACCDEV.
`// EXEC IFCEREP1,SIZE=AUTO`	Partition size must be large enough to accomodate EREP and a sort table. You may encounter storage problems if you code SIZE=AUTO and your partition is not large enough. See "VSE Storage Requirements" on page 4-16 before coding this parameter.
`SYSUM`	Requests the System Summary.
`ACC=N`	Want to keep the records in the working data set.
`TABSIZE=50K`	The default value for TABSIZE is 4K; judge the need for a larger sort table by the number of records on SYSREC and the kind of report being requested.
`ENDPARM`	End of parameters; control statements follow.
`SHARE . . .`	For I/O devices shared between processors.
`/*`	End of instream data.
`// MTC REW,SYS008`	Rewind the tape holding the working history data set.

System Exception Reports

Generate the series of System Exception reports for processors, channels, and DASD and tape subsystems:

- A System Error Summary in two parts:
  1. IPL/restarts, machine and channel checks
  2. DASD and tape I/O errors across processors.
- A Subsystem Exception report for each processor
- A Subsystem Exception report on channel failures, one for each processor
- A Subsystem Exception report on DASD failures
- A Subsystem Exception report on tape failures
- A series of detailed summaries for DASD presenting data by symptom code, storage control unit, string, and volume
- A series of detailed summaries for 34XX tape devices presenting permanent and temporary errors by CUA/device number and volume.

`// EXEC IFCEREP1,SIZE=AUTO`	Partition size must be large enough to accomodate EREP and a sort table. You may encounter storage problems if you code SIZE=AUTO and your partition is not large enough. See "VSE Storage Requirements" on page 4-16 before coding this parameter.
`SYSEXN`	Requests the System Exception report series.
`HIST`	The input records are on the working history data set, rather than SYSREC.
`ACC=N`	Want to keep them there; the same records are being used for all the reports.
`TABSIZE=100K`	The System Exception series requires a larger sort table than other reports; see "VSE Storage Requirements" on page 4-16.
`ENDPARM`	End of parameters; control statements follow.
`DASDID . . .`	For DASD without physical IDs.
`LIMIT . . .`	Limiting the number of records in the report.
`SHARE . . .`	For tape drives shared between processors.
`/*`	End of instream data.
`// MTC REW,SYS008`	Rewind the tape holding the history data set.

MCH and CCH Detail Reports

Print Detail Edits and Summaries of all machine and channel checks.

`// EXEC IFCEREP1,SIZE=AUTO`	Partition size must be large enough to accomodate EREP and a sort table. You may encounter storage problems if you code SIZE=AUTO and your partition is not large enough. See "VSE Storage Requirements" on page 4-16 before coding this parameter.
`PRINT=PS`	Requesting Detail Edits and Summaries of the input records.
`TYPE=MC`	Selecting the records by type: M MCH C CCH
`HIST`	Input records are in the history data set.
`ACC=N`	Leave them there.
`TABSIZE=50K`	Default is 4K; not enough for detail processing of specific record types.
`/*`	End of instream data.
`// MTC REW,SYS008`	Rewind the tape holding the history data set.

Threshold Summary

A summary of tape I/O errors:

- all permanent errors
- temporary errors that exceed 1 read or 15 writes.

Includes records from 3410, 3420, and 8809 devices.

`// EXEC IFCEREP1,SIZE=AUTO`	Partition size must be large enough to accomodate EREP and a sort table. You may encounter storage problems if you code SIZE=AUTO and your partition is not large enough. See "VSE Storage Requirements" on page 4-16 before coding this parameter.
`THRESHOLD=(001,015)`	Requesting a Threshold Summary.
`HIST`	Input records are in the history data set.
`ACC=N`	Leave them there.
`TABSIZE=50K`	Default is 4K.
`ENDPARM`	End of parameters; control statements follow.
`SHARE . . .`	For devices shared between processors.
`/*`	End of instream data.
`// MTC REW,SYS008`	Rewind the tape holding the history data set.

Detail Reports for Communications Devices

Print Detail Summaries of all errors for 3704, 3705 and 3725 communications controllers.

`// EXEC IFCEREP1,SIZE=AUTO`	Partition size must be large enough to accomodate EREP and a sort table. You may encounter storage problems if you code SIZE=AUTO and your partition is not large enough. See "VSE Storage Requirements" on page 4-16 before coding this parameter.
`PRINT=SU`	Requesting only Detail Summaries.
`TYPE=OT`	Selecting the records by type: O OBR T MDR
`DEV=(3704,3705,3725)`	Selecting by device type; the communications controllers.
`HIST`	Input records are in the history data set.
`ACC=N`	Leave them there.
`TABSIZE=50K`	Default is 4K.
`ENDPARM`	End of parameters; control statements follow.
`SHARE . . .`	For devices shared between processors.
`/*`	End of instream data.
`// MTC REW,SYS008`	Rewind the tape holding the history data set.

Detail Summaries for I/O Errors

Print Detail Summaries of all I/O errors not already covered in the preceding reports.

`// EXEC IFCEREP1,SIZE=AUTO`	Partition size must be large enough to accomodate EREP and a sort table. You may encounter storage problems if you code SIZE=AUTO and your partition is not large enough. See "VSE Storage Requirements" on page 4-16 before coding this parameter.
`PRINT=SU`	Requesting only Detail Summaries.
`TYPE=DOTH`	Selecting the records by type: D DDR O OBR T MDR H MIH
`DEV=(N34XX,N3704,N3705,N3725)`	Selecting by device type; excluding those already covered. Note the absence of N33XX; EREP does not produce detail summaries for 33XX DASD anyway, so they need not be excluded.
`HIST`	Input records are in the history data set.
`ACC=N`	Leave them there.
`TABSIZE=50K`	Default is 4K.
`ENDPARM`	End of parameters; control statements follow.
`SHARE . . .`	For devices shared between processors.
`/*`	End of instream data.
`// MTC REW,SYS008`	Rewind the tape holding the history data set.

Detail Reports for Software Records

Print Detail Edits and Summaries of all software and operational records.

`// EXEC IFCEREP1,SIZE=AUTO`	Partition size must be large enough to accomodate EREP and a sort table. You may encounter storage problems if you code SIZE=AUTO and your partition is not large enough. See "VSE Storage Requirements" on page 4-16 before coding this parameter.
`PRINT=PS`	Requesting both Detail Edits and Summaries.
`TYPE=SIE`	Selecting the records by type: S SIE I IPL E EOD (Termination)
`HIST`	Input records are in the history data set.
`ACC=N`	Leave them there.
`TABSIZE=50K`	Default is 4K.
`/*`	End of instream data.
`// MTC REW,SYS008`	Rewind the tape holding the history data set.

Event History Report

One-line abstracts of all records, in chronological order.

`// EXEC IFCEREP1,SIZE=AUTO`	Partition size must be large enough to accomodate EREP and a sort table. You may encounter storage problems if you code SIZE = AUTO and your partition is not large enough. See "VSE Storage Requirements" on page 4-16 before coding this parameter.
`EVENT`	Requesting an Event History. Note the absence of selection parameters; the report is to include every record.
`HIST`	Input records are in the history data set.
`ACC=N`	Leave them there.
`TABSIZE=50K`	Default is 4K.
`/*`	End of instream data. Event History uses no control statements.
`// MTC RUN,SYS008`	Rewind and unload the tape holding the history data set.

Update the History Tape

Copy the records on the input history tape (SYS008) to the permanent history tape (SYS009, as EREP.HIST.TAPE) either creating or updating it.

`// TLBL HISTOT,'EREP.HIST.TAPE',nnn`	The output (ACCDEV) data set.
`// TLBL HISTINT,'EREP.HIST.TAPE',nnn`	The input (ACCIN) data set.
`// PAUSE  MOUNT SCRATCH TAPE AND ASSGN SYS009`	Instructions for the operator; assigning the ACCDEV logical unit.
`// PAUSE  MOUNT EREP.HISTORY.TAPE AND ASSGN SYS008`	Instructions for the operator; assigning the ACCIN logical unit.
`// EXEC IFCEREP1,SIZE=AUTO`	Partition size must be large enough to accomodate EREP and a sort table. You may encounter storage problems if you code SIZE=AUTO and your partition is not large enough. See "VSE Storage Requirements" on page 4-16 before coding this parameter.
`PRINT=NO`	Requesting no report output.
`MERGE`	Merge the records from the old history data set and SYSREC, which contains the latest data because it was updated while the first 8 steps were being run.
`ACC=Y`	Write the combined records to the ACCDEV tape.
`ZERO=Y`	Clear SYSREC.
`/*`	End of instream data.
`// MTC REW,SYS009`	Rewind the ACCDEV tape; it is the input for the final step.
`// MTC RUN SYS008`	Rewind and unload the old history (ACCIN) tape.

Trends Report

Print a Trends report covering the last 30 days of records from the newly updated history tape.

`// TLBL HISTINT,'EREP.HIST.TAPE',nnn` `// ASSGN SYS008,SYS009`	Defining the former ACCDEV data set as ACCIN; the updated history tape is now being used as input.
`// EXEC IFCEREP1,SIZE=AUTO`	Partition size must be large enough to accomodate EREP and a sort table. You may encounter storage problems if you code SIZE=AUTO and your partition is not large enough. See "VSE Storage Requirements" on page 4-16 before coding this parameter.
`TRENDS`	Requesting the Trends report. Without the DATE parameter, Trends uses the last 30 days of records.
`HIST`	Input records are still in a history data set rather than SYSREC.
`ACC=N`	Leave them there.
`TABSIZE=50K`	Default is 4K.
`ENDPARM`	End of parameters; control statements follow.
`SHARE . . .`	For devices shared between processors.
`/*`	End of instream data.
`// MTC RUN,SYS008`	Rewind and unload the monthly history tape.
`/&`	

VSE System Controls

In addition to the parameters and controls EREP needs to process records and produce reports, VSE requires system controls that create the interface between EREP and the operating system's data management functions. You provide these as part of the EREP run, as follows:

// JOB JOBNAME

This statement notifies the operating system of the EREP job.

// TLBL HISTINT or // DLBL HISTIND
(// EXTENT SYS008,xxxx,1,,xxxx,x, for DASD)

Defines the tape (TLBL) or DASD (DLBL) input history data set.

The EXTENT statement is only required if the history data set resides on DASD. See the appropriate VSE publications for information on coding EXTENT statements.

Note: The input history tape must have a standard label, and the blocksize for the input history data set cannot exceed 4000.

IECEREP1, the EREP program, retrieves all or selected records from the history data set (and from SYSREC, if you code MERGE) and writes them to an output device supported by the sequential access method (SAM).

You can use the history data set or SYSREC, or both, as input to EREP, but you must use at least one of them. If you do use the history data set, you must also code the following:

// ASSGN SYS008,cuu

Assigns the history data set to a logical unit, which is at address "cuu" (one-digit channel, two-digit unit address).

The logical units are consistent for the various EREP data sets; history input is always assigned to SYS008.

// DLBL IJSYS01
// EXTENT SYS001,xxxx,1,,xxxx,xx

Defines the DASD temporary work data set required when you use history input for a report. See the appropriate VSE manual for information on the EXTENT statement.

SYS001 should already be defined for the linkage editor, so these two statements may not be necessary. The standard SYS001 EXTENT should provide enough space for most IFCEREP1 activity; but you must be sure there is enough space to hold all the records selected from the input data set(s).

This data set is needed only if you are requesting a printed report.

// ASSGN SYS001,cuu

Assigns the work data set (DIRECTWK) to logical unit SYS001 at address "cuu" (One-digit channel, two-digit unit address). SYS001 must be a single-extent data set. See "DASD Space for SYS001" on page 4-18 for more information about space for this data set.

// TLBL HISTOT or DLBL HISTOD
(// EXTENT SYS009,xxxx,1,,xxxx,xx, for DASD)

Defines the tape (TLBL) or DASD (DLBL) output history data set; the ACCDEV data set. If you code or imply ACC=Y, you must code this label statement and the following ASSGN statement, so EREP will know where to put the records it accumulates.

The output history tape must have a standard label.

// ASSGN SYS009,cuu

Assigns the output (ACCDEV) data set to logical unit SYS009.

The following assignments should have been made when the partition was initialized; otherwise, EREP cannot run. If for some reason they were not, you must re-IPL in order to make the assignments. Issuing ASSGN statements for any of these logical units while running EREP results in a JCL error.

// ASSGN SYSREC,cuu

Assigns the ERDS to system logical unit SYSREC. SYSREC is the system recording file on the system residence volume.

SYSREC is the default input data set for EREP. Unless you indicate that there is history input (EREP parameter HIST or MERGE, and the HISTINT or HISTIND label statement), EREP takes the records for the report you request from SYSREC.

// ASSGN SYSIPT,cuu

You code EREP parameters and control statements following the EXEC statement, as instream data. The system reads the data from the SYSIPT logical unit, so this ASSGN statement is always required.

// ASSGN SYSLST,cuu

Assigns the data set for EREP (TOURIST) messages and EREP report output to system logical unit SYSLST. See "The TOURIST Output" on page 1-4 for information about TOURIST output.

// ASSGN SYSLOG,cuu

Assigns the system log data set, required in case SYSLST is not available, to system logical unit SYSLOG.

Having set up the data sets required for EREP, you can now execute the program.

// EXEC PGM=IFCEREP1,SIZE=xxxK or SIZE=AUTO
Executes the EREP program. Note the use of the SIZE parameter; this may be necessary to make sure there is enough storage to hold the EREP program and its sort tables. EREP issues the GETVIS macro to obtain storage for its own modules and its sort tables. In addition to the program storage required for the initial EREP module, GETVIS requires 1K for each 100 records over 400. See "VSE Storage Requirements."

VSE Storage Requirements

EREP requires at least 100K of virtual storage. This provides for a sort table of 4K bytes, the VSE TABSIZE default. The 4K-byte sort table permits the processing of approximately 400 records for a report.

Increasing Partition Size

If you have to increase the size of the sort table using the TABSIZE EREP parameter, you will probably have to increase the size of your virtual partition, too, by the amount you specify for the TABSIZE value minus 4K bytes.

In addition, *EREP can use two different sorting algorithms for its reports; the faster one requires additional storage equal to TABSIZE.* If you can increase your partition size by the value of TABSIZE over the requirements outlined in Figure 4-1 on page 4-17, you will significantly enhance EREP's performance. EREP always tries to obtain the extra storage, and uses the faster sort routine if the storage is available.

Several cases require you to increase the partition size when running EREP. Figure 4-1 shows these cases and recommended amounts of partition increase for each.

INFLUENCING FACTOR	*AMOUNT OF PARTITION INCREASE*
You are using the TABSIZE parameter	The specified value of TABSIZE minus 4K bytes
You are including EREP control statements	The specified or default value of TABSIZE
You are using any of the following selection parameters: CPU CPUCUA CUA DEV DEVSER LIA/LIBADR MOD SYMCDE VOLID	4K bytes for any or all
You are requesting Detail Edit reports (PRINT=PT, PS or AL)	4K bytes for each processor
You are requesting a Detail Summary of 33XX records (not available in EREP 2.3 and later releases).	7K bytes
A processor requires frames for Detail Edit output	150K bytes
You are requesting the System Exception report series	6 times the specified or default value of TABSIZE

Figure 4-1. VSE Partition Size Increases for EREP

Because you might not know how many input records to expect, these amounts of partition size increase are generous. Depending on installation restrictions, you may want to code SIZE=AUTO on the EXEC statement for the cases that require a lot of virtual storage. In rare instances, this may create a storage problem. There are several ways to correct this problem:

- Increase the partition size to 1.7M or larger
- Instead of SIZE=AUTO code SIZE=xxxK where xxx is 100 plus 1 for each 100 records over 400. For example, for 900 records you would code SIZE=105K.
- Do not code the SIZE parameter at all.

Most of the steps in the VSE master example earlier in this chapter use SIZE=AUTO.

DASD Space for SYS001

In addition to the virtual storage needed to run EREP, VSE also requires DASD space for a temporary work data set whenever there are input records on a history data set. You request this auxiliary storage space on the DLBL/EXTENT statements for IJSYS01, assigned to SYS001.

If the SYS001 logical unit has already been assigned, check the space allocation on the EXTENT statement to make sure there is enough room for the input records. The amount of storage space required depends on the device type and the number of records being processed; Figure 12-8 on page 12-11 shows the approximate capacities of different types of DASD.

VSE Notes

The following short notes are meant to help you avoid potential problems as you create your EREP run for VSE.

Access Methods

EREP retrieves error records from SYSREC both sequentially, through the SAM access method, and randomly, through the EXCP (Execute Channel Program) system macro.

It writes records to an output data set or buffer sequentially, through SAM. If you request specific devices for EREP's output data, they must be supported by SAM.

Special Considerations for EREP Parameters and Controls

- You may only specify EREP parameters and control statements on input statements (as instream data; read from SYSIPT).
- If you want the latest statistical and usage data included in the reports, the operator must issue the ROD (record on demand) command before running EREP against SYSREC, to force the system to dump the in-core and buffer counters to SYSREC before EREP begins its processing.
- If VSE message OP77I appears after an EREP job is submitted, increase the SIZE parameter value on the EXEC card. It might also be necessary to increase the partition size. See "VSE Storage Requirements" on page 4-16.

Running EREP under MVS

Coding the JCL

There are several ways to code the JCL statements for your EREP run. The following examples will illustrate some of the more common of these. You should consult the JCL manual for your MVS system for further restrictions on format.

1. Coding the parameters on the EXEC statement. Control statements are in the file specified on the SYSIN statement.

```
//STEP     EXEC  PGM=IFCEREP1,PARM=('PRINT=PS,HIST,ACC=N')
//ACCIN      DD  DSN=EREP.HISTORY,DISP=(OLD,PASS)
//DIRECTWK   DD  UNIT=SYSDA,SPACE=(CYL,5,,CONTIG)
//EREPPT     DD  SYSOUT=A,DCB=BLKSIZE=133
//TOURIST    DD  SYSOUT=A,DCB=BLKSIZE=133
//SYSIN      DD  DSN=EREP.CNTRL,DISP=(OLD,PASS)
```

   The EXEC statement may be coded with or without the parentheses and with a single set of quotes ***only*** if all of the parameters fit on one line.

```
//STEP     EXEC  PGM=IFCEREP1,PARM='PRINT=PS,HIST,ACC=N'
```

   If the parameters will not fit on one line then parentheses and individual quotes are required.

```
//STEP     EXEC  PGM=IFCEREP1,PARM=('PRINT=PS',HIST,
//                                  'ACC=N','TYPE=OT')

//STEP     EXEC  PGM=IFCEREP1,PARM=('PRINT=PS',HIST,'ACC=N')
```

   Using the parentheses and the individual quotes is the preferred method.

2. Coding PARM='CARD' and entering the parameters and the control statements on the SYSIN statement.

```
//STEP     EXEC  PGM=IFCEREP1,PARM='CARD'
//ACCIN      DD  DSN=EREP.HISTORY,DISP=(OLD,PASS)
//DIRECTWK   DD  UNIT=SYSDA,SPACE=(CYL,5,,CONTIG)
//EREPPT     DD  SYSOUT=A,DCB=BLKSIZE=133
//TOURIST    DD  SYSOUT=A,DCB=BLKSIZE=133
//SYSIN      DD  *
PRINT=PS
HIST
ACC=N
TYPE=OT
ENDPARM
SHARE  . . .
LIMIT  . . .
DASDID  . . .
CONTROLLER  . . .
/*
```

All of the EREP parameters are described in detail in Chapter 7, "EREP Parameters"; the EREP control statements in Chapter 8, "EREP Control Statements."

The History Dataset

You should always create a history data set before running your EREP reports. This insures consistency across the reports.

The operating system adds records to the ERDS continually. If you run your reports against the ERDS (HIST=N) then the set of records used by the first report is not the same as the set used by the next report since records could have been added while the report was running. By the time the last report is run the set of records may be quite different.

By creating a history data set and then running all of the reports against that data set you insure that all of the reports are using the same set of records.

The JCL on pages 4-22 through 4-30 shows an MVS job with several steps. The first step creates a history data set which is used in the remainder of the steps. ***This is only an example.*** You must decide which reports are relevent to your installation, what order they should be generated in and how often they should be run.

Create a History Dataset

Create a history data set to be used in later report generation:

- Generate a System Summary report from the records in SYS1.LOGREC
- Copy the records from LOGREC to another disk data set
- Zero LOGREC.

```
//EREPDAY  JOB  [accounting information] . . . MSGLEVEL=1
```

`//STEP1   EXEC  PGM=IFCEREP1,PARM=(SYSUM,` `//             'ACC=Y',ZERO)`	Requests a System Summary. The records are to be copied from LOGREC to the ACCDEV data set and LOGREC is to be zeroed out. When you code ACC=Y with SYSUM, EREP always clears LOGREC, even if you code ZERO=N.
`//SERLOG    DD  DSN=SYS1.LOGREC,DISP=OLD`	Using the records in LOGREC. If you omit this DD statement and do not code ACCIN/HIST or MERGE, EREP still uses LOGREC as input.

`//ACCDEV    DD  UNIT=SYSDA,DSN=EHISTORY,`
`//             DISP=(NEW,PASS,CATLG),`
`//             SPACE=(CYL,(5,5)),`
`//             DCB=(RECFM=VB,BLKSIZE=4000)`
	This data set will be used as input for the rest of reports in EREPDAY.

`//EREPPT    DD  SYSOUT=A,DCB=BLKSIZE=133` `//TOURIST   DD  SYSOUT=A,DCB=BLKSIZE=133`	EREP requires a 132-position printer or both could be sent to a display device for online viewing.
`//SYSIN     DD  DSN=EREP.CONTROLS,` `//             DISP=(OLD,PASS)`	This data set contains the EREP control statements needed for all the reports. EREP uses only those that apply to the report requested in this step.
`/*`	This delimiter is optional; see the JCL manual for your MVS system.

System Exception Reports

Generate the series of System Exception reports for processors, channels, and DASD and tape subsystems:

- A System Error Summary in two parts:
  1. IPL/restarts, machine and channel checks
  2. DASD and tape I/O errors across processors.
- A Subsystem Exception report for each processor
- A Subsystem Exception report on channel failures, one for each processor
- A Subsystem Exception report on DASD failures
- A Subsystem Exception report on tape failures
- A series of detailed summaries for DASD presenting data by symptom code, storage control unit, string, and volume
- A series of detailed summaries for 34XX tape devices presenting permanent and temporary errors by CUA/device number and volume.

```
//STEP2    EXEC  PGM=IFCEREP1,REGION=4K,
//               PARM=(SYSEXN,'TABSIZE=512K',
//               HIST,'ACC=N')
```

The System Exception series requires a large sort table (TABSIZE), and thus more storage (REGION). The input records are in the history data set, and are to stay there. The DASDID, LIMIT, and SHARE statements needed for the System Exception reports are in the data set named on the SYSIN DD statement.

```
//ACCIN      DD  DSN=EHISTORY,DISP=(OLD,PASS)
```

The data set created in STEP1. Multiple history data sets can be concatenated on this DD statement.

```
//DIRECTWK   DD  UNIT=SYSDA,
//               SPACE=(CYL,5,,CONTIG)
```

Workspace for EREP, required with history input.

```
//EREPPT     DD  SYSOUT=A,DCB=BLKSIZE=133
//TOURIST    DD  SYSOUT=A,DCB=BLKSIZE=133
```

EREP requires a 132-position printer or both could be sent to a display device for online viewing.

```
//SYSIN      DD  DSN=EREP.CONTROLS,
//               DISP=(OLD,PASS)
/*
```

This data set contains the EREP control statements for all the reports. EREP uses only those that apply to the report requested in this step.

MCH and CCH Detail Reports

Print Detail Edits and Summaries of all machine and channel checks.

```
//STEP3    EXEC  PGM=IFCEREP1,PARM=('PRINT=PS',
//               'TYPE=MC',HIST,'ACC=N')
```

Requesting both Detail Edits and Summaries. Selecting the records by type; MCH (M) and CCH (C). Input records are in the history data set, and are to stay there.

```
//ACCIN      DD  DSN=EHISTORY,DISP=(OLD,PASS)
```

The history data set.

```
//DIRECTWK   DD  UNIT=SYSDA,
//               SPACE=(CYL,5,,CONTIG)
```

Workspace for EREP, required with history input.

```
//EREPPT     DD  SYSOUT=A,DCB=BLKSIZE=133
//TOURIST    DD  SYSOUT=A,DCB=BLKSIZE=133
```

EREP requires a 132-position printer or both could be sent to a display device for online viewing.

```
//SYSIN      DD  DSN=EREP.CONTROLS,
//               DISP=(OLD,PASS)
```

Control statements for all of the reports. EREP uses only those that apply to the report requested in this step.

```
/*
```

This delimiter is optional; see the JCL manual for your MVS system.

Threshold Summary

A summary of tape I/O errors:

- all permanent errors
- temporary errors that exceed 1 read or 15 writes.

Includes records from 3410, 3420, and 8809 devices.

```
//STEP4    EXEC  PGM=IFCEREP1,
//               PARM=('THRESHOLD=(001,015)',
//               HIST,'ACC=N')
```

Requesting a Threshold Summary. The input records are in the history data set, and are to stay there. The default value for ACC with THRESHOLD is N — an exception to the rule — but it is wise to code the parameter anyway.

```
//ACCIN      DD  DSN=EHISTORY,DISP=(OLD,PASS)
```

History data set.

```
//DIRECTWK   DD  UNIT=SYSDA,
//               SPACE=(CYL,5,,CONTIG)
```

Workspace for EREP, required with history input.

```
//EREPPT     DD  SYSOUT=A,DCB=BLKSIZE=133
//TOURIST    DD  SYSOUT=A,DCB=BLKSIZE=133
```

EREP requires a 132-position printer or both could be sent to a display device for online viewing.

```
//SYSIN      DD  DSN=EREP.CONTROLS,
//               DISP=(OLD,PASS)
```

Contains EREP control statements for all of the reports. EREP uses only those that apply to the report requested in this step.

```
/*
```

This delimiter is optional; see the JCL manual for your MVS system.

Detail Reports for Communications Devices

Print Detail Summaries of all errors for 3704, 3705 and 3725 communications controllers.

`//STEP5    EXEC PGM=IFCEREP1,PARM=('PRINT=SU',`
`//             'TYPE=OT',`
`//             'DEV=(3704,3705,3725)',`
`//             HIST,'ACC=N')`
	Requesting only Detail Summaries only. Selecting the records by type; OBR (O) and MDR (T). Selecting the records by device; the communications controllers. The input records are in the history data set, and are to stay there.

`//ACCIN     DD  DSN=EHISTORY,DISP=(OLD,PASS)`	History data set.
`//DIRECTWK  DD  UNIT=SYSDA,` `//             SPACE=(CYL,5,,CONTIG)`	Workspace for EREP, required with history input.
`//EREPPT    DD  SYSOUT=A,DCB=BLKSIZE=133` `//TOURIST   DD  SYSOUT=A,DCB=BLKSIZE=133`	EREP requires a 132-position printer or both could be sent to a display device for online viewing.
`//SYSIN     DD  DSN=EREP.CONTROLS,` `//             DISP=(OLD,PASS)`	Contains the control statements for all the reports. EREP uses only those that apply to the report requested in this step.
`/*`	This delimiter is optional; see the JCL manual for your MVS system.

Detail Summaries for I/O Errors

Print Detail Summaries of all I/O errors not already covered in the preceding reports.

```
//STEP6   EXEC PGM=IFCEREP1,PARM=('PRINT=SU',
//             'TYPE=DOTH',
//             'DEV=(N34XX,N3704,N3705,N3725)',
//             HIST,'ACC=N')
```

Requesting only Detail Summaries. Selecting the records by type: DDR (D), OBR (O), MDR (T) and MIH (H). Selecting by device type; excluding those already covered. Note the absence of N33XX; EREP does not produce detail summaries for 33XX DASD anyway, so they need not be excluded. The input records are in the history data set, and are to stay there.

```
//ACCIN    DD DSN=EHISTORY,DISP=(OLD,PASS)
```

History data set.

```
//DIRECTWK DD UNIT=SYSDA,
//            SPACE=(CYL,5,,CONTIG)
```

Workspace for EREP, required with history input.

```
//EREPPT   DD SYSOUT=A,DCB=BLKSIZE=133
//TOURIST  DD SYSOUT=A,DCB=BLKSIZE=133
```

EREP requires a 132-position printer or both could be sent to a display device for online viewing.

```
//SYSIN    DD DSN=EREP.CONTROLS,
//            DISP=(OLD,PASS)
```

Contains the control statements for all the reports. EREP uses only those that apply to the report requested in this step.

```
/*
```

This delimiter is optional; see the JCL manual for your MVS system.

Detail Reports for Software Records

Print Detail Edits and Summaries of all software and operational records.

`//STEP7    EXEC  PGM=IFCEREP1,PARM=('PRINT=PS',` `//              'TYPE=SIE',HIST,'ACC=N')`	Requesting both Detail Edits and Summaries. Selecting the records by type; SFT (S), IPL (I) and EOD (E). Input records are in the history data set, and are to stay there.
`//ACCIN      DD  DSN=EHISTORY,DISP=(OLD,PASS)`	History data set.
`//DIRECTWK   DD  UNIT=SYSDA,` `//              SPACE=(CYL,5,,CONTIG)`	Workspace for EREP, required for history input.
`//EREPPT     DD  SYSOUT=A,DCB=BLKSIZE=133` `//TOURIST    DD  SYSOUT=A,DCB=BLKSIZE=133`	EREP requires a 132-position printer or both could be sent to a display device for online viewing.
`//SYSIN      DD  DUMMY`	EREP control statements do not apply to software records, so the SYSIN data set is not needed.
`/*`	This delimiter is optional; see the JCL manual for your MVS system.

Event History Report

One-line abstracts of all records, in chronological order.

`//STEP8    EXEC  PGM=IFCEREP1,PARM=(EVENT,` `//              HIST,'ACC=N')`	Requesting an Event History Report. Note the absence of selection parameters; the report is to include every record. The input records are in the history data set, and are to stay there.
`//ACCIN      DD  DSN=EHISTORY,DISP=(OLD,PASS)`	History data set.
`//DIRECTWK   DD  UNIT=SYSDA,` `//              SPACE=(CYL,5,,CONTIG)`	Workspace for EREP, required with history input.
`//EREPPT     DD  SYSOUT=A,DCB=BLKSIZE=133` `//TOURIST    DD  SYSOUT=A,DCB=BLKSIZE=133`	EREP requires a 132-position printer or both could be sent to a display device for online viewing.
`//SYSIN      DD  DUMMY`	The Event History cannot use SHARE or CONTROLLER control statements, so the SYSIN data set is not needed.
`/*`	This delimiter is optional; see the JCL manual for your MVS system.

Updating History Tape

1. Copy the records on the input data set (EHISTORY) to the history tape (EREP.HIST.TAPE), either creating or updating it.
2. Delete the EHISTORY data set; the updated history tape is the input for the final step.

`//STEP`	`EXEC`	`PGM=IFCEREP1,PARM=('PRINT=NO',`	Requesting no report output, but the records are to be copied ("accumulated"; ACC=Y) from the working data set to the output data set named on the ACCDEV DD statement.
`//`		`HIST,'ACC=Y')`	
`//ACCIN`	`DD`	`DSN=EHISTORY,DISP=(OLD,PASS)`	Input records are in the history data set.
`//ACCDEV`	`DD`	`DSN=EREP.HIST.TAPE,`	The output data set to receive the records. If it did not already exist, DISP=(NEW,PASS) would create it.
`//`		`DISP=(MOD,PASS),VOL=(,RETAIN),`	
`//`		`DCB=(RECFM=VB,BLKSIZE=12000)`	
`//DIRECTWK`	`DD`	`UNIT=SYSDA,`	Workspace for EREP, required for history input.
`//`		`SPACE=(CYL,5,,CONTIG)`	
`//EREPPT`	`DD`	`DUMMY`	No printed report this time.
`//TOURIST`	`DD`	`SYSOUT=A,DCB=BLKSIZE=133`	Do want to see the messages, however. EREP requires a 132-position printer or the messages could be sent to a display device for online viewing.
`//SYSIN`	`DD`	`DUMMY`	No report, no control statements.
`/*`			This delimiter is optional; see the JCL manual for your MVS system.

Trends Report

Print a Trends report covering the last 30 days of records from the newly updated history tape.

`//STEP10   EXEC  PGM=IFCEREP1,PARM=(TRENDS,` `//              HIST,'ACC=N')`	Requesting the Trends report. Without the DATE parameter, Trends uses the last 30 days of records. The history data set referred to here is the one updated or created in Step 9.
`//ACCIN      DD  DSN=EREP.HISTORY.TAPE,` `//              DISP=(OLD,KEEP)`	The "new" input data set, containing the records EREP is to use for the Trends report.
`//DIRECTWK   DD  UNIT=SYSDA,` `//              SPACE=(CYL,5,,CONTIG)`	Workspace for EREP, required for history input.
`//EREPPT     DD  SYSOUT=A,DCB=BLKSIZE=133` `//TOURIST    DD  SYSOUT=A,DCB=BLKSIZE=133`	EREP requires a 132-position printer or both could be sent to a display device for online viewing.
`//SYSIN      DD  DSN=EREP.CONTROLS,` `//              DISP=(OLD,KEEP)`	The SHARE and CONTROLLER control statements needed for the Trends report are in this data set.
`/*`	This delimiter is optional; see the JCL manual for your MVS system.

MVS System Controls

In addition to the parameters and controls EREP needs to process records and produce reports, the MVS systems require system controls (JCL) that create the interface between EREP and the operating system's data management functions. You provide these as part of the EREP run, as follows:

//JOB statement
Required; initiates the job. May include the REGION parameter, to increase the virtual storage available to EREP via GETMAIN.

//EXEC statement
Required; executes the EREP program. This is one place to request more storage to accommodate EREP; use the REGION parameter. You may include the EREP keyword parameters (PARM='xxx=x','yyy=y') here or indicate that they are being entered as SYSIN data (PARM='CARD'). Because of fairly intricate syntax requirements when coding them here, many users prefer the PARM='CARD'/instream SYSIN data method. However, see the discussion of the SYSIN DD statement (see page 4-33).

//ACCIN DD statement
Required with history input; defines the history input data set.

IFCEREP1, the EREP program, retrieves records from the history data set (and/or from SYS1.LOGREC), and writes them to any QSAM-supported output device. See "MVS Notes" on page 4-36 for more information about LOGREC processing.

The history input can be in more than one data set; you can concatenate the DD statements, making sure the record formats (RECFM) are either blocked or unblocked but not both. The data set with the largest blocksize must be first in the concatenation, so the system will allocate a device suitable for all the data sets.

See "MVS Notes" on page 4-36 for more information about the DCB requirements for this data set.

//DIRECTWK DD statement
Required with history input. Defines and allocates DASD space for the temporary work data set needed to process history (ACCIN) input records.

//SERLOG DD statement
Required with LOGREC input in older MVS systems. Defines SYS1.LOGREC, the system error recording data set, as the input data set. You may use LOGREC or a history data set as input, but you must use at least one of them for every execution of EREP.

LOGREC is the default input data set for EREP under MVS. Unless you indicate that there is history input (EREP parameter HIST or MERGE, and the ACCIN DD statement), EREP takes the records for the report you request from LOGREC. When you code the MERGE parameter, you must also code both the ACCIN and SERLOG DD statements.

//ACCDEV DD statement

Defines and allocates space for the output history data set. You only need this DD statement if you want EREP to accumulate the records to an output data set after completing the report (ACC=Y). The data set can reside on DASD or magnetic tape.

Note: If you code this DD statement as ACCDEV DD DUMMY and then default to or specify ACC=Y, you are telling EREP to write the records to a nonexistent data set.

When you first define this data set, you will have to include some DCB information on the DD statement. See the JCL manual for your MVS system for information on how to do this.

//EREPPT DD statement

Defines and allocates space for the output data set that holds the EREP report. You must code this DD statement whenever you request a printed report. The output device need not be a printer, however; specifying the SYSOUT class for online display allows you to view the report at a TSO terminal.

//TOURIST DD statement

Defines and allocates space for the output data set that holds EREP messages and processing information. You must code the TOURIST DD statement whenever you run EREP. You can send the TOURIST output to the SYSOUT class, or let it default to the message class for the job, or spool it to a JES device.

//SYSIN DD statement

Defines the data set you use to enter EREP controls as instream data.

This data set is always required for EREP control statements; they must be coded as SYSIN data. You can include EREP parameters as well, if you code PARM='CARD' on the EXEC statement and separate the parameters from the control statements with ENDPARM. See "Coding Control Statements" on page 8-3 and "Coding EREP Parameters" on page 7-1 for how to do this.

Because control statements are fairly complicated to code, it is a good idea to put them all into a separate data set and name the data set on the DSN parameter. EREP uses only those control statements that apply to a requested report, so you need not put SHARE/CONTROLLER statements in one data set and DASDID/LIMIT statements in another (although it is possible to do so, concatenating the DD statements for the two data sets.) The sample EREP setups earlier in this chapter use this method of entering control statements.

Even when you have no control statements or parameters to enter as SYSIN data, it is wise to supply a SYSIN DD statement; the operating system expects one. Code it as `//SYSIN DD DUMMY`.

MVS Storage Requirements

EREP requires at least 100K bytes of virtual storage for its internal sort table. 100K provides for a 24K-byte sort table, the MVS TABSIZE default. EREP can process approximately 2400 records in a 24K sort table. Depending on the kind of report you are running, and on the number of records involved, you might have to increase the sort table size for a single EREP run or for all your EREP reports. The System Exception series, for example, requires a large sort table; the recommended value for TABSIZE when requesting SYSEXN in a large installation is 512K.

Increasing Region Size

If you increase the size of the sort table using the TABSIZE processing parameter, you might have to increase the virtual storage (region) size as well, using the REGION parameter on either the JOB or EXEC statement. See the JCL manual for your MVS system for how to do this.

In addition, *EREP can use two different sorting algorithms for its reports; the faster one requires additional storage equal to TABSIZE.* If you can increase your region size by the value of TABSIZE over the requirements outlined in Figure 4-2 on page 4-35, you will significantly enhance EREP's performance. EREP always tries to obtain the additional storage, and uses the faster sort routine if the storage is available.

Several conditions can require you to increase the region size when running EREP. Figure 4-2 shows these conditions and recommended amounts of region increase for each.

INFLUENCING FACTOR	*AMOUNT OF REGION INCREASE*
You are using the TABSIZE parameter	The specified value of TABSIZE minus 4K bytes
You are using EREP control statements	The specified or default value of TABSIZE
You are requesting the System Exception report series	6 times the specified or default value of TABSIZE
You are using any of the following selection parameters: CPU CPUCUA CUA DEV DEVSER LIA/LIBADR MOD SYMCDE VOLID	4K bytes for any or all
You are requesting Detail Edit reports (PRINT=PT, PS or AL)	4K bytes for each processor
You are requesting a Detail Summary of 33XX records (not available in EREP 2.3 and later releases.)	7K bytes
A processor requires frames for Detail Edit output	150K bytes

Figure 4-2. MVS Region Size Increases for EREP

DASD Storage for DIRECTWK

In addition to the virtual storage needed to run EREP, MVS also requires DASD space for EREP's temporary work data set whenever your input includes records on a history data set. You request this storage using the SPACE parameter on the DIRECTWK DD statement, making sure the storage is in a contiguous block (SPACE=(CYL,5,,CONTIG)).

The amount of storage depends on the device type and the number of records to be processed. Figure 12-8 on page 12-11 shows the capacities of different types of DASD.

MVS Notes

The following notes are meant to help you avoid potential problems as you create the interface between EREP and your system control program.

Access Methods

EREP retrieves error records from SYS1.LOGREC both sequentially, through the QSAM access method, and randomly, through the MVS system macro EXCP (Execute Channel Program).

It writes records to an output data set or buffer sequentially, through QSAM. If you request specific devices for EREP's output data, they must be supported by QSAM.

DCB Requirements

The DD statements you code in the JCL for an EREP run are there to define and allocate storage for the data sets EREP uses in its processing.

If you wish to add DCB attributes to the other information on the DD statement, make sure they are really needed to override the DCB information already built by the system. This is described in *SPL: Data Management* and the JCL manual for your system. Following is some specific information you may need when first setting up your EREP run.

Both the input (ACCIN) and output (ACCDEV) data sets have special DCB requirements.

ACCIN You must supply the RECFM and BLKSIZE values if:

- the data set resides on an unlabeled tape volume or
- the data set is not included in a DSCB (data set control block)

ACCDEV You must supply the RECFM and BLKSIZE values for this data set if it is not included in a DSCB.

RECFM=VB and BLKSIZE=12000 for tape, and full track blocking for DASD, give the best performance results.

The blocksize for a tape data set must be at least 2004.

A blocksize of 6144 for a DASD data set allows for the various blocking factors among DASD, and improves performance.

In addition to these, you can supply a blocksize for the SYSIN data set; for the usual input media — cards or card images on a terminal screen — the BLKSIZE value is 80.

Running EREP in a Multisystem Environment

If you do not combine the error data from all of the systems that share I/O devices, you will not have a complete picture of the performance of the devices.

In a multisystem environment there are several things you can do to process the data efficiently and get accurate reports. See "Information about the MVS SCP" in Part 4 for suggested procedures.

Running EREP under VM

The following series of examples do the same things as the procedures for VSE and MVS earlier in this chapter. ***These are only examples.*** You must decide which reports are relevent to your installation, what order they should be generated in and how often they should be run.

The examples are presented as pieces of a single EXEC that includes multiple executions of the CPEREP command using different sets of operands each time. The operands have been put in files created using XEDIT; the correct file is named on each CPEREP command. The actual contents of the files are included in the examples.

There are other methods for entering the operands for the CPEREP command; see "Entering CPEREP Operands" on page 9-8 and "Using EREP Controls as CPEREP Operands" on page 9-3.

The first section of the EXEC defines the input and output files needed for EREP. EREP automatically sends its output to the devices named on the EREPPT and TOURIST file definition statements. The execution of the CPEREP command includes the writing and routing of the output.

All of the EREP parameters are described in detail in Chapter 7, "EREP Parameters." The EREP control statements in Chapter 8, "EREP Control Statements."

The History Dataset

You should always create a history data set before running your EREP reports. This insures consistency across the reports.

The operating system adds records to the ERDS continually. If you run your reports against the ERDS (HIST=N) then the set of records used by the first report is not the same as the set used by the next report since records could have been added while the report was running. By the time the last report is run the set of records may be quite different.

By creating a history data set and then running all of the reports against that data set you insure that all of the reports are using the same set of records.

Create a History Dataset

Create a history data set to be used in later report generation:

- Generate a System Summary report from the records in the error-recording area
- Copy the records from the error-recording area to a working data set
- Clear the error-recording area

```
FILEDEF DIRECTWK DISK DIRECTWK EREPWORK *
```

Workspace for EREP; required when there is history input. This file is for later steps. In this first step, the input is in the file defined by CPEREP as SERLOG; see "Defining Files for CPEREP" on page 4-49.

```
DET 182
```

CPEREP expects the ACCDEV file to be on TAPE 182, already defined by the system. If you want to keep the records on disk, you must also detach (or redefine) TAPE 182. See "VM Notes" on page 4-51.

```
FILEDEF ACCDEV DISK EHISTORY DAILY *
        (RECFM VB BLKSIZE 12000
```

Output history file (ACC=Y).

```
FILEDEF EREPPT PRINTER
        (NOCHANGE BLKSIZE 133
```

Sending the System Summary report to the printer. Note that EREP requires a 132-position printer.

```
FILEDEF TOURIST TERMINAL
        (BLKSIZE 133
```

EREP informational messages will appear on the screen, instead of being printed with the report.

```
EXEC CPEREP STEP1 INPUT A
```

The contents of file STEP1 INPUT A are the CPEREP operands listed here.

SYSUM	Requesting the System Summary.
ACC=Y	Directing EREP to copy the records from the ERDS to EHISTORY DAILY.
ZERO=Y	Directing EREP to clear the ERDS.
ENDPARM	End of EREP parameters; EREP control statements follow.
SHARE= . . .	For any devices shared by processors.

System Exception Reports

Generate the series of System Exception reports for processors, channels, and DASD and tape subsystems:

- A System Error Summary in two parts:
  1. IPL/restarts, machine and channel checks
  2. DASD and tape I/O errors across processors.
- A Subsystem Exception report for each processor
- A Subsystem Exception report on channel failures, one for each processor
- A Subsystem Exception report on DASD failures
- A Subsystem Exception report on tape failures
- A series of detailed summaries for DASD presenting data by symptom code, storage control unit, string, and volume
- A series of detailed summaries for 34XX tape devices presenting permanent and temporary errors by CUA/device number and volume.

You must now define as input to this report the file containing the records copied from the ERDS in the previous step. This is the working copy of the ERDS; it is used as input for all of the remaining reports.

`DEF STOR 2M`	Need more virtual storage for a large sort table.
`FILEDEF ACCIN DISK EHISTORY DAILY *`	Redefining EHISTORY DAILY as ACCIN; the records in this file are input for the rest of the reports.
`EXEC CPEREP STEP2 INPUT A`	The contents of file STEP2 INPUT A are the CPEREP operands listed here.
SYSEXN	Requesting the System Exception series.
HIST	Input records are in the history data set.
ACC=N	Leave them there.
TABSIZE=512K	System Exception processing requires a large sort table.
ENDPARM	End of parameters; control statements follow.
DASDID . . .	For DASD not providing their own physical IDs.
LIMIT . . .	To limit the number of records appearing on the reports.
SHARE . . .	For devices shared between processors.

MCH and CCH Detail Reports

Print Detail Edits and Summaries of all machine and channel checks.

`DEF STOR 1M`	Returning virtual storage to its orginal size.
`EXEC CPEREP STEP3 INPUT A`	The contents of file STEP3 INPUT A are the CPEREP operands listed here.
PRINT=PS	Requesting Detail Edits and Summaries of the input records.
TYPE=MC	Selecting the records by type: M MCH C CCH
HIST	Input records are in the history data set.
ACC=N	Leave them there.
ENDPARM	End of parameters; control statements follow.
SHARE . . .	For devices shared between processors.

Threshold Summary

A summary of tape I/O errors:

- all permanent errors
- temporary errors that exceed 1 read or 15 writes.

Includes records from 3410, 3420, and 8809 devices.

`EXEC CPEREP STEP4 INPUT A`	**The contents of file STEP4 INPUT A are the CPEREP operands listed here.**
THRESHOLD=(001,015)	**Requesting a Threshold Summary.**
HIST	**Input records are in the history data set.**
ACC=N	**Leave them there.**
ENDPARM	**End of parameters; control statements follow.**
SHARE . . .	**For devices shared between processors.**

Detail Reports for Communications Devices

Print Detail Summaries of all errors for 3704, 3705 and 3725 communications controllers.

`EXEC CPEREP STEP5 INPUT A`	The contents of file STEP5 INPUT A are the CPEREP operands listed here.
PRINT=SU	Requesting only Detail Summaries.
TYPE=OT	Selecting the records by type: O OBR T MDR
DEV=(3704,3705,3725)	Selecting by device type; the communications controllers.
HIST	Input records are in the history data set.
ACC=N	Leave them there.
ENDPARM	End of parameters; control statements follow.
SHARE . . .	For devices shared between processors.

Detail Summaries for I/O Errors

Print Detail Summaries of all I/O errors not already covered in the preceding reports.

`EXEC CPEREP STEP6 INPUT A`	The contents of file STEP6 INPUT A are the CPEREP operands listed here.
PRINT=SU	Requesting only Detail Summaries.
TYPE=DOTH	Selecting the records by type: D DDR O OBR T MDR H MIH
DEV=(N34XX,N3704,N3705,N3725)	Selecting by device type; excluding those already covered. Note the absence of N33XX; EREP does not produce detail summaries for 33XX DASD anyway, so they need not be excluded.
HIST	Input records are in the history data set.
ACC=N	Leave them there.
ENDPARM	End of parameters; control statements follow.
SHARE ...	For devices shared between processors.

Detail Reports for Software Records

Print Detail Edits and Summaries of all software and operational records.

```
EXEC CPEREP STEP7 INPUT A
```

The contents of file STEP7 INPUT A are the CPEREP operands listed here.

PRINT=PS	Requesting both Detail Edits and Summaries.
TYPE=SIE	Selecting the records by type: S SIE I IPL E EOD (Termination)
HIST	Input records are in the history data set.
ACC=N	Leave them there.

Event History Report

One-line abstracts of all records, in chronological order.

`EXEC CPEREP STEP8 INPUT A`	The contents of file STEP8 INPUT A are the CPEREP operands listed here.
EVENT	Requesting an Event History Report. Note the absence of selection parameters; the report is to include every record.
HIST	Input records are in the history data set.
ACC=N	Leave them there.

Update the History Tape

1. Copy the records from the input history data set to the permanent history tape.
2. Delete the input history data set.

```
FILEDEF  ACCDEV TAPE HISTTAPE WKLY
           (RECFM VB BLKSIZE 133
```

This is the permanent history tape, on which the working data set is to be copied.

```
EXEC CPEREP STEP9 INPUT A
```

The contents of file STEP9 INPUT A are the CPEREP operands listed here.

PRINT=NO	Requesting no report output.
HIST	Input records are in the history data set.
ACC=Y	Move the records to another data set.

Trends Report

Print a Trends report covering the last 30 days of records from the newly updated history tape.

```
FILEDEF  ACCIN  TAPE
         HISTTAPE WKLY *
```

Redefining the former ACCDEV data set as ACCIN; the updated history tape is now being used as input.

```
EXEC CPEREP STEP10 INPUT A
```

The contents of file STEP10 INPUT A are the CPEREP operands listed here.

TRENDS	Requesting the Trends report. Without the DATE parameter, Trends uses the last 30 days of records.
HIST	Input records are still in a history data set rather than the ERDS.
ACC=N	Leave them there.
ENDPARM	End of parameters; control statements follow.
SHARE . . .	For devices shared between processors.

VM System Controls

The interface between EREP and the VM facility is the CMS command CPEREP. From the CMS environment, you enter the CPEREP command to begin an execution of the EREP program. You enter EREP parameters and control statements as operands for the CPEREP command, using one of several methods of entry.

The operands are in the same format as the EREP parameters and control statements in VSE or MVS jobs; in essence, you are submitting them as SYSIN data to the virtual machine's system control program.

See Chapter 9, "CPEREP Operands - Syntax and Coding" for detailed information on the operands and how you use them.

Defining Files for CPEREP

The CPEREP command processor invokes IFCEREP1, specifying the EREP controls you have entered as command operands. The command processor also defines the files necessary for running EREP. You can change the FILEDEFs before executing the CPEREP command and override the definitions the command processor would use.

Following are the "default" file definitions set up by CPEREP. CPEREP uses the MVS version of EREP, so the FILEDEF commands it issues correspond to the DD statements used by MVS:

```
FILEDEF EREPPT PRINTER (NOCHANGE BLKSIZE 133
FILEDEF SYSIN DISK SYSIN EREPWORK X3
FILEDEF SERLOG DISK SERLOG EREPWORK (BLOCK 4096
FILEDEF TOURIST TERMINAL (BLKSIZE 133
FILEDEF DIRECTWK DISK DIRECTWK EREPWORK X4
FILEDEF ACCDEV TAP1 (NOCHANGE RECFM VB BLKSIZE 12000
FILEDEF ACCIN TAP2 (NOCHANGE RECFM VB BLKSIZE 12000
```

EREPPT

is EREP's printer file, to which it sends the report output. You can override this FILEDEF with one of your own before issuing the CPEREP command; to change the destination from PRINTER to TERMINAL, for example.

CPEREP leaves the FILEDEF for EREPPT intact at the end of the run, in case you supplied it.

SYSIN

is the file where CPEREP puts your parameters and control statements for EREP. It is put on the read/write disk having the most available space, and is automatically erased at the end of the run. If there is no data for SYSIN, CPEREP issues FILEDEF SYSIN DUMMY. When you are entering operands by the file entry method, make sure the name of the file on this FILEDEF statement is the same as the file you name on the CPEREP command line.

SERLOG
is a simulation of the SYS1.LOGREC data set, required by the OPEN and CLOSE macros that EREP issues during its processing. When SERLOG is the input file, the records are actually read from the VM error-recording area; no SERLOG file exists on any disk. See "VM Notes" on page 4-51 for some interesting insights into VM's error recording.

TOURIST
is the message data, which is directed to your terminal screen. The messages and diagnostic information EREP writes to this file include printer control characters, which might appear on the display screen as unknown characters.

DIRECTWK
is a work file EREP uses when there is history input. This file can be quite large, because it contains all the input error records selected from the history tape. DIRECTWK is put on the read/write disk having the most available space, and is erased at the end of the run.

ACCDEV
is the output history file, used if you specify or imply ACC=Y. CPEREP puts this file on tape drive 181, but you can override that definition with your own FILEDEF prior to issuing the CPEREP command. However, defining ACCDEV to any device other than tape 181 can cause problems when CPEREP positions the tape so EREP can write records to the file; see "VM Notes" for more details.

CPEREP leaves the FILEDEF for ACCDEV intact at the end of the run, in case you supplied it.

ACCIN
is the input history file, used if you specify or imply HIST=Y or MERGE=Y. CPEREP puts this file on tape drive 182, but you can override that definition with your own FILEDEF prior to issuing the CPEREP command. However, defining ACCIN to any device other than tape 182 can cause problems when CPEREP positions the tape so EREP can read records from the file; see "VM Notes" for details.

CPEREP leaves the FILEDEF for ACCIN intact at the end of the run, in case you supplied it.

VM Notes

Running EREP under VM is quite different from running it under either of the other S/370 operating systems. Your access to the system is through the CPEREP command; CMS actually runs EREP for you, simulating the OS/VS2 environment.

VM simulates the OS/VS2 operating system when you invoke EREP, even when your input records are from an OS/VS1 or a VSE system.

Using Different Input and Output FILEDEFs for CPEREP

Before you issue the CPEREP command, you can define your own ACCDEV and ACCIN files using the FILEDEF command, and designate the devices you want to hold the files. Normally, you would only need to do this if you were using a history tape from another system as input to EREP, or if you wanted to accumulate the data to another tape drive or to a disk file.

Defining your own file for ACCDEV or ACCIN could lead to problems in reading or writing the records from or to the file, however. If you specify or imply ACC=Y for an EREP run, CPEREP rewinds tape 181, spaces forward over the existing file, and backspaces over the tape mark before writing any records to the file. It does this so it writes new records at the end of the accumulation file. Similarly, if you specify HIST=Y or MERGE=Y, CPEREP rewinds tape 182 so it always starts reading records at the beginning of the file.

For some VM systems, there is a catch to this. The catch is that, as long as tape drives 181 and 182 are attached to the virtual machine, CPEREP only, and always, positions those tapes.

In older versions of VM, CPEREP assumes, when it sees ACC=Y, HIST=Y, or MERGE=Y, that the files are on tape drives 181 and 182. If tape 181 and/or 182 is attached and ready, CPEREP positions it for writing or reading; if the tape is attached but not ready, CPEREP notifies the operator and waits for him to make it ready. It does not use the virtual device you defined in your FILEDEFS.

To avoid this problem, you must do two things in each case:

1. For ACCDEV, detach tape 181 before running CPEREP, and issue CMS TAPE commands with the appropriate "tapcmd" controls to position the tape you have defined before invoking CPEREP.[1]

   If you define the file to a disk, specify the DISP MOD option on the FILEDEF if you want to add records at the end of an existing file.

2. For ACCIN, detach tape 182 and issue a CMS TAPE command[1] to rewind the ACCIN tape you have defined, before running CPEREP.

   If you define the file to disk, you need not do anything.

---

[1] See the *CMS Command and Macro Reference* for your VM facility for information on the TAPE command and its options.

FILEDEFs for Standard-Label Tapes

If you are using standard-label tapes, you must issue your own FILEDEF for ACCDEV or ACCIN before running CPEREP, so the header labels are read correctly. This is the correct FILEDEF for ACCDEV.

```
FILEDEF ACCDEV address SL (RECFM VB BLKSIZE 12000
```

This is the correct FILEDEF for ACCIN.

```
FILEDEF ACCIN address SL (RECFM VB BLKSIZE 12000
```

IMPORTANT: The address for ACCDEV *and the address for* ACCIN ***must*** *be different.*

Running the EREP General Input Facility

Description

The EREP General Input Facility passes records to EREP for Detail Edit processing. Any program operating in an MVS environment can use this facility to pass records, one at a time, to EREP for processing.

The facility is invoked for two functions:

1. Process a record and format the Detail Edit report.
2. Perform end-of-file processing and provide a total count of records processed.

Requirements

EREP does not directly read records or print reports through the General Input Facility interface. The caller is responsible for all I/O services. The caller supplies EREP with:

- The address of a record already in storage
- The address of an output routine that will print the report output and/or the end-of-file information

Invocation

EREP is invoked through the General Input Facility by LOADing and calling module IFCRCGIF.

Input

Input to the General Input Facility consists of the following parameter list, the address of which is passed in Register 1.

FORMAT	FIELD DESCRIPTION
PTR(31)	Address of ERDS record
FIXED(31)	Length of ERDS record
PTR(31)	EPA of caller's print routine
PTR(31)	Address of data area to be passed to output routine (optional)

Note: *When the facility is invoked for end-of-file processing, the first two fields are set to zero.*

Output

The output of the General Input Facility consists of the Detail Edit report, to be printed one line at a time, and a parameter list. Both of these are passed to the caller's output routine, where the line is printed. Upon return from the output routine, EREP will return control to the caller.

Output Parameter List

FORMAT	FIELD DESCRIPTION
PTR(31)	Address of output line
PTR(31)	Address of data area from input parameter list (optional)

Output Line

FORMAT	FIELD DESCRIPTION
CHAR(1)	Output dataset indicator R report data set M message data set B both data sets
CHAR(1)	Carriage Control character X'F1' page eject X'1B' immediate triple space X'19' triple space after write X'13' immediate double space X'11' double space after write X'xx' All others cause single space
CHAR(132)	Output line

Restrictions

The record passed to EREP through the General Input Facility must reside below the 16-megabyte line.

Return Codes

The following return codes may be passed back to the caller:

CODE	DESCRIPTION
X'00'	Successful processing
X'04'	Warning, processing unsuccessful. The requestor should perform default processing for this record.
X'20'	Severe system-related errors. EREP cannot continue. The requestor should perform default processing for this and all subsequent records.

Chapter 5. Correcting Coding Problems

There are four major causes of problems in running an EREP job:

1. Incorrect syntax or use of an EREP parameter or control statement

2. Inadequate storage for the data sets and work spaces EREP requires

3. Some kind of I/O error — opening or closing data sets, reading data, writing data

4. Flawed input data, usually caused by improper management of the error data, that results in incorrect report output.

For the first three of these, you get messages from EREP or the operating system, or both, to help you identify and correct the problem. In the case of incorrect input, however, EREP usually issues no messages, so you have to identify the problem yourself.

The following paragraphs summarize the recommended actions for each of the problem causes listed here.

Syntax Errors

When your problem is a syntax error or an improper combination of parameters, the TOURIST message is quite specific about what is wrong. The reference section of this book can help you correct the problem quickly. Besides Part 3, "Reference Information," you will find device-specific information in Part 4, "Product-Dependent Information" that relates to EREP controls.

Storage Problems

You must provide enough storage for EREP's internal sort table (using the TABSIZE control parameter), or the program terminates. In addition, you must make sure the user region or virtual storage size for the job or step is big enough to hold the sort tables and the EREP program itself. If it is not, the EREP run terminates.

Messages in the MSGCLASS (MVS only) and TOURIST output indicate if one of these problems occurred; the MSGCLASS output includes the completion codes for the steps in the job.

You can check the storage requirements for the system you are running EREP under in "VSE Storage Requirements" or "MVS Storage Requirements" and make corrections before trying to rerun EREP. (The VM virtual storage requirements and capacities depend on the guest system and the virtual machine.)

I/O Errors

When EREP cannot open or close a data set, or read the input records, or write the output records to the EREPPT data set, you have a data management problem. EREP puts messages in the TOURIST output to alert you, and the MSGCLASS output from the job helps as well, by showing where the job ended and what the completion codes were. Check "VSE System Controls," or "MVS System Controls," or "Defining Files for CPEREP" for the proper data management information.

Incorrect Input

When your EREP report output is not right, the chances are good that it is because the input records were not right. In this case, you must verify the records EREP used for the report. In the absence of helpful messages, the best way to do this is via the DEBUG option; see "The EREP DEBUG Parameter" on page 11-63.

EREP input can be incorrect for one of two reasons:

1. The system recording routines recorded the records incorrectly
2. Something happened to the records on their way from the ERDS to another data set.

The second reason is the more likely; it is also the only one you can do anything about directly. Losing or duplicating records is not primarily a coding problem, however; it is really a matter of managing your error data base for EREP. See "Managing Error Data" on page 3-1 for some suggestions on preventing the destruction or duplication of records.

A good general-purpose way to check for missing records is to run an Event History report specifying DEV and TYPE to match the suspect record; the report will include data from every record that meets your selection criteria. Another way to look for a particular record is to run a Detail Edit of the record (specifying DEV, TYPE, DATE and TIME, and any other selection parameter that will narrow the choice).[1] The Detail Edit includes formatted data from the record, as well as a hexadecimal dump of the record itself.

---

1 Not necessarily the easiest way; see "Using the DEBUG Parameter" on page 5-3.

Using the TOURIST Output

Should your EREP job not run because of a syntax or other coding error, you can use the TOURIST output not only to see the message about your problem, but also to see exactly how EREP interpreted your control statements and parameters. The system control statements (JCL, for example) and the actual EREP controls, as you entered them, appear in the MSGCLASS output for the job.

Figure 12-9 on page 12-12 is an example of the typical TOURIST output generated for a single EREP report.

EREP normally prints the TOURIST messages just before the requested report, so you can look at both together in case there is a problem. However, you can control the output class and device for the TOURIST output. For example, in an MVS installation, you can change the output classes for the EREPPT and TOURIST data sets. When things are running smoothly, you can spool the TOURIST output to another device so you see only the reports from the EREPPT device. If you do this, remember to check the TOURIST messages from time to time to make sure you haven't missed anything. When you are debugging a problem with a report or with your EREP run, and you want to see the messages with the report, simply change the output class for the TOURIST data set to match that of EREPPT.

Using the DEBUG Parameter

If a problem with an EREP report or your EREP run is associated with an input record, you must be able to look at the record. You can run a Detail Edit of the record; but this can be complicated as you try to isolate a particular OBR record, for example, among many from the same device types.

The easiest way to see the records used for an EREP report is to run an Event History and include the DEBUG parameter with its option 17 in the EREP controls. The records will appear, unformatted and in hexadecimal, in the report. See "The EREP DEBUG Parameter" on page 11-63 for coding details.

When you have deciphered the contents of a record, you can compare it to the mapping of the record in Chapter 10, "Error Records for EREP," to see if they match. The IBM Customer Engineer can also help you interpret the records, referring to the maintenance documentation for the device that generated the record.

Part 3. Reference Information

How to Use Part 3

This part of the *EREP User's Guide and Reference* presents information in quick-look-up format: the syntax of EREP parameters and control statements and of CPEREP operands; tables and charts showing how the EREP controls go together and how the requirements of the operating systems differ.

The reference information also includes mappings of the formats of the records EREP edits and prints, along with a brief explanation of why each record is generated; and all the EREP messages, as they appear for each operating system.

Use the Section Table of Contents to find specific kinds of information in Part 3; more details on the various topics are in the other parts of this book.

Section Table of Contents

Chapter 6. Introduction to EREP Controls

You communicate to EREP using keyword parameters and control statements.

The parameters tell EREP which report to run, if any, and which records to use for the report, and what to do with the records when the report is complete.

Control statements tell EREP what your hardware configuration is like: how many processors you have; and whether or not your I/O devices are shared by more than one processor; and exactly where the devices are. Control statements also give EREP other information, such as limits on the number of errors to be included in a report.

Figure 7-2 shows the syntax for every EREP parameter; Figure 8-7 shows the correct format for each EREP control statement. The full descriptions of parameters and control statements are in Chapter 7, "EREP Parameters" and Chapter 8, "EREP Control Statements."

Syntax Rules and Conventions

The following paragraphs describe the notation we have used to define the syntax and format of the EREP control statements and parameters. Syntax rules define what is required or optional for the specific purpose or process you are requesting. Certain rules are common to all parameters:

- Code **uppercase letters, numbers, and the set of symbols listed here** EXACTLY as shown in the parameter/statement syntax.

apostrophe	'
asterisk	*
comma	,
equal sign	=
hyphen	-
parentheses	( )
period	.

- **Lowercase letters, and other symbols,** appearing in a parameter/statement syntax represent **variables** for which you must substitute specific information. For example, if the word "serial" appears in the parameter/statement syntax, substitute a specific serial number value (012345, or 503B) for it in the actual parameter/statement.

- We use the set of symbols listed below to **define the syntax** for you. You never use them in the actual parameter.

underscore	—
braces	{ }
brackets	[ ]
ellipses	. . .
vertical bar	|

  Here is how we use these symbols:

  – An **underscore** indicates a default option. If you want an underscored alternative, you need not specify it in the actual parameter.

  – A **vertical bar** represents logical OR, and means that you can code one or the other of two alternatives.

    For example: `KEYWORD=[ALPHA|BETA]`

    indicates that you can code either ALPHA or BETA as the value for KEYWORD.

  – **Braces** group related items, such as alternatives.

    For example: `ALPHA=({A|B|C},D)`

    indicates that you **must** choose one of the items enclosed within the braces. If you choose A, code ALPHA=(A,D).

  – **Brackets** also group related items; however, everything within the brackets is optional and may be omitted.

    For example: `ALPHA=([A|B|C],D)`

    indicates that you **may** choose one of the items within the brackets or omit all of them. If you select only D, code ALPHA=(,D).

  – An **ellipsis** indicates that the preceding item or group of items can be repeated more than once in succession.

    For example: `ALPHA [,BETA]  . . .`

    indicates that ALPHA can appear alone or can be followed by ,BETA any number of times in succession.

- A **hyphen** or **dash** between two entries indicates a range. Code the hyphen exactly where shown in the syntax.

  For example:

  ```
  hhmm-hhmm  indicates a range of time.
  addr-addr  indicates a range of continuous addresses.
  ```

- **Alphameric characters**. Unless otherwise indicated, an alphameric character is one of the following:

alphabetic	A-Z
numeric	0-9
national	$ # @

Parameter and Control Statement Layout

This section includes descriptions of all the EREP parameters and control statements as well as their correct syntax. The descriptions all look like this:

Name of Parameter	The shortened form; which kind of parameter it is; the full name.
Tells EREP to:	What it does; any related device information.
Syntax:	The format; the way it should be coded.
Default:	Description, if one exists; otherwise, "None."
Parameter Conflicts:	Other parameters that you cannot use with this parameter.
Coding:	Any special coding rules that you must follow.
Note(s):	Brief special information about the parameter/control statement.
Example(s):	The completed parameter or control statement if not obvious from the syntax; otherwise, blank.

In this part of the book, the EREP parameters and control statements are arranged alphabetically within the two groups. You will find device-specific usage notes for the parameters and some control statements in Part 4, "Product-Dependent Information."

Chapter 7. EREP Parameters

You direct EREP processing, and tailor EREP reports, using the keyword parameters listed on the following pages.

Note that none of the parameters is required; you may allow EREP to operate entirely by default. However, you should check the default options to be sure they are the ones you want.

This reference section contains the syntax and default value for each EREP parameter, along with a brief description of what the parameter does, and how it works with other EREP controls. You will find more notes on usage in Part 2, "Setting Up and Running EREP" and Part 4, "Product-Dependent Information."

Figure 7-2 on page 7-41 is a summary of the correct syntax for all the EREP parameters.

Coding EREP Parameters

The same coding rules apply to all the EREP parameters. To avoid repetition, they are listed here rather than with each parameter.

- Each parameter consists of a keyword followed by an equal sign and one or more values:

  `KEYWORD=(`*value,value . . . ,value*`)`

- Some parameters require parentheses around the value field.

- For all of the report parameters except PRINT, and for some of the control parameters, you need code only the keyword; YES is implied. For example:

  `SYSUM` is the same as `SYSUM=Y`

  `EVENT` is the same as `EVENT=Y`

  `HIST` is the same as `HIST=Y`

- There can be no imbedded blanks in the parameter expression or the parameter field.

- Use commas to separate the parameters if they are on the same line.[1] If you code the parameters as instream data, they can be entered as one record or as individual records:

```
//SYSIN   DD   *
TRENDS,HIST,ACC=N,DATE=(82032,82056)
ENDPARM
.
.
.
control statements
.
.
.
/*
```

OR

```
//SYSIN   DD   *
TRENDS
HIST
ACC=N
DATE=(82032,82056)
ENDPARM
.
.
.
control statements
.
.
.
/*
```

- If you enter the parameters as instream data and you are also entering EREP control statements, the parameters must precede the control statements and be followed by ENDPARM.

1 When entering parameters as CPEREP operands, you can separate them by commas or one or more blanks.

- When you run EREP under a VSE system, you must enter the parameters as instream SYSIPT data following the EXEC statement:

```
// EXEC  IFCEREP1
EVENT,DATE=(82330-83015),HIST,ACC=N
ENDPARM
/*
```

OR

```
// EXEC  IFCEREP1
EVENT
DATE=(82330-83015)
HIST
ACC=N
ENDPARM
/*
```

Note: ENDPARM is the delimiter EREP looks for between parameters and control statements, when the parameters are entered as instream data.

- When you run EREP under an MVS system, you code the parameters either as instream (SYSIN) data with the EREP control statements, or on the JCL EXEC statement, as

  PARM='*keyword=value,keyword=value*'.

  - If you put the parameters on the EXEC statement, and they include special characters (for example, =), you must enclose each parameter expression in single apostrophes or parentheses.
  - If the parameters are continued to another line, you must enclose the entire field in parentheses. For example:

```
//STEP1   EXEC   PGM=IFCEREP1,PARM=('PRINT=PS','TYPE=IE',
//               'ACC=N',HIST)
```

  See the *JCL* manual for your MVS system for more details on coding the PARM parameter.

Using Comments

You provide documentation of your EREP jobs in comments, identified by an asterisk (*) in the first column of the input record.

Your comments appear in the TOURIST output. See "DASDID Control Statement" on page 8-7 for examples of using comments to document your EREP controls.

Selection Parameters for EREP Reports

You can use both selection and processing parameters for EREP reports. With very few exceptions, all the processing parameters can be combined with any of the report parameters. The selection parameters, however, are less straightforward. Even though most of them are *valid* with all the report parameters, many are not meaningful for a given report.[2]

Figure 7-1 shows which selection parameters you can and cannot use for the various EREP reports. The descriptions in the rest of this chapter give more detailed information about the relevance of each selection parameter to the EREP reports.

REPORT PARAMETERS	SELECTION PARAMETERS														
	CPU	CPUCUA	CUA	DATE	DEV	DEVSER	ERRORID	LIA/LIBADR	MOD	MODE	SYMCDE	TERMN	TIME	TYPE	VOLID
EVENT=Y						NO									
PRINT=AL						NO									
=DR						NO									
=PS						NO									
=PT						NO									
=SD						NO									
=SU						NO									
PRINT=NO						NO									
SYSEXN=Y						NO									
SYSUM=Y						NO									
THRESHOLD	NO	NO			1		NO	NO	NO		NO	NO		NO	
TRENDS=Y						NO									

Figure 7-1. Selection Parameters for EREP Reports

Notes:

1. *The following devices are allowed: 3410, 3420, 8809, 34XX.*

See Figure 12-1 on page 12-2 for incorrect combinations of selection and processing parameters.

---

2 This is true for EREP 3.3 and later releases; if you are running EREP 3.2 or an earlier release, more selection parameters are actually invalid for use with some report parameters. See Chapter 12, "Summary of Tables and Charts."

Parameter Descriptions

ACC

***Accumulate Records** (Processing Parameter)*

Tells EREP to: Copy the records that passed filtering for the report onto an output data set.

Syntax: ACC[=Y]|=N

Default: If you omit this processing parameter, EREP assumes ACC=Y, except when you request a Threshold report. Then, the default is ACC=N.

Coding: Specifying ACC is the same as ACC=Y.

Parameter Conflicts: DEVSER
THRESHOLD

Note(s):

- When you use or imply ACC=Y for an EREP run, you must also code the system control statement(s) needed to define the output data set that will hold the records. This is described in the "System Controls" sections of Part 2, "Setting Up and Running EREP."

- EREP does not zero the ERDS unless *all* the records have been accumulated on an output data set. See the ZERO parameter for more details.

- If you code ZERO=Y when requesting PRINT=SU or PRINT=NO, EREP assumes ACC=Y and expects you to define the output data set.

- If you run a System Summary report with LOGREC as input and allow ACC to default, EREP also assumes ZERO=Y. This is not a problem unless you have coded ACCDEV as a dummy data set (MVS only). In that case, the records are lost.

CPU

Central Processing Unit (Selection Parameter)

Tells EREP to: Use only the records containing the model and serial numbers you specify on this parameter for the requested report. The model number is the machine type.

The valid processor model numbers for the CPU parameter are:

0115	0125	0135	0138	0145	0148	0155	0158
0165	0168	3031	3032	3033	3052	3062	3081
3083	3084	3090	4321	4331	4341	4361	4381
9081	9083	9373	9375	9377			

Syntax: **CPU = (***serial.model*[,*serial.model*] . . . **)**

serial is the six-digit decimal processor serial number.

model is the four-digit decimal processor model number.

Maximum of six entries.

Default: EREP processes records from all processors.

Coding: No special considerations.

Parameter Conflicts:
CPUCUA
MOD
THRESHOLD
ZERO

Note(s): If you use the CPU parameter, you cannot use ZERO = Y because you have excluded some records from processing.

Example(s): `CPU=(123456.0168,234567.3081)`

CPUCUA

CPU/Channel/Unit Address (Selection Parameter)

Tells EREP to: Select for the requested report only the records from the specified unit addresses associated with the specified processors.

Syntax: **CPUCUA=(*serial*.{*cua*|*cu*X}[,*serial*.{*cua*|*cu*X}] . . .)**

serial is the six-digit decimal CPU (processor) serial number.

cua is a unique three- or four-digit hexadecimal channel/unit address (in a 370/XA environment, it is the device number).

cuX is two or three hexadecimal digits followed by an X to denote the range of device addresses with those digits ending in 0 through F.

Maximum of four entries.

Default: EREP processes all available records.

Coding: No special considerations.

Parameter Conflicts:
CPU
CUA
DEVSER
MOD
THRESHOLD
ZERO

Note(s):

- If you use the CPUCUA parameter, you cannot use ZERO=Y because you have excluded some records from processing.
- CPUCUA only affects the selection of record types (TYPE Parameter) that contain a CUA: CCH, DDR, MIH, OBR, and MDR.

CUA

Channel/Unit Address (Selection Parameter)

Tells EREP to: Select for the requested report only the records related to the specified channel and or/unit addresses.

Syntax: **CUA = ({*addr*|*addr-addr*}[,*addr*|*addr-addr*] . . .)**

addr is a three- or four-digit hexadecimal address or group of addresses. The format of the address may be nnXX, nnnX, or nnnn (for example, 01XX, 038X, or 049C). nnXX means that EREP processes all controller/unit addresses on channel 'nn'; nnnX means that EREP processes all unit addresses on channel/control unit 'nnn'.

Note: The channel identifier can be one or two digits. Reports produced by EREP 3.2 show four-digit device addresses.

addr-addr is a range of contiguous hexadecimal addresses, which may include more than one channel and control unit. The lower address must appear first in the expression.

Maximum of eight entries.

Default: EREP processes records from all devices (CUAs).

Coding: No special considerations.

Parameter Conflicts: CPUCUA
ZERO

Note(s):

- If you use the CUA parameter, you cannot use ZERO=Y because you have excluded some records from processing.
- CUA only affects the selection of record types (TYPE parameter) that contain a CUA: CCH, DDR, MIH, OBR, and MDR.
- If there are alternate paths to a device, and you want EREP to process all the records for the device, you must specify the CUAs for all the alternate paths.

Example(s):

```
CUA=(012C)

CUA=(0123,032X,04XX)

CUA=(123-320,4B0-C00)
```

DATE

Date Range (Selection Parameter)

Tells EREP to: Select records created during the specified date range.

Syntax: **DATE = (***yyddd*[{,-}*yyddd*]**)**

yyddd is the year (yy) and the Julian day (ddd).

The first yyddd is the year and day when the date range begins; the second yyddd is the ending year and day. The second date is optional; you can select records from a single date as well as from a range of dates. To select a single date, code only one yyddd.

When you code a date range, the second yyddd must be equal to or greater than the first. If it is not, EREP issues a syntax-error message.

Default: None.

Except for the Trends report. For the Trends report, the default if you do not code the DATE parameter is to process 30 days of error data.

Parameter Conflicts: ZERO

Note(s):

- DATE is valid with all the report parameters.
- If you use the DATE parameter, you cannot use ZERO=Y because you have excluded some records from processing.
- To express a range of 30 days, add 29 to the beginning Julian day.
- DATE is required when you use the TIME selection parameter.

Example(s):

```
DATE=(82137)

DATE=(82136,82143)

DATE=(83152-83182)
```

DEV

Device Type (Selection Parameter)

Tells EREP to: Select — or exclude — records associated with the specified generic device type(s).

The valid device types for DEV are:

1012	1015	1017	1018	1030	1050	1052
1053	1060	1130	115A	1255	1270	1275
1285	1287	1288	1403	1419	1442	1443
2020	2150	2250	2260	2265	2280	2282
2301	2303	2305	2311	2314	2321	2400
2495	2501	2520	2540	2560	2596	2671
2701	2702	2703	2715	2740	2741	2760
2770	2780	2790	2930	2947	2955	2956
2970	2972	3036	3066	3138	3148	3158
3168	3203	3210	3211	3213	3215	3262
3277	3278	3284	3286	3289	3310	3330
3340	3350	3370	3375	3380	3410	3420
3430	3480	3504	3505	3525	3540	3670
3700	3704	3705	3725	3735	3791	3800
3838	3848	3850	3851	3886	3890	3895
3945	3968	4245	4248	5080	5203	5424
5425	7340	7443	7770	7772	83B3	8809
9332	9335	9347	BA00	CTCA		

The valid general device classes for DEV are:

23XX	27XX	32XX	33XX	34XX	37XX	38XX

Syntax: **DEV=(***type*|**N***type*[**,***type*|**N***type*] . . . **)**

type is a four-digit hexadecimal number: either a specific device type (3340, 3420) or the representation of a class of devices (33XX, 34XX, and so on).

N indicates "not"; excludes a device type from the report.

Maximum of eight entries.

Default: EREP processes records associated with all device types.

Coding: The device type number(s) must be enclosed in parentheses.

Parameter Conflicts: ZERO

See Figure 12-1 on page 12-2 and Part 4, "Product-Dependent Information" for some special restrictions.

Note(s):

- If you use the DEV parameter, you cannot use ZERO=Y because you have excluded some records from processing.
- The only record types affected by the DEV parameter are the following: DDR (D), MIH (H), OBR (O) and MDR (T). Even though they contain a device code, CCH and SLH records cannot be selected by device type.
- If a device is emulating another device, use the device type number of the emulated device on the DEV parameter.
- You cannot select and exclude devices on the same DEV parameter; DEV=(3330,N2400) is invalid. You can, however, code more than one DEV parameter.
- EREP interprets some DEV entries to mean more than just the device you have coded; see Part 4, "Product-Dependent Information" for additional device-specific considerations.

Example(s):

To select records from specific devices or a class of devices:

```
DEV=(3420)
```

```
DEV=(33XX,3705)
```

To exclude the records from specific devices or a class of devices:

```
DEV=(N3420)
```

```
DEV=(N33XX,N3705)
```

DEVSER

Device Serial Number (Selection Parameter)

Tells EREP to: Select for the Threshold Summary only those OBR records that contain the specified device serial number(s). See the notes that follow for more information about the device serial number.

Syntax: **DEVSER** = (*serial*[,*serial*] . . .)

serial is a six-digit decimal device serial number from the service data. See "34XX Tape Devices" in Part 4, "Product-Dependent Information" for the devices to which DEVSER applies.

Maximum of eight entries.

Default: EREP selects OBR records without regard for the device serial numbers they contain.

Coding: No special considerations.

Parameter Conflicts:

ACC
CPUCUA
ERRORID
EVENT
LIA/LIBADR
MOD
PRINT
SHORT
SYMCDE
SYSEXN
SYSUM
TERMN
TRENDS
ZERO

See Figure 12-1 on page 12-2 and "34XX Tape Devices" on page 17-4 for some special restrictions.

Note(s):

- DEVSER is used only for the Threshold Summary report.
- DEVSER is valid with all the EREP processing control parameters except ACC and ZERO.
- EREP forces the DEV and TYPE parameters when you use the DEVSER parameter. See "34XX Tape Devices" in Part 4, "Product-Dependent Information."

- The device serial number is a value in a two-byte field of a tape OBR record that should correspond to the external serial number of the device. If the external serial number is greater than 65535, only the four low-order digits (decimal) are correct for the device serial. To use DEVSER to specify numbers larger than 65535, do the following:

  1. Convert the external serial number to binary
  2. Reconvert the low-order (rightmost) 16 bits to decimal
  3. Pad the resulting number with leading zeros to make a six-digit decimal number.

Example(s):

```
DEVSER=(013455,113455,213455)
```

ERRORID

Error Identifier (Selection Parameter)

Tells EREP to: Select for the requested report only the records containing the specified error identifier.

Syntax: **ERRORID=(***seqno***[,***cpuid,asid,hh,mm,ss,t***])**

seqno is a five-digit decimal error identifier from an MCH record or an MVS software (SFT) record.

cpuid is a two-digit hexadecimal processor (CPU) identifier.

asid is a four-digit hexadecimal address space identifier.

hh is a two-digit decimal value representing the hour.

mm is a two-digit decimal value representing the minute.

ss is a two-digit decimal value representing the second.

t is a single decimal digit indicating tenths of the second.

Default: EREP processes all MCH and SFT records, regardless of their error identifiers.

Coding: No special considerations.

Parameter Conflicts: DEVSER
THRESHOLD
ZERO

Note(s):

- If you use the ERRORID parameter, you cannot use ZERO=Y because you have excluded some records from processing.
- The only records that contain an ERRORID are MCH records and software (SFT) records produced by MVS. Therefore, the only record TYPE values you can code with ERRORID are M and S.
- Coding only the sequence number (seqno) causes EREP to process all records with the same ERRORID, regardless of when or where they were recorded.

- If you code the time-stamp values on the ERRORID parameter, you must also code the DATE parameter.

Example(s):

```
ERRORID=(01234)

ERRORID=(23456,01,0012,06,21,31,6)
```

EVENT

Event History (Report Parameter)

Tells EREP to: Produce an Event History report — one-line abstracts of selected records, in chronological order.

Syntax: EVENT[= Y]| = N

Default: Unless you specifically code EVENT or EVENT = Y, EREP does not produce an Event History report.

Coding: Specifying EVENT is the same as EVENT = Y.

Parameter Conflicts: DEVSER

Note(s): If you do not code any selection parameters with EVENT, EREP processes all available records for the report. However, the default value for the ZERO parameter is still NO; EREP does not clear the ERDS unless you specifically request it.

HIST

History Input (Processing Parameter)

Tells EREP to: Use the records in a history data set for the requested report, instead of those in the ERDS.

Syntax: **HIST[=Y]|=N**

Default: If you omit this processing parameter, EREP assumes HIST=N and uses the ERDS as input.

Coding: Specifying HIST is the same as HIST=Y.

Parameter Conflicts: MERGE
ZERO

Note(s):

- HIST is valid for all the report parameters.
- When you use the HIST parameter, you must also code the system control statements needed to define the input data set that holds the records, and a temporary work data set. See Figure 1-9 on page 1-15 and your "System Controls" section in Chapter 4, "Running EREP."
- If you do not use the HIST (or MERGE) parameter, you are telling EREP that the ERDS is its input.
- When running EREP under an MVS system, you can use more than one data set as the history input; just concatenate DD statements for the other data sets to the ACCIN DD statement. In VSE systems, the history input must be on a single data set.

LIA/LIBADR

Line Interface Base Address (Selection Parameter)

Tells EREP to: Select MDR records according to the specified line interface base address. See Chapter 22, "Teleprocessing (TP) Devices" in Part 4, "Product-Dependent Information."

Syntax: **{LIA|LIBADR}** = *address*

address is a four-digit hexadecimal line interface base address.

Default: EREP processes all available records.

Coding: No special considerations.

Parameter Conflicts:
DEVSER
SYMCDE
TERMN
THRESHOLD
VOLID
ZERO

See Figure 12-1 on page 12-2 and Part 4, "Product-Dependent Information" for some special restrictions.

Note(s):

- You can use LIA or LIBADR; EREP accepts both forms.
- If you use the LIA/LIBADR parameter, you cannot use ZERO=Y because you have excluded some records from processing.
- EREP assumes you want records from a 3705 or 3725 communications controller when you use the LIA/LIBADR parameter, so coding the DEV parameter with any other device number causes a parameter conflict. See Chapter 22, "Teleprocessing (TP) Devices" in Part 4, "Product-Dependent Information."

LINECT

Line Count (Processing Parameter)

Tells EREP to:	Print this many lines on each page of output.
Syntax:	LINECT = *nnn*
	nnn is one to three decimal digits.
	Minimum value is 25.
Default:	For VSE systems, the number of lines set for SYSLST at SYSGEN.
	For MVS and VM systems, 50 lines per page.
Coding:	No special considerations.
Parameter Conflicts:	PRINT = NO
Note(s):	If the value you specify for LINECT is less than 25, EREP ignores it and uses the default value instead.

MERGE

Merge Input Data Sets (Processing Parameter)

Tells EREP to: Use the records from both the ERDS and a history data set as input for the requested report.

Syntax: **MERGE[=Y]|=N**

Default: If you omit this processing parameter, EREP assumes MERGE=N and uses records from only one input data set.

Coding: Specifying MERGE is the same as MERGE=Y.

Parameter Conflicts: HIST

Note(s):

- When you use the MERGE parameter, you must also make sure the system control statements needed to define both of the input data sets are present or accounted for. See Figure 1-9 on page 1-15.
- If you do not use the MERGE (or HIST) parameter, you are telling EREP that the ERDS is its only input.
- Under MVS, the history input can be in more than one data set. See "HIST" on page 7-18.

MOD

Processor Model (Selection Parameter)

Tells EREP to: Select for the requested report only those records containing the specified CPU (processor) model number(s).

The valid processor model numbers for the MOD parameter are:

```
0115  0125  0135  0138  0145  0148  0155
0158  0165  0168  3031  3032  3033  3052
3062  3081  3083  3084  3090  4321  4331
4341  4361  4381  9081  9083  9373  9375
9377
```

Syntax: **MOD=(***model*[,*model*] . . . **)**

model is a three- or four-digit decimal processor model number.

Maximum of four entries.

Default: EREP processes records regardless of which kind of processor they were created on.

Coding: No special considerations.

Parameter Conflicts:
CPU
CPUCUA
DEVSER
THRESHOLD
ZERO

Note(s):

- MOD is the processor equivalent of the DEV parameter.
- If you use the MOD parameter, you cannot use ZERO=Y because you have excluded some records from processing.

Example(s): `MOD=(168,3031)`

MODE

Operating Mode (Selection Parameter)

Tells EREP to: Select for the requested report only those records created while the system was operating in the specified mode.

Syntax: **MODE** = {*370*|*370XA*|**ALL**}

370 means 370 mode only.

370XA means 370-XA mode only.

ALL means both 370 and 370-XA mode.

Default: If you omit this selection parameter, EREP assumes MODE=ALL and processes all available records, regardless of the mode they were recorded in.

Coding: No special considerations.

Parameter Conflicts: None.

See Figure 12-1 on page 12-2 and Part 4, "Product-Dependent Information" for some special restrictions.

Note(s):

- ZERO=Y is valid only with MODE=ALL.
- If EREP is running under any MVS system except MVS/XA, it treats software (SFT) records produced by MVS/XA as unknown records. Therefore, the combination of MODE=370XA or MODE=ALL and TYPE=S is meaningful only if the records were produced by MVS/XA.
- If you code MODE=370 and TYPE=C, EREP processes CCH records; if you code MODE=370XA and TYPE=C, EREP processes SLH and CRW records. Code MODE=ALL and TYPE=C to select all available CCH, SLH, and CRW records for the report.
- If a device is supported in 370-XA mode, any Detail Summary reports you request for the device will reflect that mode, regardless of what you specify on the MODE parameter.

PRINT

Detail Print Reports (Report Parameter)

Tells EREP to: Produce the Detail PRINT report(s) specified; or (PRINT = NO) to produce NO printed report output.

Syntax: **PRINT = {AL|DR|NO|PS|PT|SD|SU}**

AL requests all the Detail PRINT reports: Detail Edits of the records, Detail Summaries, and, if applicable, Data Reduction reports.

DR requests only Data Reduction reports.

NO requests that no reports be printed at all.

PS requests both Detail Edit and Detail Summary reports.

PT requests only Detail Edit reports.

SD requests Detail Summaries and Data Reduction reports.

SU requests only Detail Summary reports.

Default: If you do not code any report parameter at all, EREP assumes PRINT = SD, which produces a Detail Summary and, if applicable, a Data Reduction report for each record and device type you select. If you code PRINT without any keyword value, it is a syntax error.

Parameter Conflicts: DEVSER

Coding: You cannot code PRINT alone; it requires one of the report designations (AL, PT, SU, and so on) as well.

Note(s):

- PRINT = SD is the default report parameter for all EREP processing. The only way to avoid getting any printed report output is to code PRINT = NO.
- The default value for the ZERO parameter with PRINT is NO; EREP does not clear the ERDS unless you specifically request it. If you use selection parameters with PRINT, you cannot clear the ERDS, because not all the records have been processed for the report.

- If you code ZERO = Y and either PRINT = NO or PRINT = SU, EREP assumes ACC = Y as well; make sure the ACCDEV output data set is present to receive the accumulated records.
- See Part 4 for product-specific notes about the PRINT parameter.

SHORT

Print Short OBR Records (Processing Parameter)

Tells EREP to: Include short OBR records in a requested Detail Edit or Summary (PRINT) report.

Syntax: **SHORT[=Y]|=N**

Default: If you omit this processing parameter, EREP assumes SHORT=N and suppresses the detail printing of short OBR records.

Coding: Specifying SHORT is the same as SHORT=Y.

Parameter Conflicts: DEVSER
PRINT=DR
PRINT=SD

Note(s): The OBR Detail Summary always includes the information in short OBR records.

SYMCDE

Fault Symptom Code (Selection Parameter)

Tells EREP to: Select for the requested report only those 33XX DASD records having the specified fault symptom code. The symptom code consists of the bit settings in a two-byte field of the sense data in an OBR record for a 33XX DASD.

The combination of digits and Xs on the parameter indicate how specific you are being: if you code 4032, you want EREP to select only the records containing that exact symptom code; if you code 40XX, you want EREP to select the records containing symptom codes that begin with 40.

Syntax: **SYMCDE = {*nnnn*|*nnn*X|*nn*XX|*n*XXX}**

n is a hexadecimal digit

X is the character X.

Default: EREP processes 33XX records regardless of their symptom code bit settings.

Coding: No special considerations.

Parameter Conflicts:
DEVSER
LIA/LIBADR
TERMN
THRESHOLD
VOLID
ZERO

Note(s): If you use the SYMCDE parameter, you cannot use ZERO=Y because you have excluded some records from processing.

Example(s): Following are some ways to code SYMCDE, and the resulting bit setting EREP looks for in the OBR sense data.

Parameter Value	Bit Setting
SYMCDE=4032	0100 0000 0011 0010
SYMCDE=193X	0001 1001 0011 xxxx
SYMCDE=92XX	1001 0010 xxxx xxxx
SYMCDE=9XXX	1001 xxxx xxxx xxxx

“x” indicates either a “0” or “1” is valid.

SYSEXN

System Exception Reports (Report Parameter)

Tells EREP to: Produce the System Exception report series — several reports covering various aspects of your processing and I/O subsystems.

Syntax: **SYSEXN[=Y]|=N**

Default: Unless you specifically code SYSEXN or SYSEXN=Y, EREP does not produce a System Exception report series.

Coding: Specifying SYSEXN is the same as SYSEXN=Y.

Parameter Conflicts: DEVSER

Note(s):

- See Part 4, "Product-Dependent Information" for device-specific information about the System Exception report series.
- Unless you use DATE and/or TIME with SYSEXN, EREP processes all the available records.
- The default value for the ZERO parameter with SYSEXN is NO; EREP does not clear the ERDS unless you specifically request it.
- EREP requires a large internal sort table to create the System Exception reports; 512K is not an unreasonable TABSIZE value. The increase in TABSIZE will probably require a corresponding increase in the virtual storage (partition or region size) available to EREP. See the sections entitled "Storage Requirements" in Chapter 4, "Running EREP."
- You need the DASDID and LIMIT control statements for System Exception reports. See Chapter 8, "EREP Control Statements."

SYSUM

System Summary (Report Parameter)

Tells EREP to: Produce a System Summary — a comprehensive report of errors for each of your system's principal elements: CPU/Channel/Subchannel/Storage/SCP, and I/O subsystem.

Syntax: **SYSUM[=Y]|=N**

Default: Unless you specifically code SYSUM or SYSUM=Y, EREP does not produce a System Summary.

Coding: Specifying SYSUM is the same as SYSUM=Y.

Parameter Conflicts: DEVSER

Note(s):

- If you do not restrict record selection by date and/or time, and the input records are on the ERDS, the default value for both ACC and ZERO is YES when you request a System Summary: EREP accumulates the records to an output (ACCDEV) data set and zeroes the ERDS unless you tell it otherwise.
- If you have coded the ACCDEV data set as DUMMY, the records from the ERDS are discarded. If there is no DD statement for ACCDEV, EREP abends.

TABSIZE

Sort Table Size (Processing Parameter)

Tells EREP to: Use a sort table of the specified size to process the records selected for the report.

The sort table is EREP's internal work space, where it arranges the records into the order required for a given report.

Syntax: **TABSIZE =** *nnn***K**

nnn is a one-, two-, or three-digit decimal number.

K means the value is in thousands of bytes.

Default: For VSE systems, 4K bytes.

For MVS systems and VM, 24K bytes.

Coding: No special considerations.

Parameter Conflicts: None.

Note(s):

- For all reports except the System Exception series, each 1K (1024 bytes) of table size holds approximately 100 entries. For System Exception reports, each 1K of table size holds approximately 20 entries.
- If you increase the table size for an EREP run, you probably will need to increase the partition or region size accordingly. See the sections on storage requirements in Chapter 4, "Running EREP."

TERMN

Terminal Name (Selection Parameter)

Tells EREP to: Select for the requested report only those TCAM and VTAM OBR records that contain the specified terminal name.

VTAM OBR records are created only for local teleprocessing devices. The terminal name in these records is the NCP, or major node name. Remote attached TP devices produce only MDR records, which contain the minor node name. See Chapter 22, "Teleprocessing (TP) Devices" in Part 4, "Product-Dependent Information" for the devices to which this parameter applies.

Syntax: **TERMN**=*name*

name is the valid one- to eight-character alphameric name assigned to a particular terminal.

Default: EREP processes TCAM and VTAM OBR records regardless of the terminal name they contain.

Coding: No special considerations.

Parameter Conflicts:

DEVSER
LIA/LIBADR
SYMCDE
THRESHOLD
VOLID
ZERO

Note(s):

- If you use the TERMN parameter, you cannot use ZERO=Y because you have excluded some records from processing.
- Although TERMN applies only to TCAM and VTAM OBR records, EREP will process other types of records for the report unless you also code the appropriate DEV value and TYPE=O. See Chapter 22, "Teleprocessing (TP) Devices."

Example(s):

```
TERMN=T001

TERMN=TERM0025
```

THRESHOLD

Threshold Summary (Report Parameter)

Tells EREP to: Produce a Threshold Summary for your supported tape devices. The report is to include only those records with read/write error counts equal to or greater than the values specified on the parameter.

Syntax: **THRESHOLD**=(*xxx*,*yyy*)

xxx is the one- to three-digit decimal (leading zeros not required) threshold value for temporary read errors. Maximum value is 255.

yyy is the one- to three-digit decimal (leading zeros not required) threshold value for temporary write errors. Maximum value is 255.

Default: Unless you specifically code THRESHOLD and some threshold values, EREP produces no Threshold Summary.

Coding: You cannot code THRESHOLD alone; you also need the threshold values on the parameter.

Parameter Conflicts:

ACC
CPU
CPUCUA
DEV=(anything except 3410 or 3420 or 8809)
ERRORID
LIA/LIBADR
MOD
SHORT
SYMCDE
TERMN
TYPE
ZERO

Note(s):

- If you do not specifically code DEV=(3410) or DEV=(3420) or DEV=(8809), EREP processes records from all three device types.
- The Threshold Summary uses only OBR and MDR records; you cannot select records by type.
- You cannot code ACC=Y with THRESHOLD; EREP assumes ACC=N for this report.

- Also, you cannot code ZERO=Y with THRESHOLD; not all the records are used for the report, so EREP will not clear the ERDS even if you request it.

- For this report, EREP accumulates STARTIO (SIO) counts for records flagged as demount records.

Example(s): `THRESHOLD=(1,5) THRESHOLD=(005,015)`

TIME

Time Range (Selection Parameter)

Tells EREP to: Select only those records created during the specified time period.

Syntax: **TIME** = (*hhmm*{;}*hhmm*)

hhmm is a valid time period, hours and minutes.

Default: EREP selects records regardless of when they were created.

Coding: You must always code DATE when you are coding TIME.

Parameter Conflicts: ZERO

Note(s):

- If you use the TIME parameter, you cannot use ZERO=Y because you have excluded some records from processing.
- You code "hhmm" using a 24-hour clock (for example, 1400 for 2:00 P.M.).
- If the second hhmm value is greater than or equal to the first, the time interval pertains to each day of the date range specified on the DATE parameter. For example:

  ```
  DATE=(76001,76003),TIME=(1000,1100)
  ```

  tells EREP to select records from 10:00 to 11:00 on each of three successive days.
- If the second hhmm value is less than the first, EREP assumes that the time interval crosses a day boundary. The interval is then regarded as two sub-intervals, one ending at 2400 and the other beginning at 0000. For example:

  ```
  DATE=(76001-76003),TIME=(1100-1000)
  ```

  tells EREP to select records from 11:00 to 24:00 on day 76001; from 0:00 to 10:00 and 11:00 to 24:00 on day 76002; and from 0:00 to 10:00 on day 76003.

TRENDS

Trends Report (Report Parameter)

Tells EREP to: Produce a Trends report — a day-by-day presentation of the same information that appears in the System Summary.

Syntax: **TRENDS[=Y]|=N**

Default: Unless you specifically code TRENDS or TRENDS=Y, EREP produces no Trends report.

Coding: Specifying TRENDS is the same as TRENDS=Y.

Parameter Conflicts: DEVSER

Note(s):

- If you request a Trends report without specifying a date range on the DATE parameter, EREP processes the last 30 days of data, ending with the current date.
- If you do specify a date range, it cannot exceed 30 days.
- The default value for the ZERO parameter with TRENDS is NO; EREP does not clear the ERDS unless you specifically request it.

TYPE

Record Type (Selection Parameter)

Tells EREP to: That you want EREP to select only the specified type(s) of records.

Syntax: **TYPE** = *code*[*code*] . . .

code is one of the following:

Code	*Record Type*
A	A1 through AF records.
B	B1 through BF records.
C	CCH/CRW/SLH: Channel check/channel report word/subchannel logout records
D	DDR: Dynamic device reconfiguration records
E	System Termination (EOD): End of day and other terminating events
F	F0 through FF records.
H	MIH: Missing interrupt records
I	System Initialization (IPL): Initial program load
M	MCH: Machine check records
O	OBR: Outboard records; unit checks
S	Software (SFT): System abends and other software events.
T	MDR (formerly TPR): Miscellaneous data records.
X	C0 through CF records.
Y	D0 through DF records.
Z	E0 through EF records.

Default: EREP uses all types of records for the report.

Coding: You do not include either parentheses or commas when coding TYPE.

Parameter Conflicts: THRESHOLD
ZERO

See Figure 12-1 on page 12-2 and Part 4, "Product-Dependent Information" for some special restrictions.

Note(s):

- Care should be taken when specifying TYPE with SYSUM or SYSEXN as report results could be misleading.
- If you use the TYPE parameter, you cannot use ZERO=Y because you have excluded some records from processing.
- Some other EREP selection parameters are meaningful with only some of the record types. The following table shows these parameters and the record-type codes they work with:

Parameter	*Record Types*
CPUCUA	C, D, H, O, T
CUA	C, D, H, O, T
DEV	D, H, O, T
DEVSER	O
ERRORID	M, S
LIA/LIBADR	T
SYMCDE	O
TERMN	O
VOLID	O, T

  However, coding these selection parameters by themselves does not fully limit the types of records EREP processes; you need the TYPE parameter as well, to improve EREP's processing efficiency.

  For example, if you want a report using CCH records selected by CPUCUA, you must code TYPE=C as well as the CPUCUA parameter. Otherwise, EREP will use all the record types that contain a CPUCUA — DDR, MIH, MCH, OBR, and MDR, as well as CCH.

- If you use the TYPE selection parameter, EREP does not process records that are truncated, invalid, or unknown.

Example(s):

To select machine-check and channel-check records:

```
TYPE=MC
```

To select all software-generated records:

```
TYPE=EIS
```

VOLID

Volume Identifier (Selection Parameter)

Tells EREP to: Select only those DASD and tape records associated with the specified volume identifier(s).

Syntax: **VOLID**=(*volser*[,*volser*] . . .)

volser is a valid volume identifier (or serial number) that can be from one to six alphameric characters long.

Maximum of four entries.

Default: EREP selects DASD and tape records regardless of their volume identifiers.

Coding: No special considerations.

Parameter Conflicts:
LIA/LIBADR
SYMCDE
TERMN
ZERO

See Figure 12-1 on page 12-2 and Part 4, "Product-Dependent Information" for some special restrictions.

Note(s):

- VOLID is meaningful only for devices providing volume serial numbers.
- If you use the VOLID parameter, you cannot use ZERO=Y, because you have excluded some records from processing.
- When you are using VOLID for a Threshold Summary, EREP assumes you want to see records from all your 34XX tape devices unless you specifically code DEV=(3410) or DEV=(3420) or DEV=(8809).

Example(s):

```
VOLID=(TPONE,TPE2,),DEV=(3420),THRESHOLD=(01,15)

VOLID=(TAPE5,CLPACK),PRINT=PS
```

ZERO

Clear the ERDS (Processing Parameter)

Tells EREP to: Erase all the records from the error-recording data set.

Syntax: **ZERO[=Y]|=N**

Default: Unless you specifically code ZERO or ZERO=Y, EREP does not clear the ERDS. However, see the notes for exceptions to this rule.

Coding: Specifying ZERO is the same as ZERO=Y.

Parameter Conflicts:
CPU
CPUCUA
CUA
DATE
DEV
DEVSER
ERRORID
HIST
LIA/LIBADR
MOD
MODE[3]
SYMCDE
TERMN
THRESHOLD
TIME
TYPE
VOLID

See Figure 12-1 on page 12-2 and Part 4, "Product-Dependent Information" for some special restrictions.

Note(s):

- There are a few circumstances under which EREP does not clear the ERDS even when you code ZERO=Y. They are:
  - If an overflow occurs in the sort table or work data set.
  - If you coded ACC=Y, but the output data set cannot be opened.
  - If you coded ACC=Y, but EREP could not process all the records because of table overflow.
- If you request a System Summary report using the ERDS as input and code ACC=Y or allow it by

[3] Exception: MODE=ALL, which indicates no record selection.

default, EREP clears the ERDS even if you code ZERO=N. If your EREP run defines the ACCDEV data set as "DUMMY," the records will be lost.

- ZERO does not erase machine-check and channel-check frame records from the ERDS; they are preserved until the data set is reinitialized.
- If you code ZERO=Y when requesting PRINT=SU or PRINT=NO, EREP assumes ACC=Y and expects you to define the output data set.

Syntax Summary for EREP Parameters

ACC[=Y]|=N
Default exception: THRESHOLD

CPU=(*serial.model*[,*serial.model*] . . .)
Maximum of 6 entries.

CPUCUA=(*serial.addr*[,*serial.addr*] . . .)
Maximum of 4 entries.

CUA=(*entry*[,*entry*] . . .)
Maximum of 8 entries.

DATE=(*yyddd*[{,;}*yyddd*])
Single date or date range.

DEV=(*type*[,*type*] . . .)
Maximum of 8 entries.

DEVSER=(*serial*[,*serial*] . . .)
Maximum of 8 entries.

ERRORID=(*seqno*[,*cpuid,asid,hh,mm,ss,t*])

EVENT[=Y]|=N

HIST[=Y]|=N

LIA|LIBADR=*address*

LINECT=*nnn*

MERGE[=Y]|=N

MOD=(*model*[,*model*] . . .)
Maximum of 4 entries.

MODE={370|370XA|ALL}

PRINT={AL|DR|PS|PT|SD|SU|NO}

SHORT[=Y]|=N

SYMCDE={*nnnn*|*nnn*X|*nn*XX|*n*XXX}

SYSEXN[=Y]|=N

SYSUM[=Y]|=N

TABSIZE=*nnn*K

TERMN=*name*

THRESHOLD=(*xxx,yyy*)

TIME=(*hhmm*{,;}*hhmm*)
Time range.

TRENDS[=Y]|=N

TYPE=[A][B][C][D][E][F][H][I][M][O][S][T][X][Y][Z]

VOLID=(*volser*[,*volser*] . . .)
Maximum of 4 entries.

ZERO[=Y]|=N

Figure 7-2. Syntax Summary for EREP Parameters

Chapter 8. EREP Control Statements

You use EREP control statements in addition to EREP parameters to direct EREP processing. Control statements give EREP more information about your configuration and about how you want it to organize the report you are requesting.

As a general rule, you have to provide several control statements for each EREP run. Often, the same control statements apply to several EREP runs — when you are producing a complete set of reports for the installation for a given day or week, for example. The control statements change only when your configuration changes. In addition, some of the EREP control statements require considerable preparation before they can be added to the EREP job. For all these reasons, most large installations create a separate data set containing the control statements for their EREP run, and name that data set rather than entering the individual control statements in the input data stream.[1]

1 When EREP runs under a VSE system, all the EREP controls and system controls are read from the system input data set, SYSIPT, so a separate data set for control statements may not be feasible.

Control Statements and Reports

Some EREP control statements are fairly general-purpose, applying to most of the reports and most kinds of devices. Others are quite report-specific and product-specific.

Figure 8-1 shows which control statements you can use with the various EREP report parameters.

REPORT PARAMETERS	CONTROLLER	DASDID	LIMIT	SHARE
EVENT				
PRINT = AL				[1]
DR				YES
PS				[1]
PT				
SD				[1]
SU				YES
PRINT = NO				
SYSEXN		YES	YES	YES [2]
SYSUM				YES
THRESHOLD	YES			YES
TRENDS				YES

Figure 8-1. EREP Control Statements for EREP Reports

Notes:

1. *These PRINT options include Detail Summaries, which can include shared I/O devices.*

2. *Tape only.*

Coding Control Statements

Each EREP control statement has its own coding rules, which are detailed under the individual control statement headings. There are a few general rules, however, that you must follow:

- **Using *ENDPARM***: Control statements cannot be intermixed with EREP parameters. If parameters and control statements are in the same data set (separate or instream), you must code **ENDPARM** to indicate the end of parameters before coding any control statements. If you omit ENDPARM between parameters and control statements, EREP considers it a syntax error.

- **Entering Control Statements**: The EREP control statements must always be entered as SYSIN data (SYSIPT, for VSE systems). In the cases of VSE and MVS JCL, when there are not many control statements involved it may be easiest to do this by entering the control statements as instream data. For example, under VSE:

```
// EXEC   IFCEREP1
SYSUM,DATE=84254
ENDPARM
SHARE=(XA.0111111.032X,XA.0222222.03FX)
SHARE=(XA.0111111.09A0,XA.0222222.09A6)
/*
```

  When running EREP under MVS, the control statements need not be instream. Putting even a few control statements into a separate data set is almost always more accurate. See the SYSIN DD statement under "MVS System Controls" on page 4-32 for more information on how to do this.

  When running EREP under VM, you can specify all EREP controls as separate operands of the CPEREP command or put them in a separate data set named on the command. In either case, CPEREP sees them as instream input records.

- **Continuing Control Statements**: You cannot continue a control statement from one line to the next. However, with a few exceptions, you can code several control statements of the same type in order to convey your information to the EREP program. See the individual control statements for more details.

Figure 8-7 on page 8-22 summarizes the EREP control statements' syntax. More detailed coding considerations are in the following pages.

CONTROLLER Control Statement

The CONTROLLER statement indicates to EREP which CUAs or device addresses (that is, I/O units) are attached to a particular control unit. Using this information, EREP can combine the errors for that control unit in the report you are requesting. The CONTROLLER control statement is used primarily, if not only, for the Threshold Summary report.

Indicates: The range of device addresses (CUAs) on a control unit, allowing you to total the error counts from one control unit. You can specify up to 32 CUAs for a single control unit.

Each entry on the CONTROLLER statement defines a controller grouping; that is, the range of devices on a particular control unit. Additional entries on this and other CONTROLLER statements define other controller groupings. Every entry on a CONTROLLER statement must, therefore, define the complete set of devices attached to a particular control unit.

Syntax: CONTROLLER=(*cpuser*.{*ccua*|*ccu*X|*ccua-ccua*}[,*cpuser*.{*ccua*|*ccu*X|*ccua-ccua*}] . . .)

cpuser is a six-digit decimal CPU serial number (digits 0-9).

ccua is a three- or four-digit hexadecimal channel-control unit-device address (digits 0-F).

*ccu*X is a two- or three-digit hexadecimal channel-control unit number with X indicating all the device addresses (0-F) attached to that control unit.

ccua-ccua is a range of continuous addresses. The low end of the range must be first. The range must be at least one, and cannot exceed 32.

Default: None.

Unless you use CONTROLLER statements to request the totaling of errors by control unit, EREP presents the errors by individual device address.

Coding:

- "CONTROLLER" must be the first word in the statement, followed by an equal sign and the desired values in parentheses. No embedded blanks are allowed.
- Where the CONTROLLER statement specifies only part of the 0-F range of device addresses, and physical devices are attached to addresses in the remaining portion of the range, you should use another CONTROLLER entry to define the remaining devices, thus preventing misleading output.

- You cannot overlap device address ranges on two CONTROLLER statements. That is, when you specify a range (cpuser.ccua-ccua) of addresses on a CONTROLLER statement, you must specify that range the same way each time you use it on any other CONTROLLER statement.

- If you specify a particular processor-device address combination on a CONTROLLER statement, you cannot specify a range that includes that particular combination on that or any other CONTROLLER statement.

- Coding a range of device addresses (ccua-ccua):

  - If the control unit digit (u) in the low CUA is odd, the high CUA must have the same ccu digits.

    For example:

    `0350-0357` is valid;
    `0358-0367` is not valid.

  - If the control unit digit (u) in the low CUA is even, the high CUA must have the same even ccu digits, or the next-greater odd u digit.

    For example:

    `0368-036F` is valid;
    `0368-0377` is valid;
    `0368-0388` is not valid.

Notes:

- You can combine CONTROLLER statements with SHARE statements to make EREP combine the errors for shared devices by control unit. (See "SHARE Control Statement" on page 8-18.)

- The number of distinct CPUs (cpuser) specified on CONTROLLER statements, or on SHARE and CONTROLLER statements combined, cannot exceed 16. (System Summary uses only the first nine.)

- The CPU entries that appear on CONTROLLER statements override the default letter assignments EREP makes for processors that appear in reports. See "How EREP Assigns Letters to CPUs" on page 2-92.

Example: The following example illustrates the use of the CONTROLLER statement to define a controller grouping containing the full range of 32 devices:

```
CONTROLLER=(011111.0480-049F)
```

The result of this statement is that EREP combines the errors reported from the devices at addresses 0480 through 049F on CPU 011111 in one report entry.

DASDID Control Statement

You need DASDID control statements to identify certain DASD devices to EREP for the System Exception report series, so EREP can arrive at the correct probable failing unit (PFU). See Chapter 15, "Direct-Access Storage Devices (DASD)" for the devices involved.

The DASDID statements provide *physical identifiers* for the DASD in your installation that do not provide them directly. They also define the different paths from processors to devices, in much the same way as do SHARE statements. For this reason, the DASDID statements take the place of SHARE statements for the DASD Subsystem Exception reports; you can include the SHARE statements for DASD when you run the System Exception report series, but EREP ignores them and uses the DASDID information instead.

The DASDID control statements reflect your hardware configuration, identifying the devices in your installation and the paths (channel-storage control unit-controller) between them and the processors they work with.

In order to use DASDID statements, you will need to set them up before you request the System Exception report series. In order set up the statements, you must properly define the configuration of DASD in your installation. See "Setting up DASDID Controls" on page 8-9 for detailed directions on preparing DASDID controls.

Indicates: The paths from a processor through channel, storage control unit and controller, to each drive; to identify the device for the DASD System Exception reports.

Syntax: DASDID statements' formats differ depending on whether the processor is running in 370 mode or 370-XA mode.

The syntax of the ***370-mode DASDID control statement*** is:

DASDID CPU=*nnnnnn*,CH=*xx*,SCU=*ss*,STR=*ccuu*,STR=*ccuu*,STR=*ccuu*,STR=*ccuu*

nnnnnn is a six-digit decimal CPU serial number.

xx is a two-digit hexadecimal number identifying the channel (CH) between this CPU and the storage control unit.

ss is the physical identifier of the storage control unit (SCU). For 3880s, the number is the physical ID set for the storage director; for 3830s, it is any hexadecimal number you assign, in the range of 02-FF. Each SCU must have a unique ID number.

ccuu is a four-digit hexadecimal value representing the controller and unit address for each DASD string (STR). The DASD string is the set of eight unit addresses assigned to one controller (or pair of controllers):

cc is the number you assign, in the range of 01-FE, to each controller. Each controller must have a unique ID number; however, controllers with string switch and 3350s with alternate controllers should have only one ID number.

uu is the last two digits from the lowest address on the string. The second digit should be 0 or 8.

The format of the ***370-XA mode DASDID control statement*** is:

DASDID CPU=X*nnnnn*,CHP=*xx*,SCU=*ss*,STR=*ccddd*,STR=*ccddd*,STR=*ccddd*,STR=*ccddd*

X*nnnnn* is a five-digit decimal CPU serial number preceded by an X in the central processor (CP) identifier position.

xx is the two-digit hexadecimal number identifying the channel path identifier (CHP) between this CPU and the storage control unit.

ss is the physical identifier of the storage control unit (SCU). For 3880s, the number is the physical ID set for the storage director; for 3830s, it is any hexadecimal number you assign, in the range of 02-FF. Each SCU must have a unique ID number.

ccddd is a five-digit hexadecimal value representing the controller device number for each DASD string (STR). The DASD string is the set of eight device numbers assigned to one controller (or pair of controllers):

cc is the number you assign, in the range of 01-FE, to each controller. Each controller must have a unique ID number; however, controllers with string switch, and 3350s with alternate controllers, should have only one ID number.

ddd is the lowest device number on the string.

Default: None.

If you omit DASDID statements, those DASD that do not provide their own physical IDs are identified on the reports only by device type.

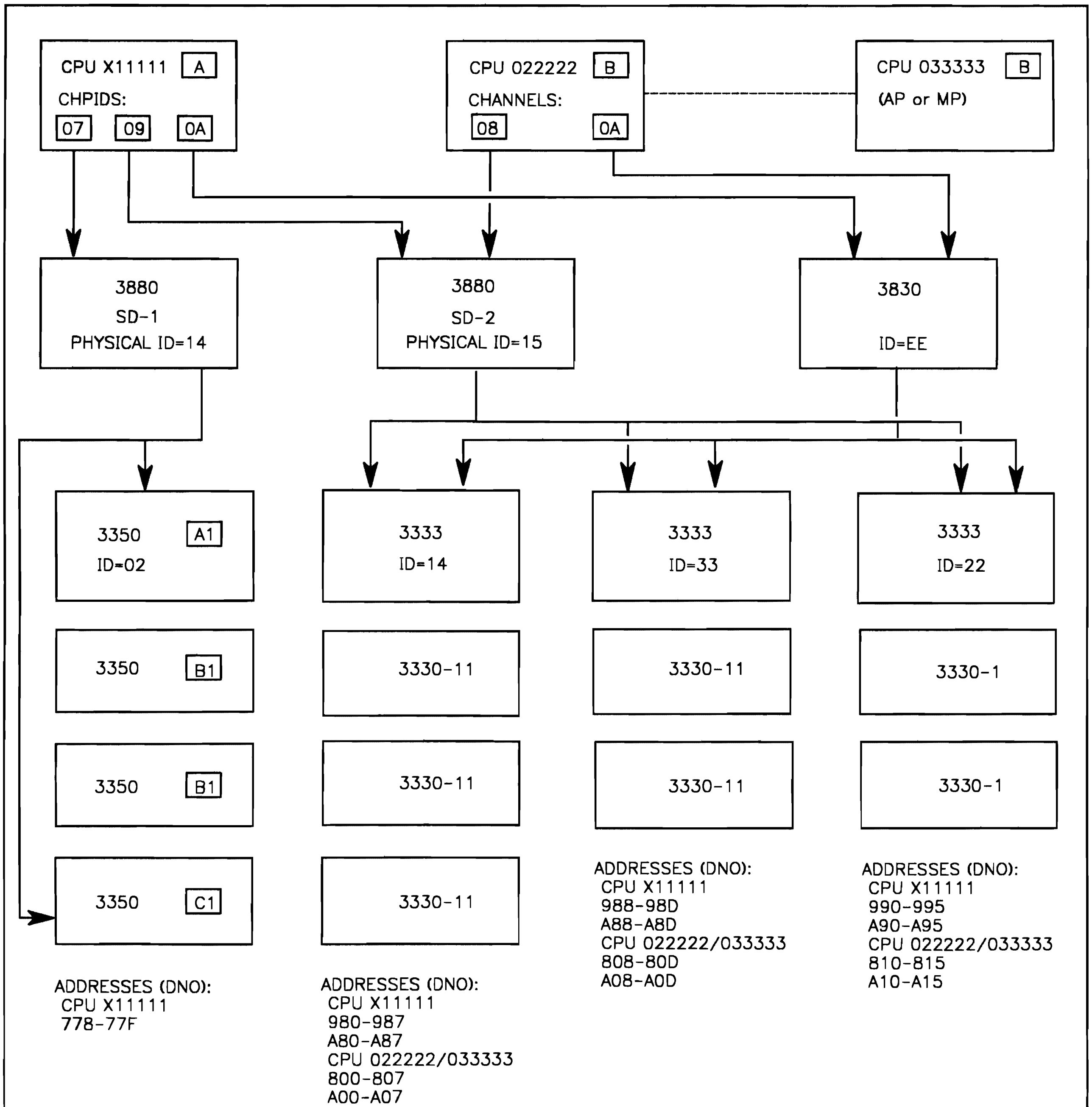
Coding: "DASDID" must be the first word in the statement, followed by one blank and the CPU= keyword with its associated value. The keywords on this statement are positional, and must be separated by commas.

Examples: You will probably never need just one DASDID statement. See the detailed description of setting up DASDID controls on the following pages for examples of DASDID statements.

Setting up DASDID Controls

You will find comment statements useful to describe a group of DASDID statements, documenting the statements for future updates. The TOURIST data set output for the System Exception report series includes the DASDID statements used and a table showing the generated configuration. This must agree with your actual configuration if the probable failing unit assignments in the System Exception reports are to be accurate.

Figure 8-2 on page 8-10 shows one way to graphically define the DASD configuration in an installation. Note that it is an example, *not* a model configuration.



Notes:

1. The PHYSICAL IDs for the 3880s (14 and 15) are those switched into the storage director.
2. The IDs for the 3830, the 3350, and the 3333s are arbitrary unique numbers that you assign.

Figure 8-2. DASD Configuration Diagram for DASDID Statements

Preparing to Code DASDID Controls

You do not need DASDID statements for DASD that provide their own physical IDs (for example, 3375s and 3380s). If you choose to code control statements for these devices, make sure the physical IDs you create match those switched into the storage director(s).

To set up DASDID controls for your DASD subsystem, you may follow three basic steps:

1. Set up a diagram of your actual DASD configuration, if you do not already have one.

   a. Show all connections between DASD controllers, storage control units, and channels or channel paths, as is done in Figure 8-2.

   b. Include any attached processors or multiprocessors that can record data on the ERDS. See CPU 033333 in Figure 8-2.

   c. Label each channel or channel path, as shown.

   d. Label the devices that have physical IDs.

   e. Create physical IDs for the devices that do not provide their own:

      1) Assign a unique ID to each 3830. Do not duplicate IDs used on other storage control units.

      2) Assign a unique ID to each controller that does not have one. Do not duplicate IDs used on other controllers.

      3) Determine the lowest unit address (or device number; the last two digits of the device address) for each string, by processor (CPU).

   f. Assign a unique label to each processor in the diagram.

2. Create a comment line (asterisk in the first column) for each storage control unit, indicating the controllers connected to it and the DASD strings connected to the controllers. For example:

   ```
   *SCU15  CTRL14,33,22  CPU A (980-995) B (800-815)
   ```

   describes one of the storage control units shown in Figure 8-2. This is storage control unit 15, which is connected to strings 980-987, 988-98D, and 990-995 from CPU A (X11111); and to strings 800-807, 808-80D, and 810-815 from CPU B (022222). The paths to the devices are through controllers 14, 33, and 22, in that order.

   a. The comment lines serve two purposes:

      1) They outline the DASDID statements.

      2) They document the DASDID statements in case of future configuration changes.

   b. The STR value in the DASDID statement consists of the controller ID and the lowest address or device number from the string attached through that controller to the CPU.

3. Create DASDID statements according to the comment lines.

   Figure 8-3 shows the completed comments and DASDID statements for the configuration shown in Figure 8-2.

```
**********************************************************************
******************INPUT FOR DASD CONFIGURATION CHART******************
*******************DISK DATASET SYSEXN.CONTROLS***********************
**********************************************************************
SYSEXN,TABSIZE=512K,HIST,DATE=84348,ACC=N
ENDPARM
**********************************************************************
* CPU DEFINITIONS  A=X11111 B=022222 and 033333
* SCU 14     CTRL 02    A(778-77F)
DASDID CPU=X11111,CHP=07,SCU=14,STR=02778
* SCU 15     CTRL 14,33,22    A(980-995)  B(800-815)
DASDID CPU=X11111,CHP=09,SCU=15,STR=14980,STR=33988,STR=22990
DASDID CPU=022222,CH=08,SCU=15,STR=1400,STR=3308,STR=2210
DASDID CPU=033333,CH=08,SCU=15,STR=1400,STR=3308,STR=2210
* SCU EE     CTRL 14,33,22    A(A80-A95)  B(A00-A15)
DASDID CPU=X11111,CHP=0A,SCU=EE,STR=14A80,STR=33A88,STR=22A90
DASDID CPU=022222,CH=0A,SCU=EE,STR=1400,STR=3308,STR=2210
DASDID CPU=033333,CH=0A,SCU=EE,STR=1400,STR=3308,STR=2210
*********************************************************************
```

Figure 8-3. Examples of DASDID Control Statements

Checking Your DASDID Statements

To check the accuracy of your DASDID statements, you can do the following:

1. Run EREP, requesting the System Exception reports. Use the following JCL (or its VSE or VM equivalent):

   ```
   //SYSEXN   EXEC  PGM=IFCEREP1,REGION=1000K,PARM='CARD'
   //ACCIN    DD    DUMMY,DCB=(RECFM=VB,BLKSIZE=3200)
   //TOURIST  DD    SYSOUT=A,DCB=BLKSIZE=133
   //EREPPT   DD    DUMMY,DCB=BLKSIZE=133
   //DIRECTWK DD    DUMMY
   //SYSIN    DD    DSN=SYSEXN.CONTROLS,DCB=(RECFM=FB,
   //               BLKSIZE=3200,LRECL=80),DISP=SHR
   /*
   ```

   The SYSIN data set consists of the EREP parameters needed to run the System Exception reports, and the comment lines and DASDID control statements developed earlier. Figure 8-3 on page 8-12 includes the contents of the SYSIN data set for the DASD configuration in Figure 8-2.

2. When the TOURIST data set (informational messages) output appears, check the configuration chart against your comment lines to be sure that your DASDID statements reflect your actual configuration. Figure 8-4 on page 8-14 shows the configuration chart produced for the DASDID statements in Figure 8-2.

The DASDID Configuration Chart

The DASDID configuration chart can only handle the more common DASD configurations. In addition to the information shown in Figure 8-4, there might be one or more notes following the chart itself. Figure 8-5 shows the possible notes and their meanings.

```
LEVEL = VERSION 3 RELEASE 1                    FEATURE LEVEL = 84251

INPUT PARAMETER STRING       HIST,ACC=N,SYSEXN

        * CPU DEFINITIONS  A=X11111 B=022222 C=033333
        * SCU 14    CTRL 02    A(778-77F)
        DASDID CPU=X11111,CHP=07,SCU=14,STR=02778
        * SCU 15    CTRL 14,33,22  A(980-995) B(800-815)
        DASDID CPU=X11111,CHP=09,SCU=15,STR=14980,STR=33988,STR=22990
        DASDID CPU=022222,CH=08,SCU=15,STR=1400,STR=3308,STR=2210
        DASDID CPU=033333,CH=08,SCU=15,STR=1400,STR=3308,STR=2210
        * SCU EE    CTRL 14,33,22  A(A80-A95) B(A00-A15)
        DASDID CPU=X11111,CHP=0A,SCU=EE,STR=14A80,STR=33A88,STR=22A90
        DASDID CPU=022222,CH=0A,SCU=15,STR=1400,STR=3308,STR=2210
        DASDID CPU=033333,CH=0A,SCU=15,STR=1400,STR=3308,STR=2210

DASDID CONFIGURATION CHART
 CPU(S) - CPUS WITH IDENTICAL CONFIGURATIONS ARE IN THE SAME COLUMN
 SCU - STORAGE CONTROL UNIT ID
 CC,CC,CC,CC - CONTROLLER IDS ORDERED BY PHYSICAL UNIT ADDRESS
 CHAN - CHANNELS WHICH CONNECT TO THE STORAGE CONTROL UNIT
 UA-UA - LOWEST PHYSICAL UNIT ADDRESS OF FIRST AND LAST STRING (370 MODE)
 DNO-DNO - LOWEST DEVICE NUMBER OF FIRST AND LAST STRING (370-XA MODE)

        CPU(S)             CPU(S)

        X11111             022222
                           033333

 SCU    CC,CC,CC,CC      CHAN  DNO-DNO     CHAN  UA-UA
 --------------------------------------------------------------------------
 14     02               07    778
 --------------------------------------------------------------------------
 15     14,33,22         09    980-990       08  00-10
 EE     14,33,22         0A    A80-A90       0A  00-10

PARAMETER OPTIONS VALID FOR THIS EXECUTION

 RECORD TYPES(MCH,CCH,OBR,SOFT,IPL,DDR,MIH,EOD,MDR,MODE ALL),
     SYSTEM EXCEPTION,HISTORY
 DATE/TIME RANGE - ALL
 TABLE SIZE - 512K,LINE COUNT - 050
 NONE

IFC221I NO SHARE CARD
IFC120I     3 RECORDS SAVED FOR SYSEXN
IFC120I       3 RECORDS THAT PASSED FILTERING
```

Figure 8-4. TOURIST Output: DASDID Configuration Chart

```
NOTE
THE SCU(S) COULD NOT BE FORMATTED. CC, CHANNEL, AND UA/DNO ARE GIVEN BY CPU.
THE SCU(S) INDICATED ABOVE COULD NOT BE FORMATTED FOR THE FOLLOWING REASON(S).

 1. THE NUMBER OF CONTROLLER IDS DOES NOT EQUAL THE NUMBER OF UA/DNOS FOR A CPU.

 2. THE CONTROLLER IDS ARE NOT THE SAME FOR ALL THE CPUS ATTACHED TO THE SCU.

 3. THE UA/DNOS FOR A CPU ARE EXPECTED TO CONSECUTIVELY INCREASE BY EIGHT.
         THIS MAY NOT NECESSARILY BE AN ERROR.

 4. THERE ARE MORE THAN FOUR UA/DNOS FOR A CPU.

 5. THERE ARE MORE THAN THREE CHANNELS FOR A CPU IN 370 MODE.

 6. THERE ARE MORE THAN FOUR CHANNELS FOR A CPU IN 370-XA MODE.

 7. THERE ARE MORE THAN FOUR CONTROLLER IDS FOR AN SCU.
```

Figure 8-5. Notes Accompanying the DASDID Configuration Chart

EXPLANATION:

1. *The program generating the configuration table found that there was not a controller ID for each set of addresses or device numbers. Because the controller ID defines a string of devices, there must be a unique controller ID for each string defined by its lowest unit address/device number. The controller ID is the first two digits of the STR parameter.*

2. *There should be only one SCU or controller assigned to a specific ID for the installation. The controller ID must be the same for a string no matter which CPU it is accessed from. Check the STR parameters to determine which string(s) have different controller IDs defined for the same string.*

3. *In order to format the unit addresses (UAs) or device numbers (DNOs) as a range (for example, 120-12F), the numbers must be consecutive. The numbers in the group were not increasing consecutively by eight; that is, the low-order digits of the UA/DNOs were more than eight apart.*

4. *A maximum of four strings can connect to one SCU (unless a switch is used). At least one CPU was found to have more than four strings defined by controller ID or unit address/device number.*

5. *The configuration generator provides space in the format for only three channels from one CPU to an SCU, in 370 mode.*

6. *The configuration generator provides space in the format for only four channel paths from one CPU complex to an SCU, in 370-XA mode.*

7. *Four is the maximum number of strings allowed per SCU.*

LIMIT Control Statement

The LIMIT control statement allows you to set error thresholds, or limits, for EREP to use for the Subsystem Exception reports. The keyword values you specify on LIMIT statements control the printing of temporary and soft (non-terminating) errors: the reports include data only for devices with errors that equal or exceed any of the limits you specify.

You can cut down on the number of records EREP uses for the System Exception reports, and, thus, on the amount of printed report output, using the LIMIT statement.

Also, the only way to see temporary error data in the *Subsystem Exception reports for your supported tape subsystems* is to set up LIMIT statements. (The *tape Temporary Error Summary* shows all temporary errors regardless of LIMIT statements.)

Indicates: The limits EREP is to apply to temporary/soft errors produced by the indicated device type or processor model, for the System Exception reports.

Syntax: LIMIT {*dasd,dkeyword*[*,dkeyword*] . . . |*tape,tkeyword*[*,tkeyword*] . . . |*cpu,ckeyword*[*,ckeyword*] . . .

dasd represents the generic device type designation for DASD products.

tape represents the generic device type designation for tape products.

cpu represents the generic machine type designation for processor products.

dkeyword represents one or more product-dependent keyword parameters with associated numeric limits.

tkeyword represents one or more product-dependent keyword parameters with associated numeric limits.

ckeyword represents one or more product-dependent keyword parameters with associated numeric limits.

Because the possible device types, keywords and numeric expressions are product-specific, their descriptions are in Part 4, "Product-Dependent Information"; see the sections for DASD, magnetic tape and processors for details.

Default: The default action for the LIMIT statement varies according to the product involved. See the discussions of the LIMIT statement in Part 4, "Product-Dependent Information."

See the discussion of the LIMIT statement in the DASD section of Part 4, "Product-Dependent Information" for ways to force EREP to include all temporary error records in the System Exception reports.

Coding: The LIMIT statement is different for each of the product groups to which it applies, so you must code it differently for each. The details are in Part 4, "Product-Dependent Information." There are a few general rules that apply, as well:

- "LIMIT" must be the first word in the statement, followed by one blank, the device or machine type, and the keyword parameters, separated by commas.
- If you code more than one LIMIT statement for a device type, EREP uses the temporary error limits set in the latest LIMIT statement; the values on a second statement override those on a previous one.

Example: See the DASD, tape, and processor sections of Part 4, "Product-Dependent Information" for the details of coding and using LIMIT statements, including examples.

Exceptions: The LIMIT statement is not valid for these devices:

9332
9335
9347

SHARE Control Statement

The SHARE control statement directs EREP to combine errors for a single device that is shared between processors or systems. The report you request then associates all the errors for that device with the device address or number you have specified, rather than with the different processors.

You can use SHARE statements for all the EREP reports except the Event History. When you include SHARE statements in your EREP controls, each report indicates whether a particular set of error data represents a device that you specified on SHARE statements.

You can also use SHARE statements to influence the way EREP assigns alphabetic identifiers to the processors that appear on its reports. This is described in "How EREP Assigns Letters to CPUs" on page 2-92.

Indicates: The possible paths to a device being shared between processors.

Syntax: SHARE=([XA.]*cpuser*.{*ccua*|*ccu*X|*ccua-ccua*}[,[XA.]*cpuser*.{*ccua*|*ccu*X|*ccua-ccua*}] . . .)

[XA.]*cpuser* is a six-digit decimal CPU serial number (digits 0-9). As *cpuser*, it indicates the processor is running in 370 mode; as **XA.***cpuser*, it indicates the processor is running in 370-XA mode.

ccua is a three- or four-digit hexadecimal channel-control unit-device address or device number (digits 0-F). The first digit is the channel designated to the operating system as "primary"; hence, this is the primary CUA for the device.

*ccu***X** is a two- or three-digit hexadecimal channel-control unit address. The X implies device addresses 0 through F.

ccua-ccua is a range of continuous addresses. The low end of the range must be first. The range must be at least one, and cannot exceed 32.

Default: None.

If you omit this control statement, EREP presents each device's error records individually by device type in its reports.

Coding:

- "SHARE" must be the first word on the statement, followed by the equal sign and the desired values in parentheses. You must put at least two entries (cpuser.ccua|ccuX|ccua-ccua) on each statement.

- The number of distinct CPUs specified (cpuser) on SHARE statements cannot exceed 16. (System Summary uses only the first 10; see "System Summary Part 1" on page 2-2.)

  Should you use both SHARE and CONTROLLER statements, to make EREP group errors for shared devices by both device and control unit, the total number of CPUs on SHARE and CONTROLLER statements combined cannot exceed 16.

- You might need more than one SHARE statement to show all the possible paths to one device. If so, you should repeat the first entry on the statements for the remaining paths, because EREP equates all the paths on the SHARE statement to the one you specify first. For example:

```
SHARE=(011111.01F0,022222.0330,022222.06F0,022222.0FF0)
SHARE=(011111.01F0,033333.03F0,033333.0630,033333.0F30)
```

- The "cpuser" values on SHARE statements override EREP's default CPU letter assignment, which is in ascending alphabetical order starting with the first model/serial number encountered. See "How EREP Assigns Letters to CPUs" on page 2-92.

- Once you have specified a range (cpuser.ccua-ccua) on a SHARE statement, you must specify that range the same way each time you use it on any other SHARE statement. This prevents the overlapping of ranges.

Examples: See the following section for more detailed information, including examples.

Using SHARE Statements to Combine Data in EREP Reports

Figure 8-6 is an example of the kind of I/O configuration that could require SHARE statements. The text on the next page explains how you would set up SHARE controls for the illustrated configuration.

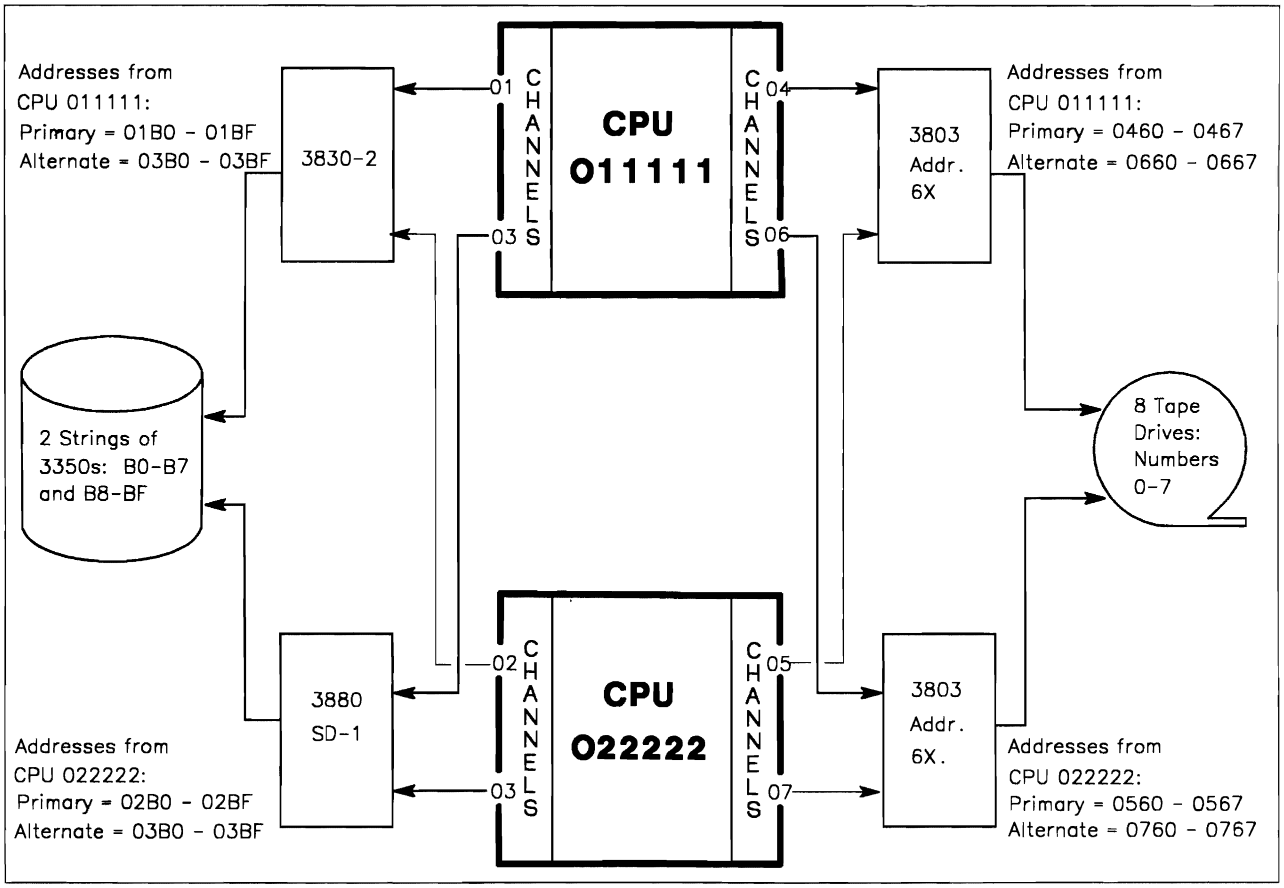


Figure 8-6. Configuration for SHARE Statements

SHARE Statements for DASD Drives

To cause EREP to combine all records for the DASD drives in the strings, use:

```
SHARE=(011111.01BX,022222.02BX)
```

OR

```
SHARE=(011111.01B0-01BF,022222.02B0-02BF)
```

Either of these SHARE statements causes the records from DASD drive 0 (device addresses/numbers 01B0 and 02B0) to be combined and presented as data for 01B0 on CPU 011111.

Without the SHARE statements, the records would be presented by the primary channel address for each processor. That is, records for drive 0 on CPU 011111 would be presented as 01B0, regardless of whether they were recorded on channel 01 or 03; and records for drive 0 on CPU 022222 would be presented as 02B0, regardless of whether they were recorded on channel 02 or 03.

SHARE Statements for Tape Drives

To cause EREP to combine all records for the tape drives in the strings, use:

```
SHARE=(011111.0460-0467,022222.0560-0567)
```

This SHARE statement causes all records from drive 7 (device address/numbers 0467 and 0567) to be combined and presented as data for 0467 on CPU 011111.

Without the SHARE statements, the records would be presented by the primary channel address for each processor. That is, records for drive 5 on CPU 011111 would be presented as 0465, regardless of whether they were recorded on channel 04 or 06; and records for drive 5 on CPU 022222 would be presented as 0565, regardless of whether they were recorded on channel 05 or 07.

Syntax Summary for EREP Control Statements

CONTROLLER = (*cpuser*.{*ccua*|
*ccu***X**|*ccua-ccua*}[,*cpuser*.{*ccua*|*ccu***X**|*ccua-ccua*}] . . .)

The syntax of the 370-mode DASDID control statement is:

DASDID CPU = *nnnnnn*,**CH** = *xx*,**SCU** = *ss*,**STR** = *ccuu*,**STR** = *ccuu*,**STR** = *ccuu*,**STR** = *ccuu*

The syntax of the 370-XA mode DASDID control statement is:

DASDID CPU = **X***nnnnn*,**CHP** = *xx*,**SCU** = *ss*,**STR** = *ccddd*,**STR** = *ccddd*,**STR** = *ccddd*,**STR** = *ccddd*

LIMIT {*dasd,dkeyword*[*,dkeyword*] . . . |*tape,tkeyword*[*,tkeyword*] . . . |*cpu,ckeyword*[*,ckeyword*] . . .

SHARE = ([XA.]*cpuser*.{*ccua*|*ccu***X**|*ccua-ccua*}[,[XA.]*cpuser*.{*ccua*|*ccu***X**|*ccua-ccua*}] . . .)

Figure 8-7. Syntax Summary for EREP Control Statements

Chapter 9. CPEREP Operands - Syntax and Coding

To run EREP under a VM system, you use the CMS command, CPEREP. CPEREP has the following syntax:

```
CPEREP [filename filetype {filemode|*}]
```

You must have the proper user privilege class to run CPEREP. See the note under "Entering CPEREP Operands" on page 9-8 for more information.

You control EREP under VM using the same parameters and control statements that you use for running EREP under MVS or VSE; however, those controls become *operands* for the CPEREP command.

Using the CPEREP Command

You have several options for entering and executing the CPEREP command:

1. Enter CPEREP by itself, and allow the system to prompt you for the individual operands you want to specify. An example of this method is in "Prompting Method" on page 9-10.

2. Create a file, using the system editor, that contains the operands for this execution of CPEREP. Include the name, type and mode of the file on the CPEREP command, as shown in the syntax. This method is used in the VM master examples in Chapter 4, under "Running EREP under VM."

3. Within a CMS EXEC, use &STACK control statements to enter the operands as instream data before the EXEC statement. An example of this is in "Stacked Entry Method" on page 9-13.

4. Combine the above methods in various ways. This option is discussed in more detail in "Mixed Entry Method" on page 9-14.

See the User's Guide for your version of CMS for more information on coding CMS EXECs and executing CMS commands.

Unique CPEREP Operands

In addition to the regular EREP parameters and control statements, you need two other operands for running EREP via CPEREP. These are the CLEARF operand and the TERMINAL operand, both of which act as processing controls.

CLEAR/CLEARF operand
Clears (zeroes) and reinitializes the error-recording area (ERDS).

Note: You must have the proper user privilege class to erase and reinitialize the error-recording area. In most VM installations, this is privilege class F.

If there are SRF frame records on the ERDS for 303X processors, you use CLEARF to rewrite the frames on the data set. CLEAR/CLEARF is never combined with other EREP controls; you are reinitializing the ERDS. See "Information about the VM SCP" on page 24-8 and the discussion of the 303X processors in Chapter 20, "Processors (CPUs)" for more about the ERDS and frame records.

TERMINAL operand
Instructs CPEREP to stop reading EREP parameters and control statements from a separate file and to prompt the user for them instead. You use this CPEREP operand to change your EREP controls dynamically. Having set up an input file containing CPEREP operands, you have the choice of using those operands or overriding them with others entered from the terminal. See "Mixed Entry Method" on page 9-14 for more information.

Using EREP Controls as CPEREP Operands

The same rules and restrictions apply to EREP controls regardless of the system they are being used for. However, when you enter them as CPEREP operands you must also follow the rules imposed by the VM facility. "Entering CPEREP Operands" on page 9-8 lists those rules along with several possible ways to present the operands to CPEREP. The table on the next few pages lists all the EREP controls/operands in aphabetical order and indicates where you can find more detailed information on each one. See Figure 9-2 on page 9-7 for a summary of the operands' syntax. Examples of their use are in "Entering CPEREP Operands" on page 9-8.

CPEREP OPERAND	*TYPE OF EREP CONTROL*	*NOTES AND REFERENCE*
ACC	An EREP processing parameter.	When you use this operand, you must also make sure the ACCDEV file has been defined. See "ACC" on page 7-5 and "Defining Files for CPEREP" on page 4-49 for more information.
CLEAR/CLEARF	An operand unique to CPEREP.	Use this operand, by itself, to clear *and reinitialize* the error-recording area. See "VM System Controls" on page 4-49.
CONTROLLER	An EREP control statement — combining errors from devices associated with a single controller.	See "CONTROLLER Control Statement" on page 8-4 for more information.
CPU	An EREP selection parameter — selecting records by processor model and serial number.	See "CPU" on page 7-6 for more information.
CPUCUA	An EREP selection parameter — selecting records by processor serial number and device address.	See "CPUCUA" on page 7-7 for more information.
CUA	An EREP selection parameter — selecting records by device address.	See "CUA" on page 7-8 for more information.
DASDID	An EREP control statement — identifying DASD units that do not provide their own physical IDs.	This operand is for the System Exception report series. You will need to set up your DASDID controls before running CPEREP, and will probably want to put these and other control statements into a separate data set. See "DASDID Control Statement" on page 8-7 and "Entering CPEREP Operands" on page 9-8 for more information.
DATE	An EREP selection parameter — selecting records by date.	See "DATE" on page 7-10 for more information.

Figure 9-1 (Part 1 of 4). EREP Controls as CPEREP Operands

CPEREP OPERAND	*TYPE OF EREP CONTROL*	*NOTES AND REFERENCE*
DEV	An EREP selection parameter — selecting (or excluding) records by device type.	See "DEV" on page 7-11 for more information.
DEVSER	An EREP selection parameter — selecting records by the tape device serial number.	This operand is used for the Threshold Summary report. See "DEVSER" on page 7-13 for more information.
ENDPARM	The delimiter required between EREP parameters and control statements.	If you are entering both parameters and control statements, the parameters must precede ENDPARM and the control statements must follow it. See "Coding Control Statements" on page 8-3 and the examples on the following pages for more information.
ERRORID	An EREP selection parameter — selecting MCH and SFT records by software error ID.	This operand applies to records created by an MVS guest system. See "ERRORID" on page 7-15 for more information.
EVENT	An EREP report parameter — requesting an Event History report.	See "EVENT" on page 7-17 and "Event History Report" on page 2-12 for more information.
HIST	An EREP processing parameter — indicating that input records for EREP reports are in a history file rather than on the ERDS.	When you use this operand, you must also make sure the ACCIN and DIRECTWK files have been defined. See "HIST" on page 7-18 and "Defining Files for CPEREP" on page 4-49 for more information.
LIA/LIBADR	An EREP selection parameter — selecting records by line interface base address.	This operand applies to records associated with 3705 and 3725 communications controllers. See "LIA/LIBADR" on page 7-19 for more information.
LIMIT	An EREP control statement — setting threshold values for the System Exception report series.	The syntax of this control statement varies widely from one product (processor/channel, DASD, tape) to another. See "LIMIT Control Statement" on page 8-16 and the appropriate sections in Part 4 for more information.
LINECT	An EREP processing parameter — indicating the number of lines to be printed on a page of output.	The default value when running EREP under VM is 50. See "LINECT" on page 7-20 for more information.
MERGE	An EREP processing parameter — indicating that input records for reports are to be taken from both the ERDS and a history file.	When you use the MERGE operand, you must also make sure the ACCIN, DIRECTWK and SERLOG files have been defined. See "MERGE" on page 7-21 and "Defining Files for CPEREP" on page 4-49 for more information.

Figure 9-1 (Part 2 of 4). EREP Controls as CPEREP Operands

CPEREP OPERAND	*TYPE OF EREP CONTROL*	*NOTES AND REFERENCE*
MOD	An EREP selection parameter — selecting records by processor model number.	See "MOD" on page 7-22 for more information.
PRINT	An EREP report parameter — requesting one or more Detail reports of specific records and/or specific devices.	Use PRINT=NO to prevent CPEREP from producing any report output. See "PRINT" on page 7-24 and "Detail Edit and Summary (PRINT) Reports" on page 2-62 for more information.
SHARE	An EREP control statement — combining errors associated with shared devices.	You will probably want to put SHARE and other control statements into a single data set. See "SHARE Control Statement" on page 8-18 and the examples on the following pages for more information.
SHORT	An EREP processing parameter — indicating whether short OBR records are to be printed.	Use this operand to suppress the printing of short OBR records in Detail Edit reports. See "SHORT" on page 7-26 for more information.
SYMCDE	An EREP selection parameter — selecting 33XX DASD records by fault symptom code.	See "SYMCDE" on page 7-27 for more information.
SYSEXN	An EREP report parameter — requesting the System Exception report series.	This operand produces a group of reports about major processing and I/O subsystems. See "SYSEXN" on page 7-28 and "System Exception Reports" on page 2-20 for more information.
SYSUM	An EREP report parameter — requesting a System Summary.	See "SYSUM" on page 7-29 and "System Summary" on page 2-2 for more information.
TABSIZE	An EREP processing parameter — indicating the amount of storage to be used for EREP's sort tables.	A TABSIZE value substantially larger than 24K could require a corresponding increase in the storage size of the virtual machine that CPEREP runs in. See "TABSIZE" on page 7-30 and the sections on storage requirements in Chapter 4 for more information.
TERMINAL	An operand unique to CPEREP processing — requesting that the operator be prompted for CPEREP operands.	See "VM System Controls" on page 4-49 and the discussion on the following pages for more information.
TERMN	An EREP selection parameter — selecting records by terminal name.	This operand applies only to TP OBR records for Event History and Detail PRINT reports. See "TERMN" on page 7-31 and Chapter 22, "Teleprocessing (TP) Devices" for more information.

Figure 9-1 (Part 3 of 4). EREP Controls as CPEREP Operands

CPEREP OPERAND	*TYPE OF EREP CONTROL*	*NOTES AND REFERENCE*
THRESHOLD	An EREP report parameter — requesting a Threshold Summary for some tape devices.	A dual-purpose operand; you request the report and set the error-frequency thresholds at the same time. See "THRESHOLD" on page 7-32 and "Threshold Summary Report" on page 2-58 for more information.
TIME	An EREP selection parameter — selecting records by time.	See "TIME" on page 7-34 for more information.
TRENDS	An EREP report parameter — requesting a Trends report.	See "TRENDS" on page 7-35 and "Trends Report" on page 2-7 for more information.
TYPE	An EREP selection parameter — selecting records by record type.	See "TYPE" on page 7-36 for more information.
VOLID	An EREP selection parameter — selecting records by volume ID (volume serial number).	This operand applies to 33XX DASD and 34XX tape records only. See "VOLID" on page 7-38 for more information.
ZERO	An EREP processing parameter — indicating whether the records on the ERDS (error-recording area) are to be erased when CPEREP has completed a report.	This operand must be used carefully. See "ZERO" on page 7-39 for more information.

Figure 9-1 (Part 4 of 4). EREP Controls as CPEREP Operands

ACC[=Y]|=N

CLEAR|CLEARF
Unique to CPEREP; see "VM System Controls" on page 4-49.

CONTROLLER=(*cpuser*.{*ccua*|*ccu*X|*ccua-ccua*}[,*cpuser*.{*ccua*|*ccu*X|*ccua-ccua*}] . . .)

CPU=(*serial.model*[,*serial.model*] . . .)

CPUCUA=(*serial.addr*[,*serial.addr*] . . .)

CUA=(*entry*[,*entry*] . . .)

DASDID CPU=*nnnnnn*,**CH**=*xx*,**SCU**=*ss*,**STR**=*ccuu*,**STR**=*ccuu*,**STR**=*ccuu*,**STR**=*ccuu*
370 mode:

DASDID CPU=X*nnnnn*,**CHP**=*xx*,**SCU**=*ss*,**STR**=*ccddd*,**STR**=*ccddd*,**STR**=*ccddd*,**STR**=*ccddd*
370-XA mode:

DATE=(*yyddd*[{;}*yyddd*])

DEV=(*type*[,*type*] . . .)

DEVSER=(*serial*[,*serial*] . . .)

ERRORID=(*seqno*[,*cpuid,asid,hh,mm,ss,t*])

EVENT[=Y]|=N

HIST[=Y]|=N

LIA|LIBADR=*address*

LIMIT {*dasd,dkeyword*[,*dkeyword*] . . . |*tape,tkeyword*[,*tkeyword*] . . . |*cpu,ckeyword*[,*ckeyword*] . . .

LINECT=*nnn*

MERGE[=Y]|=N

MOD=(*model*[,*model*] . . .)

MODE={370|370XA|ALL}

PRINT={AL|DR|PS|PT|SD|SU|NO}

SHARE=([XA.]*cpuser*.{*ccua*|*ccu*X|*ccua-ccua*}[,[XA.]*cpuser*.{*ccua*|*ccu*X|*ccua-ccua*}] . . .)

SHORT[=Y]|=N

SYMCDE={*nnnn*|*nnn*X|*nn*XX|*n*XXX}

SYSEXN[=Y]|=N

SYSUM[=Y]|=N

TABSIZE=*nnn*K

TERMINAL=[=Y]|=N
Unique to CPEREP; see "VM System Controls" on page 4-49.

TERMN=*name*

THRESHOLD=(*xxx,yyy*)

TIME=(*hhmm*{;}*hhmm*)

TRENDS[=Y]|=N

TYPE=[A][B][C][D][E][F][H][I][M][O][S][T][X][Y][Z]

VOLID=(*volser*[,*volser*] . . .)

ZERO[=Y]|=N

Figure 9-2. Syntax of the CPEREP Operands

Entering CPEREP Operands

In order to use the CPEREP command, you must be in the CMS environment and have a user privilege class that allows access to the records in the error-recording area.

Note: With the latest releases of VM, it is possible for an installation to redefine the privilege classes in effect for its systems, overriding those set by IBM. Make sure you have the proper privilege class for this operation, whether it is C, E, F, or some other class.

Figure 9-2 on page 9-7 shows the syntax of all the EREP parameters as CPEREP operands — essentially, identical to the way you would code them to run EREP under VSE or MVS.

However, when running CPEREP, you cannot include the operands on the command line, because many of them exceed the record length allowed for CMS commands. Instead, you enter the operands using one of the following methods:

- You can enter the operands individually in response to prompting by the system
- You can put all the operands for a single report in a separate file whose name you include on the CPEREP command line
- You can use the &STACK control statement to enter the operands automatically
- You can use combinations of two of the above.

The next several sections of this chapter describe in detail the methods of entering operands for CPEREP, including examples of each.

Invoking CPEREP

You can supply operands to CPEREP in any of several ways, depending on how automated you want the procedure to be. Regardless of the way you enter the operands, however, the invocation of CPEREP is the same. The entire sequence for invoking CPEREP is as follows:

1. Log on to a virtual machine that has been established for the CE (normally, a Class F userID).
2. IPL CMS.
3. Issue FILEDEF statements for the files required by EREP, if they are not already defined to the system.
4. Have the system operator mount any required tape volumes for use as input and output data sets.
5. Issue the CPEREP command.
6. Enter CPEREP operands via one of the methods detailed on the next several pages.

Coding Rules

When entering operands, you must follow these rules:

- Separate a keyword and its associated values from a following keyword operand by one or more blanks, or by a comma. (You can put each parameter/control statement on a separate line.)
- Enter embedded commas, periods, and parentheses that define the extent of variable operands as shown in the CPEREP operand syntax in Figure 9-2 on page 9-7.
- You may enter operands in any sequence.
- When specifying an operand where the allowed values are Y and N, you may omit =Y, coding only the keyword itself. CPEREP always interprets this form of the operand as specifying YES, regardless of the default value.

Prompting Method

The basic way to supply operands for CPEREP is to respond to its prompting messages. After you type CPEREP on the command line and press ENTER, the system prompts you with:

```
ENTER PARAMETER STATEMENTS OR NULL TO PROCESS
```

You then type in your CPEREP operands. If you fill up the entire command line before all the operands are entered, press ENTER again for another prompting message and a clear command line. You can continue in this way until all your operands for that report are entered. When finished, press ENTER to signal with a null command line the end of the string of operands.

To invoke CPEREP using only EREP's default values, respond to the first prompting message by pressing ENTER, to enter a null line.

Prompting Method Example

The following example shows a complete CPEREP operation as initiated from the virtual machine console, starting with the logon step. It illustrates the prompting method of entering CPEREP operands.

Lowercase letters indicate entries by the terminal user; uppercase letters indicate system responses.

When execution is complete, the TOURIST output appears on the terminal screen unless you have requested another output class for the TOURIST data set.

Note: *Any ICF**** messages in the TOURIST output are from EREP; all the EREP messages are documented in Chapter 11, "EREP Messages." CMS may also issue messages in the course of EREP processing, prefixed with DMSIFC or DMSREA. These, too, are in Chapter 11, "EREP Messages," under "CPEREP Messages for VM Users" on page 11-48.*

In this example, the requested report output, consisting of Detail Edits and Summaries of all records containing the device type code for the 2310 DASD, is sent to the system printer. The records used for the report are copied onto the data set at virtual address TAPE 181.

CONSOLE ACTION	*COMMENTS*
`logon . . .`	User logs on to the CE userID.
`ipl cms`	User IPLs CMS

`.`
`.`
`.`
`R;`
	The system indicates successful initialization of CMS.

`mount accum 181   please put ring in.`	User requests a tape for CPEREP use. 181 is the virtual address of the default ACCDEV tape.
`TAPE 181 ATTACHED`	Note that, if both HIST and ACC functions are to be used, you must also request that the history tape be attached at address 182.

`fi tourist terminal`
`fi directwk disk erep cmsutl a`
`fi ereppt print`
`fi accdev tape sl (recfm vb blksize 12000`
`fi accin disk erep hist a (recfm vb`
	User defines the files needed to run EREP from VM. You can allow these to default to VM system FILEDEFS. See "Defining Files for CPEREP" on page 4-49.

`cperep`	Invoking CPEREP, without naming a control file. The operation defaults to the prompting method of operand entry.
`ENTER PARAMETER STATEMENTS OR NULL TO PROCESS`	The system prompts the user for EREP parameters and control statements as operands for the CPEREP command.
`print=ps hist acc=y dev=(2310)`	User enters CPEREP operands
`ENTER PARAMETER STATEMENTS OR NULL TO PROCESS`	The system prompts again for operand input.
`endparm` `share=XA.cpuser.ccuX,share=XA.cpuser.ccuX`	User enters more operands.
`ENTER PARAMETER STATEMENTS OR NULL TO PROCESS`	The system prompts again for operand input.
`*`	User enters a null line to start processing.
`.` `.` `.`	
`PRT FILE 2546 FROM userid SENT TO printaddr NOHOLD`	The end of EREP processing is signalled by the CMS PRINT message. Any TOURIST messages generated by the EREP program or CMS would appear on the terminal screen because of the TOURIST FILEDEF.

File Entry Method

Create a file, using the system editor, that contains the operands you want in effect for this execution of CPEREP, and include the name, type and mode of the file on the CPEREP command line. Make sure there is a FILEDEF in effect for SYSIN with the same data set name as your operand file. The operands are arranged in the file according to the rules listed under "Entering CPEREP Operands" on page 9-8. Note that input records are truncated at column 71.

To invoke CPEREP using only EREP's default values, issue the SYSIN FILEDEF for an empty file.

In practice, you will probably want to have several different files containing the various operand combinations needed to run CPEREP for your installation. See the User's Guide for your system editor for information on how to create a file for the operands.

File Entry Method Example

The following EXEC illustrates the use of a separate file to enter CPEREP operands.

```
&TRACE ERR
FILEDEF EREPPT PRINTER (NOCHANGE BLOCK 133 PERM
FILEDEF TOURIST PRINTER (NOCHANGE BLOCK 133 PERM
FILEDEF DIRECTWK DISK EREP CMSUT1 &DISK?
FILEDEF ACCIN DISK HIST RECORDS A (RECFM VB
FILEDEF SYSIN DISK EREP PARMS A (RECFM F
EXEC CPEREP EREP PARMS A
&EXIT
```

The file named EREP PARMS A contains:

```
HIST
ACC=N
TABSIZE=500K
SYSEXN
ENDPARM
SHARE=(020402.0736,220402.0736)
SHARE=(020402.0735,220402.0735)
LIMIT 33XX,ALL=15
LIMIT 3420,HR1600=025(1),HW1600=010(15)
LIMIT 3420,VR1600=025(1),VW1600=010(15)
DASDID CPU=020402,CH=07,SCU=14,STR=0238
```

In this example the requested System Exception report output and the TOURIST message output both go to the printer. The series of sample EXECs earlier in this chapter all use this method of entering operands. It avoids the necessity of re-typing complex EREP control statements each time you invoke CPEREP.

Note, too, that the records for EREP to process are in a history file on disk.

Stacked Entry Method

The CMS EXEC &STACK control statement allows you to enter commands or operands as instream data before coding the CPEREP command. It is another way to avoid having to re-code parameters and control statements each time you run EREP.

You precede each operand by "&STACK," one to each input record. CPEREP reads the operands in the order in which you have stacked them.

To invoke CPEREP from an EXEC using only EREP's default values, code only a null &STACK statement in the EXEC.

Stacked Entry Method Example

The following example illustrates the use of the &STACK control statement within a CMS EXEC to enter operands for the CPEREP command.

```
&TRACE ERR
FILEDEF EREPPT PRINTER (NOCHANGE BLOCK 133 PERM
FILEDEF TOURIST TERMINAL (NOCHANGE BLOCK 133
FILEDEF DIRECTWK DISK EREP CMSUT1 &DISK?
FILEDEF ACCIN DISK HIST RECORDS A (RECFM VB
&STACK HIST
&STACK ACC=N
&STACK TABSIZE=500K
&STACK SYSEXN
&STACK ENDPARM
&STACK SHARE=(020402.0736,220402.0736)
&STACK SHARE=(020402.0735,220402.0735)
&STACK LIMIT 33XX,ALL=15
&STACK LIMIT 3420,HR1600=025(1),HW1600=010(15)
&STACK LIMIT 3420,VR1600=025(1),VW1600=010(15)
&STACK DASDID CPU=020402,CH=07,SCU=14,STR=0238
&STACK
EXEC CPEREP
&EXIT
```

In this example, the EREP controls are inline, to be read by CPEREP in the order they are listed. Note the null &STACK statement following the DASDID statement. Without it, CPEREP would prompt the terminal user for more EREP control statements.

The report output produced by this example is sent to the printer, but the TOURIST output appears on the terminal screen.

See the User's Guide for your version of CMS for information about coding and using EXECs.

Mixed Entry Method

You can combine the previous methods of entering CPEREP operands to make the process more efficient, yet flexible. For example, enter the CPEREP command followed by the name of a file containing operands, one of which is the TERMINAL operand. CPEREP reads the operands from the named file until it reaches TERMINAL. At this point, it reads no more input from the file; instead, it prompts the terminal user for operands. You can thus enter operands at the time of EREP's execution to dynamically tailor a report to your immediate requirements.

Note that the position of the TERMINAL operand is important: if it follows ENDPARM, you can enter only EREP control statements when prompted.

Not coding a null &STACK statement following a series of stacked operands has the same effect; unless it encounters a null &STACK statement, CPEREP prompts the terminal user for further input after reading in the stacked operands.

Mixed Entry Method Examples

The following examples show two ways to enter operands using the mixed entry method.

1. With the TERMINAL operand in an input file:

   ```
   &TRACE ERR
   FILEDEF EREPPT PRINTER (NOCHANGE BLOCK 133 PERM
   FILEDEF TOURIST PRINTER (NOCHANGE BLOCK 133 PERM
   FILEDEF DIRECTWK DISK EREP CMSUT1 &DISK?
   FILEDEF ACCIN DISK HIST RECORDS A (RECFM VB
   FILEDEF SYSIN DISK EREP PARMS A (RECFM F
   EXEC CPEREP EREP PARMS A
   &EXIT
   ```

   The file named EREP PARMS A contains:

   ```
   PRINT=PT
   HIST
   ACC=N
   TYPE=M
   TERMINAL
   ```

   This EXEC invokes CPEREP to produce Detail Edit reports of all the MCH records in the history file (ACCIN — HIST RECORDS A). The reports and TOURIST output are being sent to the printer.

   The TERMINAL operand following the other EREP parameters causes CPEREP to prompt the user for more input. You could then enter a DATE parameter, for example, or the CPU parameter, to narrow further the selection of records from the history file.

   You could also enter SHARE or CONTROLLER control statements, if appropriate (they aren't, in this example). In that case, you would first have to enter ENDPARM.

2. With no null &STACK statement in a CMS EXEC:

```
&TRACE ERR
FILEDEF EREPPT PRINTER (NOCHANGE BLOCK 133 PERM
FILEDEF TOURIST PRINTER (NOCHANGE BLOCK 133 PERM
FILEDEF DIRECTWK DISK EREP CMSUT1 &DISK?
FILEDEF ACCIN DISK HIST RECORDS A (RECFM VB
&STACK PRINT=PT HIST ACC=N
EXEC CPEREP
&EXIT
```

This EXEC produces Detail Edit reports of all the records on input file HIST RECORDS A. In the absence of a null &STACK statement following the statement with parameters, CPEREP will prompt for more input. Then, you could specify the TYPE parameter, or DATE and/or TIME, or DEV, to limit the records EREP processes. Following ENDPARM, you could also enter SHARE or CONTROLLER statements.

In this example, the report and TOURIST output are being sent to the printer. To change the destination of output, you must change the FILEDEFs for the output files. See "Defining Files for CPEREP" on page 4-49 for more information.

Chapter 10. Error Records for EREP

This chapter contains reference information about the records EREP uses to produce its reports.

The Error-Recording Process

Each S/370 operating system writes error and operational records to its error-recording data set (ERDS). The records are created primarily for the hardware — processors and devices — that make up the S/370 data processing installation, although the operating system also creates some records to document its own processing.

The ERDS is different for each operating system:

- For VSE, it is system logical unit SYSREC, file name IJSYSRC, residing on the SYSRES disk. The data set is initialized by the IPL command SET RF=CREATE. See the *System Management Guide* for your VSE system.

- For MVS systems, it is system data set SYS1.LOGREC, residing on the system residence volume.[1] The data set is initialized by the IFCDIP00 service aid during system generation, and can be reinitialized at IPL. See the *System Generation Reference* manual for your MVS system.

- For VM systems, it is the error-recording area assigned on the system residence volume and initialized during system generation. See the *Planning and System Generation* guide for your VM facility.

---

1 For MVS/XA, LOGREC can be another cataloged data set, and need not be on the system residence volume.

Record Formats

The records on a system's ERDS must conform to a standard of both format and content, regardless of the system that records them. EREP assumes that the records it receives as input conform to the standard; it has only one map for each type of record.

In this section of the book are mappings of the error and operational records. The contents of the various fields in the records are identified and described in general terms, because that is how EREP sees them.

The mapping used to present the records in this section is:

Offset Dec(Hex)	Size (Bytes) Alignment (Bits)	Field Name	Description

Offset is the numeric address of the field relative to the beginning of the data area.

Dec(Hex) is the offset in decimal, followed by the hexadecimal equivalent in parentheses. For example: 16(10).

Size (Bytes) is the field size in bytes.

Alignment (Bits) shows the bit settings of switch or flag fields, as follows:

`.... ....`	indicates the eight bit positions (0-7) in a byte. For ease of scanning, the high-order (left-hand) four bits are separated from the low-order four bits.
`x... ....`	is a reference to bit 0.
`1... ....`	indicates that bit 0 is on.
`0... ....`	indicates that bit 0 is off.
`.... ..xx`	is a reference to bits 6 and 7.

The record mappings include significant bit settings. Bits described as "reserved" are not significant for this release.

Field Name is a label (acronym) that identifies the field.

Description indicates how the field is used. Where the field's use relates directly to a value you would code, the coded value is shown. Where the hexadecimal code for a particular bit setting would be helpful, it is shown separated from the rest of the description.

ERDS Formats

The error-recording data sets for the different operating systems are similarly formatted. Each has a header record, followed by frame records if the installation includes 303X processors, followed by the individual error and operational records. The characteristics of each operating system dictate the exact format of the ERDS; the system manual that describes error-recording procedures provides more specific information. Following are brief descriptions of the parts of the ERDS not taken up by error records.

The ERDS Header Record

SYSREC, SYS1.LOGREC, and VM's error-recording area (cylinders) each start with a ***header record*** that provides information to the system recording routines about the device on which the ERDS resides, where to write new records, and when the data set is getting full. The figures on the next few pages represent the header records for each system:

- Figure 10-1 on page 10-4 shows the header record for SYSREC on count-key-data (CKD) devices
- Figure 10-2 on page 10-5 shows the header record for SYSREC on fixed-block-architecture (FBA) devices
- Figure 10-3 on page 10-6 shows the header record for SYS1.LOGREC
- Figure 10-4 on page 10-7 shows the header record for the VM error-recording cylinder(s).

The CKD Header Record for SYSREC

This figure shows the format of the header record when IJSYSRC is on a count-key-data device.

Offset Dec(Hex)	Size (bytes) Alignment (bits)	Field Name	Description
0(0)	2	CLASRC	Header record identifier. This field is set to X'FF00' unless critical data has been destroyed.
2(2)	4	LOWLIMIT	Address of low extent. Track address (in CCHH format) of first extent of SYSREC.
6(6)	4	UPLIMIT	Address of high extent. Track address (in CCHH format) of last extent of SYSREC.
10(A)	1	TRKSPER	Highest addressable track for each cylinder on the volume containing SYSREC.
11(B)	7	RESTART	Address of record entry area. Starting track address (in BBCCHHR format) for recording area on SYSREC.
18(12)	2	BYTSREM	Remaining bytes on track: number of bytes remaining on the track upon which the last record entry was written.
20(14)	2	TRKCAP	Total bytes on track. Number of bytes that can be written on a track of the volume containing SYSREC.
22(16)	7	LASTTR	Address of last record written. Track address (in BBCCHHR format) of last record written on SYSREC.
29(1D)	2	PUBNUM	Number of PUBS in the system.
31(1F)	2	EWMCNT	Warning count. Number of bytes remaining on early warning message track of SYSREC when 90% full point of data set is reached. When this is detected by a recording routine, it issues a message and turns on the early-warning-message switch at displacement 38.
33(21)	1	DEVCODE	Device code. Code indicating device type of system volume on which SYSREC resides: **Code Device** 01 2311 02 2301 03 2303 04 2302 06 2305 MOD 1 07 2305 MOD 2 08 2314 09 3330 and 3333 MOD 1 or 3350 operating in 3330-1 compatibility mode 0A 3340 and 3344 0B 3350 native mode 0C 3375 0D 3330 and 3333 MOD 11 or 3350 operating in 3330-11 compatibility mode
34(22)	4	EWMTRK	Early warning message track. Track address (in CCHH format) on which the 90% full point will be found.
38(26)	1	EWMSW	Switch byte:
	1...		90% full point message has been issued. This switch is turned on by recording routine detecting 90% full point and is turned off by IFCEREP1 when clearing SYSREC.
	.1..		An emergency recording has occurred. This switch is turned on when the system terminates because SYSREC is full.
	..1.	FRAMES	Machine-check and channel-check frames exist on SYSREC.
	...x xxxx		Reserved.
39(27)	1	SFTYBYT	Check byte. Each bit in this field is set to 1 (X'FF'); used to check the validity of the header-record identifier.

Figure 10-1. CKD SYSREC Header Format

The FBA Header Record for SYSREC

This figure shows the format of the header record when IJSYSRC is on a fixed-block-architecture device.

Offset Dec(Hex)	Size (bytes) Alignment (bits)	Field Name	Description
0(0)	2	CLASRC	Header record identifier. This field is set to X'FF00' unless critical data has been destroyed.
2(2)	4	LOWLIMIT	Address of low extent. Block number of the first extent of SYSREC.
6(6)	4	UPLIMIT	Address of high extent. Block number of the last extent of SYSREC.
10(A)	1		Reserved.
11(B)	4	RESTART	Address of record entry area. Block number of the start of the recording area of SYSREC.
15(F)	7		Reserved.
22(16)	4	LSTREC	Address of last record. Block number of the last record written on the recording area.
26(1A)	7		Reserved.
33(21)	1	DEVCODE	X'0F'. Device code for FBA device.
34(22)	4	EWMTRK	Early-warning-message block. Block number on which the 90%-full point will be found.
38(26)	1	EWMSW	Switch byte:
	1...		90%-full-point message has been issued. This switch is turned on by recording routine detecting 90% full point and is turned off by IFCEREP1 when clearing SYSREC.
	.1..		An emergency recording has occurred. This switch is turned on when the system terminates because SYSREC is full.
	..1.	FRAMES	Machine-check and channel-check frames exist on SYSREC.
	...x xxxx		Reserved.
39(27)	1	SFTYBYT	Check byte. Each bit in this field is set to 1 (X'FF'); used to check the validity of the header-record identifier.

Figure 10-2. FBA SYSREC Header Format

The Header Record for SYS1.LOGREC

Offset Dec(Hex)	Size (bytes) Alignment (bits)	Field Name	Description
0(0)	2	CLASRC	Header-record identifier. Each bit in this field is set to 1 unless critical data has been destroyed.
2(2)	4	LOWLIMIT	Address of low extent. Track address (in CCHH format) of first extent of SYS1.LOGREC.
6(6)	4	UPLIMIT	Address of high extent. Track address (in CCHH format) of last extent of SYS1.LOGREC.
10(A)	1	MSGCNT	Count of the number of times LOGREC-full message has been issued (maximum is 15).
11(B)	7	RESTART	Address of record entry area, and address of time-stamp record. Starting track address (in BBCCHHR format) of the recording area on SYS1.LOGREC. If a time-stamp record is present, it begins at the address pointed to by this field.
18(12)	2	BYTSREM	Remaining bytes on track. Number of bytes remaining on the track upon which the last record entry was written.
20(14)	2	TRKCAP	Total bytes on track. Number of bytes that can be written on a track of the volume containing SYS1.LOGREC.
22(16)	7	LASTTR	Address of last record written. Track address (in BBCCHHR format) of last record written on SYS1.LOGREC.
29(1D)	2	TRKSPER	Highest addressable track for each cylinder on volume containing SYS1.LOGREC.
31(1F)	2	EWMCNT	Warning count. Number of bytes remaining on early-warning-message track of SYS1.LOGREC when 90%-full point of data set is reached. When this is detected by a recording routine, it issues a message and turns on the early-warning-message switch at displacement 38.
33(21)	1	DEVCODE	Device code, indicating the device type of the volume on which SYS1.LOGREC resides: **Code Device** 01 2311 (not supported by VS) 02 2301 03 2303 04 2302 06 2305 MOD 1 07 2305 MOD 2 08 2314 09 3330 and 3333 MOD 1 or 3350 operating in 3330-1 compatibility mode 0A 3340 and 3344 0B 3350 native mode 0C 3375 0D 3330 and 3333 MOD 11 or 3350 operating in 3330-11 compatibility mode 0E 3380
34(22)	4	EWMTRK	Early-warning-message track. Track address (in CCHH format) on which 90% full point for data set exists.
38(26)	1	EWMSW	Switch byte:
	1...		90%-full-point message has been issued. This switch is turned on by the recording routine detecting 90% full point and is turned off by IFCEREP1 when clearing SYS1.LOGREC.
	..1.	FRAMES	Machine-check and channel-check frames exist on SYS1.LOGREC.
	.x.x xxxx		Reserved.
39(27)	1	SFTYBYTS	Check byte. Each bit in this field is set to 1 (X'FF'); the field is used to check the validity of the header-record identifier.

Figure 10-3. SYS1.LOGREC Header Record

The VM Error-Recording Area (Cylinder) Header

Offset Dec(Hex)	Size (bytes) Alignment (bits)	Field Name	Description
0(0)	4	RECCCPD	Address of this cylinder.
4(4)	2	RECNXT	Displacement to the next available space for records.
6(6)	1	RECFLAG1	Record usage flags:
	1...	RECPAGIU	The page contains valid data.
	.1..	RECPAGFR	The page is cleared. This bit is set by EREP when it clears the error-recording area.
	..1.	RECPAGFL	The page is full. When this bit is set, a message is issued to the operator to clear the error-recording area.
	...1	RECPAGER	The next page is unreadable.
	.... 1...	RECPAGFA	Frame records exist for this page.
	xxx		Reserved.
7(7)	1	RECFLAG2	Record format flags:
	1...	RECPAGFM	The cylinder is being formatted. This bit is turned on in the first page of a recording cylinder while the cylinder is being formatted. The field is reset only when all pages are cleared.
	0000 0000	RECPAGDN	The cylinder has been formatted. If this field is non-zero, the cylinder is in the process of being formatted.

Figure 10-4. VM Error-Recording Cylinder Header

Other ERDS Data

Time-Stamp Record for IPL Records

For an MVS or MVS/XA system, the first record following the ERDS header on SYS1.LOGREC is a time-stamp record used by the error-recording routines when they are building IPL records.[2] The current date and time information in an IPL record allows you to measure the interval between system termination and re-initialization.

The time-stamp record consists of a standard 24-byte header plus 16 bytes that are reserved for system use. Within the header, at offsets 8 and 12, are the system date and time fields. These fields are updated at pre-set intervals, so the date and time are kept current.

When building an IPL record, the recording routines take the current date and time from the time-stamp record and put them in the system date and time fields of the IPL record header.

Frames for 303X Machine- and Channel-Check Records

When an installation includes processors from the 303X family, the ERDS contains special frame records for EREP to use in processing and formatting the MCH and CCH records from the processors. The frames govern the formatting of model-dependent data.

The frame records are written in a reserved area of the ERDS. They are not erased or changed by the EREP ZERO parameter; you must reinitialize the ERDS in order to change the frame records.

The frame records consist of a standard 24-byte record header followed by 1920 bytes of frame instructions that direct EREP in processing the MCH and CCH records for its reports.

When VSE writes frame records, it puts identifying information into the record-dependent switches in the record header, so the records can be subdivided if necessary.

See Chapter 20, "Processors (CPUs)" in Part 4, "Product-Dependent Information" for more specific information about the 303X frame records and how they are updated.

---

[2] In early versions of the MVS system, time-stamp recording only takes place when the system includes the RDE option. In later versions, some of the functions of RDE are included in the base control program.

Information in Error and Operational Records

The records on the system's ERDS are of two types:

1. **Hardware and software errors**, which reflect the failure and recovery of processors, channels, I/O devices, and operating system software.

2. **Software operational data**, which indicates the time and circumstances of the failures, as well as of other conditions.

While the records reflect different events and are of different lengths, they all contain the following kinds of information:

- Relevant system information at the time the record was generated
- Device hardware status at the time the record was generated
- Results of any device/control unit recovery attempt
- Results of any software system recovery attempt
- Statistical data about device usage and/or recoverable errors

Each record begins with a standard 24-byte header, which contains the information needed to identify the type and origin of the record:

- **Type information** includes the specific type of the record, the specific source of the record, the general reason the record was created, and special record-dependent data.

- **Origin information** includes the operating system under which the record was generated, the date and time the record was generated, and the identity of the processor (CPU) on which the record was generated.

Note: *For CCH, MCH and OBR records, the processor generating the record is also the processor associated with the error. In a tightly-coupled multiprocessing environment, this is not necessarily true for other types of records.*

The Standard Record Header: Data Common to All Record Types

The origin information and some of the type information is the same for all records; Figure 10-5 shows the contents of the fields that are the same. In the mappings on later pages, the record header (the first 24 bytes of the record) is separated from the rest of the record by the phrase **END OF STANDARD HEADER**.

Offset Dec(Hex)	Size (bytes) Alignment (bits)	Field Name	Description
1(1)	1	xxxKEY2	System/release level:
	000.		OS/360.
	001.		DOS (VSE systems).
	010.		OS/VS1.
	011.		CP67, VM/370, and later VM systems.
	100.		OS/VS2 and later MVS systems.
	...x		Reserved.
	 xxxx		Release level (0-1F).
2(2)	1	xxxSMS	Record-independent switches:
	1...		More records follow.
	0...		Last record.
	.x..		Time-of-Day clock instruction issued (0 = IBM System/360, 1 = IBM System/370). Used in conjunction with date and time values at displacements 8 and 12.
	..1.		Record truncated.
	...1		370-XA mode record.
	.... 1...		TIME macro used (MVS).
	.... 0...		Time in timer units (VSE).
	xxx		Reserved.
.	.	.	.
.	.	.	.
.	.	.	.
6(6)	1	xxxRCDCT	Record count:
	xxxx		Sequence number of this physical record.
	 xxxx		Total number of physical records in this logical record.
.	.	.	.
.	.	.	.
.	.	.	.
8(8)	8	xxxDT	System date and time, as:
8(8)	4	xxxDATE	System date of failure.
12(C)	4	xxxTIME	System time of failure.
16(10)	8	xxxCPUID	CPU identification, as:
16(10)	1	xxxVER	Machine version code:
	xxxx xxx.		Reserved.
	0		Version I CPUs.
	1		Version II CPUs.
17(11)	3	xxxSER	CPU serial number.
20(14)	2	xxxMOD	CPU machine model number (3033, 4341, etc.).
22(16)	2	xxxCEL	Maximum length of machine- (CPU-) dependent machine-check extended logout area.

Figure 10-5. Header Data Fields Common To All Records

The Record Type/Class Codes

The first field in the standard record header is a one-byte hexadecimal code that identifies the type (or class) and source of the record.

While all three S/370 operating systems create identical records, they do not all record every possible kind of record. Some record types are not relevant for all S/370 operating systems; and some operating systems simply do not support the recording of all record types. See "VM Notes" in Part 4, "Product-Dependent Information."

Figure 10-6 on page 10-12 shows the record types that each of the operating systems records on its ERDS, listed according to the record class code.

Note that, when VM writes a record, it can be doing so in its own behalf or on behalf of another operating system running in a virtual machine.

Note, also, that MVS creates different versions of some records.

Record Type Code and Description	VSE	MVS	VM
1X Machine Check (MCH)			
10 MCH	X	X	X
13 MCH in multiple storage environment		X	X[2]
2X Channel Check (CCH)			
20 CCH	X	X	X[3]
21 CCH in multiple storage environment		X[1]	
23 SLH subchannel logout		X[4]	
25 CRW channel report word		X[4]	
3X Unit Check (OBR)			
30 OBR	X	X	X[2]
34 TCAM OBR		X	
36 VTAM OBR	X	X	
3A DPA OBR		X	
4X Software Error (SFT)			
40 Software-detected		X	
42 Hardware-detected		X	
44 Operator-detected		X	
48 Hardware-detected hardware		X	
4F Lost record		X	
5X System Initialization (IPL)			
50 IPL	X	X	
6X Reconfiguration (DDR)			
60 DDR		X[1]	X[5]
7X Missing Interrupt (MIH)			
70 MIH	X[6]	X[1]	X[3]
71 MIX		X[4]	
8X Recovery/Termination			
80 EOD	X	X	
81 Non-restartable wait state (MCH)		X[1]	
84 Restartable wait state (IOS)		X[1]	
9X Non-Standard (MDR)			
90 SVC	X	X	
91 Miscellaneous data recorder	X	X	X[2]
A0 MCH frame	X	X	X
Ax A1 through AF records	X	X	X
B0 CCH frame	X	X	X
Bx B1 through BF records	X	X	X
Cx C0 through CF records	X	X	X
Dx D0 through DF records	X	X	X
Ex E0 through EF records	X	X	X
Fx F0 through FF records	X	X	X

Figure 10-6. Record Types and Systems Recording Them

Notes:

1. *MVS only*
2. *For both VM and the virtual machine*
3. *For VM only; SVC 76 is reflected back to the virtual machine*
4. *MVS/XA only*
5. *For the virtual machine only*
6. *VSE/Advanced Functions only*

Record Mappings

The figures on the following pages show the layouts of each of the general record types, accompanied by general descriptions. The record formats are in alphabetical order.

Information about frame records (A0 and B0) is under "Frames for 303X Machine- and Channel-Check Records" on page 10-8.

Channel Check (CCH) Record

In the S/370 environment, the operating system writes a CCH record when a channel failure occurs but does not terminate the system control program. The errors recorded include channel control checks, channel data checks, and interface control checks.

MVS writes an extended version of the CCH record; see offset 80.

Figure 10-7 shows the format of the CCH record.

Offset Dec(Hex)	Size (bytes) Alignment (bits)	Field Name	Description
0(0)	1	CCHKEY1	Class/Source:
	..1.		CCH record; type = X'20'.
	..1. ...1		CCH record recorded in multiple virtual storage environment (MVS and later VS2 releases); type = X'21'.
1(1)	1	CCHKEY2	System/release level.
	000.		OS/360.
	001.		DOS (VSE systems).
	010.		OS/VS1.
	011.		CP67, VM/370, and later VM systems.
	100.		OS/VS2 and later MVS systems.
	...x		Reserved.
	 xxxx		Release level (0-1F).
2(2)	1	CCHSMS	Record-independent switches.
	1...		More records follow.
	0...		Last record.
	.x..		Time-of-Day clock instruction issued (0 = IBM System/360, 1 = IBM System/370). Used in conjunction with date and time values at displacements 8 and 12.
	..1.		Record truncated.
	...1		370-XA mode record.
	.... 1...		TIME macro used (OS/VS).
	.... 0...		Time in timer units (VSE).
	.... .1..	CCHPASS	Passed, recovery status. VM passed error to guest.
	xx		Reserved.
3(3)	2		Record-dependent switches:
	Byte 0	CCHSW1	Switch 1 (all systems):
	1...		Operator message required.
	.1..		Record incomplete.
	..1.		System terminated (Not used by MVS).
	...1		Channel unsupported or failed to log.
	.... 1...		Illegal CUA.
	.... .1..		Portion of data overlaid.
	.... ..1.		ERP in progress.
	x		Reserved.
	Byte 1	CCHSW2	CCH internal switches from PCCA (MVS only):
	1...		Command register parity is valid.
	.1..		No recording by CCH.
	..1.		CCH FRR in FRR stack.
	...1		Record SYS1.LOGREC record only.
	.... 1...		Attention bit in CSW is on.
	.... .1..		ERPIB has already been created for error.
	.... ..1.		UCB is invalid.
	x		Reserved.
5(5)	1	CCHSWS	CCH internal switches from PCCA (MVS only):
	1...		I/O restart function required.
	.1..		Alternate return to I/O supervisor requested (CCH retry not required).
	..1.		Channel analysis module is unavailable to CCH to analyze error.
	...1		Channel failed to log.
	.... 1...		Channel availability table (CAT) entry is valid, but channel type is not recognized.
	.... .1..		Channel reconfiguration hardware (CRH) active on channel at time of channel error.
		CCHSFTSW	Recovery status bits. See "3090 Processor" on page 20-7.
	00		Invalid or, for VM, passed to guest.
	01		Hard fail.
	10		Degrade fail.
	11		Soft fail.
6(6)	1	CCHRCDCT	Record count:
	xxxx		Sequence number of this physical record.
	 xxxx		Total number of physical records in this logical record.
7(7)	1		Reserved.
8(8)	8	CCHDT	System date and time of failure. *See Figure 10-5 on page 10-10 for the details in these fields.*
16(10)	8	CCHCPUID	CPU identification:
16(10)	1	CCHVER	Machine version code.

Figure 10-7 (Part 1 of 3). Channel Check (CCH) Record Format

Offset Dec(Hex)	Size (bytes) Alignment (bits)	Field Name	Description
17(11)	3	CCHSER	CPU serial number. *Note: If bit 5 of switch 3 (CCHSWS) is set, indicating channel reconfiguration hardware (CRH) is active, the following information applies (for IBM Model 168 MP CPUs only):* *1. The CPU serial number field (offset 17 decimal) is that of the "live" or running CPU and not that of the CPU to which the channel is actually attached (i.e., the "dead" CPU).* *2. The channel set ID and channel status fields of the MP information area (offset 88 decimal plus a variable logout length identified by labels CCHCMPPA and CCHCMPCS) always reflect the address and channel status of the channel set connected to the dead CPU. That is, the first entry for these fields is for the dead CPU, and the second entry is for the live CPU (the one on which CRH receives control).*
20(14)	2	CCHMOD	CPU machine model number (3033, 4341, etc.).
22(16)	2	CCHCEL	Maximum length of machine- (CPU) dependent, machine check extended logout area. **END OF STANDARD HEADER**
24(18)	8	CCHJOBID	Alphameric name assigned to job (as identified, for example, by a jobname on a JCL JOB statement) being executed and/or requesting service at time of channel-detected error.
32(20)	16	CCHATIO	List of active I/O units or addresses, one to eight devices, on failing channel that were found to be busy (device end outstanding). List includes device address associated with failure.
48(30)	8	CCHFCCW	Last real CCW executed before failure.
56(38)	8	CCHCSW	Contents of CSW that was stored following detection of I/O failure.
64(40)	4	CCHECSW	Contents of extended CSW (limited channel logout), or last four bytes of ERPIB for 28XX channels.
68(44)	4	CCHCODE	Code for the device type of the failing device. The first two bytes are VSE PUB bytes 4 and 5; the second two bytes are MVS UCB device class and type bytes 18 and 19 (decimal). See "Device Type Codes in the OBR Record" on page 12-13 for a list of the UCB codes.
72(48)	1	CCHTYPE	Channel type associated with the failing channel: **Code Meaning** 00 Channel unknown (CCH provides a 155II channel analysis by default, assuming the channel adheres to System/370 channel design.) 01 Integrated multiplexor (MPX) 02 Integrated selector 03 Integrated block MPX 04 Reserved 05 Standalone selector (2860) 06 Standalone MPX (2870) 07 Standalone block MPX (2880) 08 Selector channel (2880) 0A Integrated file adaptor 0F Channel unknown
	3	CCHCUA	The channel and unit address of the last path used to address the failing device — the failing path and the path over which the sense was issued and data retrieved.
76(4C)	2	CCHCUA2	The last two bytes, right-justified, of the CUA stored by hardware in machine storage locations 186-187 (decimal). VSE systems set bit 0 of the first byte on to indicate to EREP that the I/O device is invalid.
78(4E)	2	CCHLOG	Maximum length of the channel logout that begins at offset 80 or 84 (decimal).
80(50)	4	TIOADDR	In EC-mode records: CUA, right-justified, as stored by hardware in machine storage locations 186-187. In records with this field, the channel logout begins at offset 84. In VSE and MVS records, and for 28XX and 303X channels, there is no TIOADDR field, and the CHNLOG field begins at offset 80.

Figure 10-7 (Part 2 of 3). Channel Check (CCH) Record Format

Offset Dec(Hex)	Size (bytes) Alignment (bits)	Field Name	Description
80(50)	Variable	CHNLOG	Machine-dependent channel logout associated with failure that caused channel check. Logout size is model- and channel-dependent: **Channel** **Length (Bytes)** 2860 24 2870 24 2880 112 135 24 145 96 (maximum) 155II/158 0 165II/168 0 303X 576 43XX 0 308X 0 3090 0 ***The remainder of this record is created only by MVS.***
Variable	2	CCHCFT	CCH footprints: for use in debugging suspected CCH problems.
	4		Reserved.
	2	CCHMPNO	Number of online processors (CPUs), in hexadecimal.
	Variable	CCHMP	Multiprocessing information that consists of a variable number of CCHMPPA and CCHMPCS fields. The number of fields corresponds to the number of online CPUs in field CCHMPNO. *Note: The following two fields, CCHMPPA and CCHMPCS, represent a format that is repeated a variable number of times within field CCHMP. The first CCHMPPA/CCHMPCS pair belongs to the CPU with the failing channel.*
	2	CCHMPPA	Channel set ID or CPU address.
	2	CCHMPCS	Channel status for each channel (0-15) associated with the system. Each channel, beginning with channel 0 as the high-order bit, is represented by a one-bit code (0 = online, 1 = offline).

Figure 10-7 (Part 3 of 3). Channel Check (CCH) Record Format

Channel Report Word (CRW) Record

In a S/370-XA environment, the channel report word (CRW) describes channel incidents reported through machine checks. The CRW specifies the error environment and the severity of the error.

As part of the recovery process, MVS/XA assigns a unique sequence number to each CRW it retrieves, and includes the number in the CRW record it builds for SYS1.LOGREC. If the same error environment produces more than one CRW, MVS/XA associates subsequent sequence numbers with the number it assigned to the first CRW.

Figure 10-8 shows the format of the CRW record created by MVS/XA.

Offset Dec(Hex)	Size (bytes) Alignment (bits)	Field Name	Description
0(0)	1	CRWKEY1	Class/Source:
	..1. .1.1		CRW record; type = X'25'.
1(1)	1	CRWKEY2	System/release level:
	000.		OS/360.
	001.	DOS (VSE systems).	
	010.		OS/VS1.
	011.		CP67, VM/370, and later VM systems.
	100.		OS/VS2 and later MVS systems.
	...x		Reserved.
	 xxxx		Release level (0-1F).
2(2)	1	CRWSMS	Record-independent switches:
	1...		More records follow.
	0...		Last record.
	.x..		Time-of-Day clock instruction issued (0 = IBM System/360, 1 = IBM System/370). Used in conjunction with date and time values at displacements 8 and 12.
	..1.		Record truncated.
	...1		370-XA mode record.
	.... 1...		TIME macro used (MVS).
	.... 0...		Time in timer units (VSE).
	xxx		Reserved.
3(3)	3		Record-dependent switches:
	Byte 0	CRWBYTE1	Reserved.
	Byte 1	CRWBYTE2	Reserved.
	Byte 2	CRWBYTE3	Reserved.
6(6)	1	CRWRCDCT	Record count:
	xxxx		Sequence number of this physical record.
	 xxxx		Total number of physical records in this logical record.
7(7)	1		Reserved.
8(8)	8	CRWDT	System date and time, as:
8(8)	4	CRWDATE	System date of failure.
12(C)	4	CRWTIME	System time of failure.
16(10)	8	CRWCPUID	CPU identification, as:
16(10)	1	CRWVER	Machine version code:
	xxxx xxx.		Reserved.
	0		Version I CPUs.
	1		Version II CPUs.
17(11)	3	CRWSER	CPU serial number.
20(14)	2	CRWMOD	CPU machine model number (3033, 4341, etc.).
22(16)	2	CRWCEL	Maximum length of machine- (CPU-) dependent machine-check extended logout area.
			END OF STANDARD HEADER

Figure 10-8 (Part 1 of 2). Channel Report Word (CRW) Record Format

Offset Dec(Hex)	Size (bytes) Alignment (bits)	Field Name	Description
24(18)	8	CRWMODUL	CSECT name of module doing recording.
32(20)	1	CRWRECCD	CRW recording code; identifies the format of the variable portion of the record.
33(21)	1	CRWFLAG1	Flag byte 1:
	1...	CRWHARD	Hardware-stored CRW.
	.1..	CRWSOFT	Software-created CRW.
	..xx xxx.		Reserved.
	1	CRWINVAL	Invalid CRW.
34(22)	1	CRWFLAG2	Flag byte 2.
35(23)	1	CRWCODE	CRW Origin Code:
	0000 0000		CRW origin unknown.
	0000 0001		CRW-pending machine check.
	0000 0010		System-damage machine check.
	0000 0011		Alternate CPU recovery.
	0000 0100		Reserved.
	0000 0101		Reserved.
	0000 0110		Hot I/O: Recover channel path.
	0000 0111		Hot I/O: Remove channel path.
	0000 1000		Vary channel path - forced.
	X'09' - X'FF'		Reserved.
36(24)	2	CRWCP	Processor address on which the CRW was retrieved.
38(26)	2		Reserved.
40(28)	4	CRWCRW	Channel report word.
44(2C)	2	CRWDEV	Device number (binary).
46(2E)	2		Reserved.
48(30)	Variable	CRWDEPEN	Device- and system-dependent data. For example, for MVS/XA, the fields contain:
48(30)	4	CRWSEQNO	CRW sequence number.
52(34)	4	CRWASEQN	Associated CRW sequence number.
56(38)	2	CRWDEVST	UCB device status flags; zero if UCB not available.
58(3A)	2	CRWPMCW	Path management control word; zero if UCB not available.
60(3C)	1	CRWCHPCT	Channel path recovery count; zero if UCB not available.
61(3D)	2		Reserved.
63(3F)	1	CRWLEVEL	UCB level value; zero if UCB not available.
64(40)	4	CRWLVMSK	UCB level bit mask; zero if UCB not available.
68(44)	4	CRWSCHRC	UCB subchannel recovery anchor; zero if UCB not available.
72(48)	1		Reserved.
73(49)	1	CRWICHPT	ICHPT flags associated with the CRW channel path ID.
74(4A)	8	CRWISDT	Copy of the IOS interrupt subclass definition table.

Figure 10-8 (Part 2 of 2). Channel Report Word (CRW) Record Format

Dynamic Device Reconfiguration (DDR) Record

MVS creates a DDR record for each operator-initiated or system-initiated swap between direct-access devices having buffered logs and demountable disk packs (such as the IBM 3330, 3330 MOD 11, and 3340 devices), and between magnetic tape devices.

The DDR record identifies the physical devices involved in the swap. Figure 10-9 shows the format of the DDR record.

Offset Dec(Hex)	Size (bytes) Alignment (bits)	Field Name	Description
0(0)	1	DDRKEY1	Class/source:
	.11.		DDR record; type = X'60'.
1(1)	1	DDRKEY2	System/release level:
	000.		OS/360.
	001.		DOS (VSE systems).
	010.		OS/VS1.
	011.		CP67, VM/370, and later VM systems.
	100.		OS/VS2 and later MVS systems.
	...x		Reserved.
	 xxxx		Release level (0-1F).
2(2)	1	DDRSMS	Record-independent switches:
	1...		More records follow.
	0...		Last record.
	.x..		Time-of-Day clock instruction issued (0 = IBM System/360, 1 = IBM System/370). Used in conjunction with date and time values at displacements 8 and 12.
	..1.		Record truncated.
	...1		370-XA mode record.
	.... 1...		TIME macro used (MVS).
	.... 0...		Time in timer units (VSE).
	xxx		Reserved.
3(3)	3		Record-dependent switches:
	Byte 0	DDRSW2	
	1...		Primary storage reconfiguration.
	.1..		Secondary storage reconfiguration.
	..1.		Operator requested reconfiguration.
	...1		Permanent error caused reconfiguration.
	 xxxx		Reserved.
	Bytes 1 and 2		Reserved.
6(6)	1	DDRRCDCT	Record count:
	xxxx		Sequence number of this physical record.
	 xxxx		Total number of physical records in this logical record.
7(7)	1		Reserved.
8(8)	8	DDRDT	System date and time of incident:
8(8)	4	DDRDATE	System date of failure.
12(C)	4	DDRTIME	System time of failure.
16(10)	8	DDRCPUID	CPU identification, as:
16(10)	1	DDRVER	Machine version code:
	xxxx xxx.		Reserved.
	0		Version I CPUs.
	1		Version II CPUs.
17(11)	3	DDRSER	CPU serial number.
20(14)	2	DDRMOD	CPU machine model number (3033, 4341, etc.).
22(16)	2	DDRCEL	Maximum length of machine- (CPU-) dependent machine-check extended logout area.
			END OF STANDARD HEADER
24(18)	8	DDRJOBID	Name of job using 'FROM' device. Valid only for a system-initiated swap for a permanent error or operator-initiated tape swaps.
32(20)	6	DDRVOL1	Volume ID (VOLSER) of volume mounted on 'FROM' swap device.
38(26)	6	DDRVOL2	Volume ID (VOLSER) of volume mounted on 'TO' swap device. Field is zero if no volume is mounted on 'TO' device.
44(2C)	1	DDRFPHD	Physical ID (not address) of "FROM" device. (DASD only.)
45(2D)	3	DDRFCUA	Primary CUA or device number of 'FROM' device.
48(30)	4	DDRFDEV	Device type of 'FROM' device.
52(34)	1	DDRTOPHD	Physical ID (not address) of 'TO' device. (DASD only.)
53(35)	3	DDRTOCUA	Primary CUA or device number of 'TO' device.
56(38)	4	DDRTODEV	Device type of 'TO' device.

Figure 10-9. Dynamic Device Reconfiguration (DDR) Record Format

Recovery/Termination (EOD) Record

This record, type X'8X', is used differently by different operating systems:

- Both MVS and VSE systems write type X'80' records to indicate that the system terminated normally under program control, at the request of the operator (HALT EOD). This orginal record type is the source for the "EOD" prefixes on field names in the mapping.

- MVS and MVS/XA also write type X'8X' records to document abnormal terminations:

  - The type X'81' record is written when the system is put in a non-restartable wait by the operating system following a machine check.

  - The type X'84' record indicates a restartable wait state requiring operator intervention. Examples of this condition are hot I/O and intervention required on a paging pack.

The recovery/ termination record contains information relating to the cause of termination and system environmental information.

If the record is documenting normal termination, it consists only of the 24-byte header. In a record written for abnormal termination, the header is followed by fields of variable length containing data relevant to the system termination or wait state codes.

Figure 10-10 on page 10-24 shows the format of the recovery/termination record.

Offset Dec(Hex)	Size (bytes) Alignment (bits)	Field Name	Description
0(0)	1	EODKEY1	Class/Source:
	1...		EOD Record; type = X'80'.
	1... ...1		MCH-forced termination (non-restartable wait); type = X'81'.
	1... .1..		IOS-forced termination (restartable wait); type = X'84'.
1(1)	1	EODKEY2	System/release level:
	xxx.		OS/360.
	001.		DOS (VSE systems).
	xxx.		OS/VS1.
	xxx.		CP67, VM/370, and later VM systems.
	100.		OS/VS2 and later MVS systems.
	...x		Reserved.
	 xxxx		Release level (0-1F).
2(2)	1	EODSMS	Record-independent switches:
	1...		More records follow.
	0...		Last record.
	.x..		Time-of-Day clock instruction issued (0 = IBM System/360, 1 = IBM System/370). Used in conjunction with date and time values at displacements 8 and 12.
	..1.		Record truncated.
	...1		370-XA mode record.
	.... 1...		TIME macro used (MVS).
	.... 0...		Time in timer units (VSE).
	xxx		Reserved.
3(3)	1		Record-dependent switches; not used for this record
4(4)	1	EODDEV	MDR device class of failing device, if any.
5(5)	1		Incremental release number (alphameric) of operating system.
6(6)	2		Reserved.
8(8)	8	EODDT	System date and time, as:
8(8)	4	EODDATE	System date of failure.
12(C)	4	EODTIME	System time of failure.
16(10)	8	EODCPUID	CPU identification.
16(10)	1	EODVER	Machine version code:
	xxxx xxx.		Reserved.
	0		Version I CPUs.
	1		Version II CPUs.
17(11)	3	EODSER	CPU serial number.
20(14)	2	EODMOD	CPU machine model number (3033, 4341, etc.).
22(16)	2	EODCEL	Maximum length of machine- (CPU-) dependent machine-check extended logout area.
			END OF STANDARD HEADER
24(18)	Variable		Information for abnormal termination records only. (See note.)
24(18)	4		Length, plus 8, of data field.
28(1C)	4		Wait state code.
32(20)	Variable		Data.
			Note: If the wait state code is 66, 67, 68, 69 or 6A, MVS hot I/O recovery processing writes this termination record. A 32-byte data field contains the SCD entry for the channel with the "hot" condition (see the MVS or MVS/XA Debugging Handbook for a detailed description of the SCD). For other wait state codes that use these fields, the length of the data field varies.

Figure 10-10. Recovery/Termination Record Format

System Initialization (IPL) Record

Both OS/VS and VSE write IPL records to document operating system initialization. MVS, MVS/XA and VSE/Advanced Functions always write the record, while other OS/VS and VSE systems do so only if the RDE (reliability data extractor) option is in effect.

Under MVS, an IPL record is also generated to provide information about power-line disturbances that cause system termination.

If the RDE option is in effect, the operator is prompted by a message (IFB010D, in MVS) to supply a code for the reason for the IPL and for the subsystem responsible for the IPL. These are then included in the record when it is written to the ERDS.

Figure 10-11 shows the format of the IPL record itself; Figure 10-12 on page 10-27 shows the reason codes that can be associated with IPLs, and Figure 10-13 on page 10-27 shows the codes for the responsible subsystems. A subsystem code is meaningful only when the reason code indicates that an IBM product was responsible for the IPL. See *SPL: SYS1.LOGREC Error Recording* for MVS or MVS/XA and *VSE/Advanced Functions System Management Guide.*

Offset Dec(Hex)	Size (bytes) Alignment (bits)	Field Name	Description
0(0)	1	IPLKEY1	Class/Source:
	.1.1		IPL Record; type = X'50'
1(1)	1	IPLKEY2	System/release level:
	000.		OS/360.
	001.		DOS (VSE systems).
	010.		OS/VS1.
	011.		CP67, VM/370, and later VM systems.
	100.		OS/VS2 and later MVS systems.
	...x		Reserved.
	 xxxx		Release level (0-1F).
2(2)	1	IPLSMS	Record-independent switches:
	1...		More records follow.
	0...		Last record.
	.x..		Time-of-Day clock instruction issued (0 = IBM System/360, 1 = IBM System/370). Used in conjunction with date and time values at displacements 8 and 12.
	..1.		Record truncated.
	...1		370-XA mode record.
	.... 1...		TIME macro used (OS/VS).
	.... 0...		Time in timer units (VSE).
	xxx		Reserved.
3(3)	2		Reserved.
5(5)	1		Incremental release number (alphameric) of operating system.
6(6)	2		Reserved.
8(8)	8	IPLDT	System date and time when record was created:
8(8)	4	IPLDATE	System date of failure.
12(C)	4	IPLTIME	System time of failure.
16(C)	8	IPLCPUID	CPU identification:
16(10)	1	IPLVER	Machine version code:
	xxxx xxx.		Reserved.
	0		Version I CPUs.
	1		Version II CPUs.
17(11)	3	IPLSER	CPU serial number.
20(14)	2	IPLMOD	CPU machine model number (3033, 4341, etc.).
22(16)	2	IPLCEL	Maximum length of machine- (CPU-) dependent machine-check extended logout area.
			END OF STANDARD HEADER
24(18)	1	IPLSYSID	Device type or program that caused restart. See Figure 10-13.
25(19)	3		Not used for IPL record.
28(1C)	2	IPLREAS	Alphameric code representing the reason for IPL. See Figure 10-12.
30(1E)	2	IPLCHNM	Channel map.
32(20)	8	IPLCHAN	Channel type assignment at IPL. See the CCH record for the channel type codes. *Note:* The channel assignment table is not used for 370-XA.
40(28)	4	IPLHADDR	Address of the last valid byte of storage found at IPL time. (MVS only.)
44(2C)	4	IPLSDATE	Time-stamp date. (MVS only.)
48(30)	4	IPLSTIME	Time-stamp time. (MVS only.)

Figure 10-11. System Initialization (IPL) Record Format

IPL Reason Codes

Code	Reason	Description
NM	Normal	Normal system initialization.
IE	IBM hardware/programming problem, CE/PSR not required	System restarted after a stop caused by a hardware failure or IBM programming problem, but a CE/PSR was not required.
IM	IBM hardware/programming problem, CE/PSR required	System restarted after a stop caused by a hardware failure or IBM programming problem, and it was necessary for a CE/PSR to correct problem.
ME	Media	An IBM hardware unit failed because of faulty or damaged media (such as a damaged tape or disk).
UN	Unknown	An undetermined hardware or software failure.
OP	Operational	An operator error or procedural problem.
UP	User program	A program other than an IBM-supplied system control program or programming product failed in such a way as to cause a system restart.
EN	Environmental	A failure other than hardware/software or operational caused system to be restarted (power failure, air conditioning, etc.).
CE	CE/PSR has system	System restarted at CE/PSR request to correct problem.
DF	Default	Operator replied "U" or entered a null line in response to message IFB010D.

Figure 10-12. IPL Reason Codes

IPL Subsystem ID Codes

ID	Subsystem Name	Description/Components
00	Null	Subsystem is unknown or subsystem code is not required by reason code.
10	Processor	CPU, channels, storage units, operator consoles.
20	Direct Access (DASD)	Direct access storage devices and their control units.
30	Other	All devices other than those included under other subsystem IDs.
40	Tape	Magnetic tape devices and their control units.
50	Card/Print	Card (unit record) and printing devices.
60	MICR/OCR	Magnetic ink (MICR) and optical (OCR) character recognition devices.
70	Teleprocessing	Teleprocessing devices and their control units.
80	Graphics/Display/Audio	Graphic, display, and audio devices.
90	IBM System Control Program	IBM programming system.
91	IBM Programming Product	IBM programming products such as FORTRAN, COBOL, or RPG.

Figure 10-13. Subsystem ID Codes for IPL

Machine Check (MCH) Record

All three operating systems write MCH records to document the occurrence of processor, storage, storage key or timing facility (external damage) failures.

The records are written in response to a machine check interrupt, which can be for one of three reasons:

1. The problem was recovered by the hardware or (for MVS systems only) recovered by the software.

2. The problem was not corrected by hardware. For MVS, a hard machine check is one that could not be corrected or circumvented, so the software recovery routines were given control for the task.

3. The problem resulted in the loss of a processor.

Figure 10-14 shows the MCH record format for systems other than MVS. The MVS systems write an extended form of the MCH record; Figure 10-15 on page 10-31 shows the format of the MVS version of the record.

Non-MVS MCH Record Format

Offset Dec(Hex)	Size (bytes) Alignment (bits)	Field Name	Description
0(0)	1	MCHKEY1	Class/Source:
	...1		MCH record; type = X'10'.
	...1 ...1		Generated by diagnostic program = X'11'.
	...1 .1.1		Generated by cycle trace (0168 only) = X'14'.
1(1)	1	MCHKEY2	System/release level:
	000.		OS/360.
	001.		DOS (VSE systems).
	010.		OS/VS1.
	011.		CP67, VM/370, and later VM systems.
	100.		OS/VS2 and later MVS systems.
	...x		Reserved.
	 xxxx		Release level (0-1F).
2(2)	1	MCHSMS	Record-independent switches:
	1...		More records follow.
	0...		Last record.
	.x..		Time-of-Day clock instruction issued (0 = IBM System/360, 1 = IBM System/370). Used in conjunction with date and time values at displacements 8 and 12.
	..1.		Record truncated.
	...1		370-XA mode record.
	.... 1...		TIME macro used (MVS).
	.... 0...		Time in timer units (VSE).
	.... .1..	MCHPASS	Passed, recovery status. VM passed error to guest.
	xx		Reserved.
3(3)	1	MCHSW2	Record-dependent switches:
	1...	MCHSYSTR	Short form of record; only MCIC portion of model-independent logout.
	.x..		Reserved.
	..1.	MCHSYSTM	System terminated by MCH.
	.... x...		Reserved.
		MCHSFTSW	Recovery status bits. See "3090 Processor" on page 20-7.
	.... .00.		Invalid or, for VM, passed to guest.
	.... .01.		Degrade fail.
	.... .10.		Soft fail.
	.... .11.		Hard fail.
	x		Reserved.
4(4)	2		Reserved.
6(6)	1	MCHRCDCT	Record count:
	xxxx		Sequence number of this physical record.
	 xxxx		Total number of physical records in this logical record.
7(7)	1		Reserved.
8(8)	8	MCHDT	System date and time of failure:
8(8)	4	xxxDATE	System date of failure.
12(C)	4	xxxTIME	System time of failure.
16(10)	8	MCHCPUID	CPU identification:
16(10)	1	MCHVER	Machine version code:
	xxxx xxx.		Reserved.
	0		Version I CPUs.
	1		Version II CPUs.
17(11)	3	MCHSER	CPU serial number.
20(14)	2	MCHMOD	CPU machine model number (3033, 4341, etc.).
22(16)	2	MCHCEL	Maximum length of machine- (CPU-) dependent machine-check extended logout area. *Note: This field is meaningful only for records created in S/370 operating mode.*
			END OF STANDARD HEADER
24(18)	8	MCHPGMID	Program (module) being processed at the time of failure.
32(20)	8	MCHJOBID	Job executing at time of failure.
40(28)	8	MCHOPSW	Machine check old PSW from storage locations 48-55.
48(30)	280	MCHILOG	Machine-independent logout. Data from storage locations 232 - 511:

Figure 10-14 (Part 1 of 2). Machine Check (MCH) Record Format - Non-MVS

Offset Dec(Hex)	Size (bytes) Alignment (bits)	Field Name	Description
48(30)	8	MCHMCIC	Machine-check interrupt code (from storage locations 232 - 239) as stored by hardware routines at time of machine check.
48(30)	Byte 0	MCHMCIC0	
	1...	MCHMFSD	System damage (SD).
	.1..	MCHMFPD	Processing damage (PD).
	..1.	MCHMFSR	System recovery (SR).
	...1	MCHMFTD	Timer damage, for S/370 (TD). Unassigned, for 370-XA.
	.... 1...	MCHMFCD	Clock damage (CD).
	.... .1..	MCHMFED	External damage, for S/370 (ED).
	.... ..x.		Reserved.
	1	MCHMFDG	Degradation (DG).
49(31)	Byte 1	MCHMCIC1	
	1...	MCHMFWN	Power warning (W).
	.1..	MCHMPCRW	Pending CRW report, for 370-XA. Unassigned, for S/370 (CP).
	..1.		Service processor damage.
	...1	MCHMCHSD	Channel subsystem damage, for 370-XA (CK). Unassigned, for S/370.
	.... xx..		Reserved.
	.... ..1.	MCHMIBU	Backed-up indicator (B).
	1	MCHMIDY	Delayed, for S/370 (D). Unassigned, for 370-XA.
50(32)	Byte 2	MCHMCIC2	
	1...	MCHMFSE	Storage error (SE).
	.1..	MCHMFSC	Storage error corrected (SC).
	..1.	MCHMFKE	Key error, for S/370. Storage key error, uncorrected, for 370-XA (KE).
	...1	MCHMDFDS	Double-bit storage error (DS).
	.... 1...	MCHMVWP	PSW EMWP is valid (WP).
	.... .1..	MCHMVMS	PSW masks and key are valid (MS).
	.... ..1.	MCHMVPM	Program masks and condition code are valid (PM).
	1	MCHMVIA	Instruction address is valid (IA).
51(33)	Byte3	MCHMCIC3	
	1...	MCHMVFA	Failing storage address is valid (FA).
	.1..	MCHMVRC	Region code is valid (RC).
	..1.	MCHMVEC	External damage code is valid (EC).
	...1	MCHMVFP	Floating point register is valid (FP).
	.... 1...	MCHMVGR	General purpose register is valid (GP).
	.... .1..	MCHMVCR	Control register is valid (CR).
	.... ..1.	MCHMVLG	Logout (MCEL) is valid (LG).
	1	MCHMVST	Storage logical is valid (ST).
52(34)	Byte4	MCHMCIC4	Reserved.
53(35)	Byte5	MCHMCIC5	
	xxxx xx..		Reserved.
	.... ..1.	MCHMVPT	Processor timer is valid.
	1	MCHMVCC	Clock comparator is valid.
54(36)	2	MCHMCIC6	Actual length of MCEL data stored for this machine-check interrupt.
56(38)	8	MCHS240	Data from storage locations 240-247.
64(40)	4	MCHMFSA	Failing storage address from storage location 248.
68(44)	260	MCHS252	Data from storage locations 252-511.
328(148)	Variable	MCHMCEL	Model-dependent machine-check extended logout area. Maximum length is in MCHCEL field at offset 22 decimal; minimum length is zero. Contains model-dependent logout information. Size is machine-dependent. *Note:* No MCEL for 4331, 4341, 3081, or 3090. **Model** — **Maximum Length** 135 — 0 bytes 145 — 192 bytes 155II/158 — 672 bytes 165II/168 — 1416 bytes 3031 — 772 bytes 3032 — 1416 bytes 3033 — 1224 bytes 43XX — 0 bytes 308X — 0 bytes 3090 — 0 bytes
Variable	Variable		Damage assessment: model- and SCP-dependent field identifying the extent of damage found by a recovery management program. (OS/VS1 only.)

Figure 10-14 (Part 2 of 2). Machine Check (MCH) Record Format - Non-MVS

MVS MCH Record Format

Offset Dec(Hex)	Size (bytes) Alignment (bits)	Field Name	Description
0(0)	1	MCHKEY1	Class/Source:
	...1		MCH record; type = X'10'.
	...1 ..11		MCH Record for MVS; type = X'13'.
1(1)	1	MCHKEY2	System/release level:
	000.		OS/360.
	001.		DOS (VSE systems).
	010.		OS/VS1.
	011.		CP67, VM/370, and later VM systems.
	100.		OS/VS2 and later MVS systems.
	...x		Reserved.
	 xxxx		Release level (0-1F).
2(2)	1	MCHSMS	Record-independent switches:
	1...		More records follow.
	0...		Last record.
	.x..		Time-of-Day clock instruction issued (0 = IBM System/360, 1 = IBM System/370). Used in conjunction with date and time values at displacements 8 and 12.
	..1.		Record truncated.
	...1		370-XA mode record.
	.... 1...		TIME macro used (MVS).
	.... 0...		Time in timer units (VSE).
	.... .1..	MCHPASS	Passed, recovery status. VM passed error to guest.
	xx		Reserved.
3(3)	1	MCHSW2	Record-dependent switches:
	1...	MCHSYSTR	System terminated by MCH.
	.x..		Reserved.
	..1.	MCHSYSTM	System terminated by MCH.
	...1		Record contains an ERRORID.
	.... x...		Reserved.
		MCHSFTSW	Recovery status bits. See "3090 Processor" on page 20-7.
	.... .00.		Invalid or, for VM, passed to guest.
	.... .01.		Degrade fail.
	.... .10.		Soft fail.
	.... .11.		Hard fail.
	x		Reserved.
4(4)	2		Reserved.
	 xxxx		Reserved.
4(4)	2		Reserved.
6(6)	1	MCHRCDCT	Record count:
	xxxx		Sequence number of this physical record.
	 xxxx		Total number of physical records in this logical record.
7(7)	1		Reserved.
8(8)	8	MCHDT	System date and time:
8(8)	4	MCHDATE	System date of failure.
12(C)	4	MCHTIME	System time of failure.
16(10)	8	MCHCPUID	CPU identification:
16(10)	1	MCHVER	Machine version code:
	xxxx xxx.		Reserved.
	0		Version I CPUs.
	1		Version II CPUs.
17(11)	3	MCHSER	CPU serial number.
20(14)	2	MCHMOD	CPU machine model number (3033, 4341, etc.).
22(16)	2	MCHCEL	Maximum length of machine- (CPU-) dependent machine-check extended logout area. *Note: This field is meaningful only for records created in S/370 operating mode.*
			END OF STANDARD HEADER
24(18)	4	MCHLNG	Length of this record, in hexadecimal.
28(1C)	4	MCHWSC	Wait state code.

Figure 10-15 (Part 1 of 4). Machine Check (MCH) Record Format - MVS

Offset Dec(Hex)	Size (bytes) Alignment (bits)	Field Name	Description
32(20)	4	MCHEIA	MCH error indicator area; flags reflecting the cause and severity of the failure. See mapping macro IHALRB, in the MVS or MVS/XA *Debugging Handbook*.
32(20)	Byte 0	MCHTERM	Terminating error flags:
	xxx. ..x.		Reserved
	...1	MCHTSEC	Secondary error.
	.... 1...	MCHTCKS	Check stop.
	.... .1..	MCHTWRN	Power warning.
	.... ..1.	MCHTDMG	System damage.
33(21)	Byte 1	MCHHARD	Hard machine error flags:
	1...	MCHHHRD	Hard error assumed.
	.xxx		Reserved.
	.... 1...	MCHHRPI	Register or PSW invalid.
	.... .1..	MCHHSTO	Hard storage error.
	.... ..1.	MCHHSPF	Hard storage protection key error.
	1	MCHHIPD	Instruction processing damage.
34(22)	Byte 2	MCHINTM	Intermediate error flags:
	xxxx		Reserved.
	.... 1...	MCHITOD	TOD clock error.
	.... .1..	MCHICKC	Clock comparator error.
	.... ..1.	MCHICTM	CPU timer error.
	1	MCHIL80	Interval timer error.
35(23)	Byte3	MCHSOFT	Soft machine error flags:
	1...	MCHSSFT	Soft error assumed.
	.xxx		Reserved.
	.... 1...	MCHSEXD	External damage.
	.... .1..	MCHSECC	ECC-corrected storage error.
	.... ..1.	MCHSHIR	HIR-corrected processor (CPU) error.
	1	MCHSBUF	Buffer error.
36(24)	3	MCHPDAR	PDAR (program damage assessment and repair) data.
	Byte 0	MCHPRTM	Software status supplied by RTM:
	xxx.		Reserved.
	...1	MCHINVP	Storage reconfigured; page invalidated.
	.... 1...	MCHRSRC	Storage reconfiguration status available at offset 37.
	.... .1..	MCHRSRF	Storage reconfiguration not attempted.
	xx		Reserved.
37(25)	2	MCHRSRS	Storage reconfiguration status:
	Byte 1	MCHRSR1	
	xxxx xx..		Reserved.
	.... ..1.	MCHSER	Storage error was already set in frame.
	1	MCHCHNG	Frame had change indicator on.
	Byte 2	MCHRSR2	
	1...	MCHOFLN	Frame offline or scheduled to go offline.
	.1..	MCHINTC	Intercept; frame scheduled to go offline, has a permanent storage error or scheduled for V=R status.
	..1.	MCHSPER	Permanent error occurred in frame.
	...1	MCHNUCL	Frame contains permanently resident system storage.
	.... 1...	MCHFSQA	Frame is in use for SQA.
	.... .1..	MCHLSQA	Frame is in use for LSQA.
	.... ..1.	MCHPGFX	Frame contains page-fixed data.
	1	MCHVEQR	Frame is in use for V=R or scheduled for V=R.
39(27)	1	MCHPWL	Physical word length (length of checking block used by machine model).
40(28)	8	MCHOPSW	Machine-check old PSW from storage locations 48-55.

Figure 10-15 (Part 2 of 4). Machine Check (MCH) Record Format - MVS

Offset Dec(Hex)	Size (bytes) Alignment (bits)	Field Name	Description
48(30)	280		Machine-independent logout. Data from storage locations 232 - 511:
48(30)	8	MCHMCIC	Machine-check interrupt code (from storage locations 232 - 239) as stored by hardware routines at time of machine check:
	Byte 0	MCHMCIC0	
	1...	MCHMFSD	System damage (SD).
	.1..	MCHMFPD	Processing damage (PD).
	..1.	MCHMFSR	System recovery (SR).
	...1	MCHMFTD	Timer damage, for S/370 (TD). Unused, for 370-XA.
	.... 1...	MCHMFCD	Clock damage (CD).
	.... .1..	MCHMFED	External damage, for S/370 (ED). Unused, for 370-XA.
	.... ..x.		Reserved.
	1	MCHMFDG	Degradation (DG).
49(31)	Byte 1	MCHMCIC1	
	1...	MCHMFWN	Power warning (W).
	.1..	MCHMPCRW	Pending CRW report, for 370-XA (CP). Unused, for S/370.
	..1.		Service processor damage.
	...1	MCHMCHSD	Channel subsystem damage, for 370-XA (CK). Unused, for S/370.
	.... xx..		Reserved.
	.... ..1.	MCHMIBU	Backed-up indicator (B).
	1	MCHMIDY	Delayed, for S/370. Unused, for 370-XA (D).
50(32)	Byte 2	MCHMCIC2	
	1...	MCHMFSE	Storage error (SE).
	.1..	MCHMFSC	Storage error corrected (SC).
	..1.	MCHMFKE	Key error, for S/370. Storage key error, uncorrected, for 370-XA (KE).
	...x		Reserved.
	.... 1...	MCHMVWP	PSW EMWP is valid (WP).
	.... .1..	MCHMVMS	PSW masks and key are valid (MS).
	.... ..1.	MCHMVPM	Program masks and condition code are valid (PM).
	1	MCHMVIA	Instruction address is valid (IA).
51(33)	Byte 3	MCHMCIC3	
	1...	MCHMVFA	Failing storage address is valid (FA).
	.1..	MCHMVRC	Region code is valid (RC).
	..1.	MCHMVDS	Storage degradation (DS).
	...1	MCHMVFP	Floating point register is valid (FP).
	.... 1...	MCHMVGR	General purpose register is valid (GP).
	.... .1..	MCHMVCR	Control register is valid (CR).
	.... ..1.	MCHMVLG	Logout (MCEL) is valid (LG).
	1	MCHMVST	Storage logical is valid (ST).
52(34)	Byte 4	MCHMCIC4	
	1...	MCHMNVF	MCH may not be valid. Set if PSAMCHIC and PSAMCHFL = 0.
	.xxx xxxx		Reserved.
53(35)	Byte5	MCHMCIC5	
	xxxx xx..		Reserved.
	.... ..1.	MCHMVPT	Processor timer is valid.
	1	MCHMVCC	Clock comparator is valid.
54(36)	Bytes 6 and 7	MCHMCIC6	Actual length of MCEL data stored for this machine-check interrupt.
56(38)	8	MCHS240	Data from storage locations 240-247.
64(40)	4	MCHMFSA	Failing storage address from storage location 248.
68(44)	260	MCHS252	Data from storage locations 252-511.

Figure 10-15 (Part 3 of 4). Machine Check (MCH) Record Format - MVS

Offset Dec(Hex)	Size (bytes) Alignment (bits)	Field Name	Description
328(148)	Variable	MCHMCEL	Model-dependent machine check extended logout area. Maximum length is in MCHCEL field; minimum length is zero. Contains model-dependent logout information. Size is machine-dependent. *Note:* No MCEL for 4331, 4341 or 3081. **Model** **Maximum Length** 135 0 bytes 145 192 bytes 155II/158 672 bytes 165II/168 1416 bytes 3031 772 bytes 3032 1416 bytes 3033 1224 bytes 4331 0 bytes 4341 0 bytes 3081 0 bytes
Variable	10	ERRORID	RTM-generated error identifier, consisting of: 2-byte sequence number 2-byte CPU identifier 2-byte ASID 4-byte time stamp If present, the error ID is the last 10 bytes of the record.

Figure 10-15 (Part 4 of 4). Machine Check (MCH) Record Format - MVS

Miscellaneous Data (MDR) Record

MDR records can contain error and usage data from buffered control units or communications controllers, or they can document device failures on teleprocessing (TP) devices connected to a communications controller.

Some of the events that can cause MDR recording are:

- Overflow of the statistical counters in a buffered control unit
- Overflow of the NCP counter in a communications controller
- TP device failure
- DASD volume demounts
- Operator-initiated EOD (End of Day) or ROD (Record on Demand), or VARY OFFLINE commands
- In an MVS system, some invocations of EREP, which force the writing of statistical data to LOGREC.

Figure 10-16 shows the format of the MDR record.

Offset Dec(Hex)	Size (bytes) Alignment (bits)	Field Name	Description
0(0)	1	MDRKEY1	Class/Source:
	1..1		MDR record formatted by SVC 91; type = X'90'.
	1..1 ...1		MDR record; type = X'91'.
1(1)	1	MDRKEY2	System/release level:
	000.		OS/360.
	001.		DOS (VSE systems).
	010.		OS/VS1.
	011.		CP67, VM/370, and later VM systems.
	100.		OS/VS2 and later MVS systems.
	...x		Reserved.
	 xxxx		Release level (0-1F).
2(2)	1	MDRSMS	Record-independent switches:
	1...		More records follow.
	0...		Last record.
	.x..		Time-of-Day clock instruction issued (0 = IBM System/360, 1 = IBM System/370). Used in conjunction with date and time values at displacements 8 and 12.
	..1.		Record truncated.
	...1		370-XA mode record.
	.... 1...		TIME macro used (MVS).
	.... 0...		Time in timer units (VSE).
	xxx		Reserved.
3(3)	3		Record-dependent switches:
	Byte 0	MDRSW2	
	x...		Reserved.
	.1..		Record incomplete.
	..xx xxxx		Reserved.
4(4)	Byte 1	MDRDEV	Device code:
	x.xx xxx.		See "Device Type Codes in the MDR Record" on page 12-16 for all the codes currently supported by EREP.
	.x.. ...x		Reserved
5(5)	Byte 2	MDRBUFC	Channel set ID, for S/370.
	1...		Sub-ID field in this record is of variable length.
	.xxx		Reserved.
	 xxxx		Number of characters (in hexadecimal) in sub-ID field of device identified at offset 26.
6(6)	1	MDRRCDCT	Record count:
	xxxx		Sequence number of this physical record.
	 xxxx		Total number of physical records in this logical record.
7(7)	1	MDRCHPID	Channel path ID, for 370-XA.
8(8)	8	MDRDT	System date and time of incident:
8(8)	4	MDRDATE	System date of failure.
12(C)	4	MDRTIME	System time of failure.
16(10)	8	MDRCPUID	CPU identification:
16(10)	1	MDRVER	Machine version code:
	xxxx xxx.		Reserved.
	0		Version I CPUs.
	1		Version II CPUs.
17(11)	3	MDRSER	CPU serial number.
20(14)	2	MDRMOD	CPU machine model number (3033, 4341, etc.).
22(16)	2	MDRCEL	Maximum length of machine- (CPU-) dependent machine-check extended logout area.
			END OF STANDARD HEADER
24(18)	2	MDRCUA	Device address of data identified in this record. (Primary CUA, for S/370. Device number, for 370-XA).
26(1A)	Variable	MDRSUBID	Field (2-15 bytes) that identifies the device whose address is at offset 24. The length of this field is at offset 5. *Note:* Depending on the device, this field can be the volume serial number (VOLID) or CUA/device number (SCUA) of the unit.
	Variable	MDRINFO	Device-dependent information supplied by the ERP that detected the error.
Variable	2	MDRRCTWD	Flag bytes from the record control table (RCT) used to create this record. (MVS and MVS/XA only.)
	Variable		More device-dependent information.

Figure 10-16. Miscellaneous Data (MDR) Record Format

Missing Interrupt Handler (MIH) Record

MIH records are produced by MVS/370 and MVS/XA for missing channel-end (primary status) and/or device-end (secondary status) interrupts on non-TP devices. The records use fields from the UCB to define the origin and status of the missing interrupt.

The VSE/Advanced Function system also produces MIH records. See the appropriate licensed program manual for the VSE/Advanced Functions MIH record format.

Figure 10-17 shows the format of the MIH record produced by MVS/370; Figure 10-18 on page 10-39 shows the format of the MIH record produced by MVS/XA.

370-Mode MIH Record Format

Offset Dec(Hex)	Size (bytes) Alignment (bits)	Field Name	Description
0(0)	1	MIHKEY1	Class/Source:
	.111		MIH record, S/370 mode; type = X'70'.
1(1)	1	MIHKEY2	System/release level:
	000.		OS/360.
	001.		DOS (VSE systems).
	010.		OS/VS1.
	011.		CP67, VM/370, and later VM systems.
	100.		OS/VS2 and later MVS systems.
	...x		Reserved.
	 xxxx		Release level (0-1F).
2(2)	1	MIHSMS	Record-independent switches:
	1...		More records follow.
	0...		Last record.
	.x..		Time-of-Day clock instruction issued (0 = IBM System/360, 1 = IBM System/370). Used in conjunction with date and time values at displacements 8 and 12.
	..1.		Record truncated.
	...1		370-XA mode record.
	.... 1...		TIME macro used (MVS).
	.... 0...		Time in timer units (VSE).
	xxx		Reserved.
3(3)	2		Record-dependent switches:
	Byte 0	MIHSW2	Reserved.
	Byte 1	MIHSW3	
	1...	MIHCEBIT	Channel-end interrupt found pending.
	.1..	MIHDEBIT	Device-end interrupt found pending.
	..1.		Device found idle with request queued.
	...x xxxx		Reserved.
5(5)	Byte 1	MIHCSID	Channel set ID for MVS.
6(6)	1	MIHRCDCT	Record count:
	xxxx		Sequence number of this physical record.
	 xxxx		Total number of physical records in this logical record.
7(7)	1		Reserved.
8(8)	8	MIHDT	System date and time, as:
8(8)	4	MIHDATE	System date of failure.
12(C)	4	MIHTIME	System time of failure.
16(10)	8	MIHCPUID	CPU identification, as:
16(10)	1	MIHVER	Machine version code:
	xxxx xxx.		Reserved.
	0		Version I CPUs.
	1		Version II CPUs.
17(11)	3	MIHSER	CPU serial number.
20(14)	2	MIHMOD	CPU machine model number (3033, 4341, etc.).
22(16)	2		Not used.
			END OF STANDARD HEADER
24(18)	8	MIHJOBID	Alphameric name assigned to job (as identified, for example, by a jobname on a JCL JOB statement) with an I/O request pending. If jobname cannot be determined, field is set to blanks.
32(20)	3	MIHCUA2	CUA used to address the device.
35(23)	3	MIHCUA1	Primary CUA of device.
38(26)	6	MIHVOL	VOLSER of volume on device associated with pending I/O request.
44(2C)	4	MIHDEV	Device class and type (from UCB) of unit associated with pending I/O request.
48(30)	8	MIHINT	Time interval (decimal) used by MIH to check for pending conditions.
56(38)	Variable		System-dependent data. For example, for MVS the fields contain:
56(38)	24		UCB common portion.
80(50)	Variable		UCB device-dependent data: 24 bytes for tape and DASD 16 bytes for graphics devices 8 bytes for all other devices.

Figure 10-17. MVS/370 Missing Interrupt Handler (MIH) Record Format

MVS/XA Missing Interrupt Handler (MIX) Record Format

Offset Dec(Hex)	Size (bytes) Alignment (bits)	Field Name	Description
0(0)	1	MIXKEY1	Class/Source:
	.111 ...1		S/370-XA MIH record; type = X'71'.
1(1)	1	MIXKEY2	System/release level:
	000.		OS/360.
	001.		DOS (VSE systems).
	010.		OS/VS1.
	011.		CP67, VM/370, and later VM systems.
	100.		OS/VS2 and later MVS systems.
	...x		Reserved.
	 xxxx		Release level (0-1F).
2(2)	1	MIXSMS	Record-independent switches:
	1...		More records follow.
	0...		Last record.
	.x..		Time-of-Day clock instruction issued (0 = IBM System/360, 1 = IBM System/370). Used in conjunction with date and time values at displacements 8 and 12.
	..1.		Record truncated.
	...1		370-XA mode record.
	.... 1...		TIME macro used (MVS).
	.... 0...		Time in timer units (VSE).
	xxx		Reserved.
3(3)	1	MIXSW2	Reserved.
5(5)	1	MIXCSID	Channel set ID for MVS.
6(6)	1	MIXRCDCT	Record count:
	xxxx		Sequence number of this physical record.
	 xxxx		Total number of physical records in this logical record.
7(7)	1		Reserved.
8(8)	8	MIXDT	System date and time, as:
8(8)	4	MIXDATE	System date of failure.
12(C)	4	MIXTIME	System time of failure.
16(10)	8	MIXCPUID	CPU identification, as:
16(10)	1	MIXVER	Machine version code:
	xxxx xxx.		Reserved.
	0		Version I CPUs.
	1		Version II CPUs.
17(11)	3	MIXSER	CPU serial number.
20(14)	2	MIXMOD	CPU machine model number (3033, 4341, etc.).
22(16)	2		Not used.
			END OF STANDARD HEADER
24(18)	8	MIXJOBID	Jobname from ASID.
32(20)	52	MIXSCHIB	Subchannel information block:
32(20)	4	MIXPMCW0	Interrupt parameter.
36(24)	4	MIXPMCW1	Path-management control word 1.
40(28)	1	MIXLPM	Logical path mask.
41(29)	1	MIXPNOM	Path-not-operational mask.
42(2A)	1	MIXLPUM	Last-path-used mask.
43(2B)	1	MIXPIM	Path-installed mask.
44(2C)	2	MIXMBI	Measurement block index.
46(2E)	1	MIXPOM	Path-operational mask.
47(2F)	1	MIXPAM	Path-available mask.
48(30)	8	MIXCHPID	CHPIDs 0-7.
56(38)	4	MIXPMCW6	Path-management control word 6.
60(3C)	12	MIXSCSW	Subchannel status words.
72(48)	12	MIXMDEP	Model-dependent area.
84(54)	8	MIXINTVL	Time interval used for detection.

Figure 10-18 (Part 1 of 2). MVS/XA Missing Interrupt Handler (MIX) Record Format

Offset Dec(Hex)	Size (bytes) Alignment (bits)	Field Name	Description
92(5C)	1	MIXITYPE	Type of missing interrupt (MIH code):
	1...		Missing CSCH interrupt.
	.1..		Missing HSCH interrupt.
	..1.		Device found idle with request queued.
	...1		Start pending in subchannel.
	.... x...		Reserved.
	.... .1..		Mount pending in subchannel.
	.... ..1.		Missing primary status.
	1		Missing secondary status.
93(5D)	3		Missing interrupt handler actions:
93(5D)	1	MIXDEFLT	Default actions to attempt.
94(5E)	1	MIXATMPT	Actions to be attempted.
95(5F)	1	MIXTRIED	Actions actually tried.
	1...		Halt or clear subchannel.
	.1..		Simulated interrupt.
	..1.		Re-drive device.
	...1		Requeue I/O request.
	.... 1...		Issue message.
	.... .1..		Log the condition.
	xx		Reserved.
96(60)	4	MIXSID	Subchannel ID number.
100(64)	2	MIXPMCW	Path-management control word from UCBPMCW1.
102(66)	1	MIXULPM	Logical-path mask from UCBLPM.
103(67)	1	MIXULPUM	Last-Path-Used mask from UCBLPUM.
104(68)	1	MIXUPIM	UCBPIM.
105(69)	8	MIXUCPID	CHPIDs from UCBCHPID.
113(71)	1	MIXUCBLV	UCB level byte.
114(72)	1	MIXIOSFG	IOS Flags.
115(73)	4	MIXLVMSK	Level mask from UCBLVMSK.
119(77)	1	MIXFLAGS	MIH flag proc (UCBMIHTI).
120(78)	1	MIXFLAG1	Flag byte.
	1...		UCBALTCU.
	.xxx xxxx		Reserved.
121(79)	1	MIXFLAG2	Flag byte from UCBFLC.
122(7A)	2	MIXUCHAN	Device number from UCBCHAN.
124(7C)	2	MIXFLAG3	Flag bytes from UCBSFLS.
126(7E)	4	MIXDEVTY	UCB device class/type.
130(82)	6	MIXVOLUM	Volume serial number.
136(88)	1	MIXFLAG4	Flag byte.
	1...		UCBMOUNT.
	.xxx xxxx		Reserved.
137(89)	1	MIXFLAG5	Flag byte from UCBFL4 (DASD only).
138(8A)	1	MIXFLG1	MIH record flags:
	1...	MIXADDL1	Additional-data flag.
	.xxx xxxx		Reserved.
139(8B)	1	MIXFLG2	Reserved.
140(8C)	1	MIXRSNC	Reason code associated with MIXITYPE (offset 92 decimal).
141(8D)	3		Reserved.
144(90)	1	MIXHLTRC	Halt-request return code from IOSVHSCH.
145(91)	1	MIXCLRRC	Clear-request return code from IOSVHSCH.
146(92)	1	MIXSTRC1	Store-subchannel-request retrun code from IOSVHSCH.
147(93)	1	MIXSTRC2	Store-subchannel-request retrun code from IOSVHSCH.
148(94)	4	MIXCIRB1	CSCH IRB word 1.
152(98)	4	MIXCIRB2	STSCH IRB word 1.
156(9C)	8		Reserved.

Figure 10-18 (Part 2 of 2). MVS/XA Missing Interrupt Handler (MIX) Record Format

Outboard (OBR) Record

OBR records document a variety of I/O errors and statistical data. They can take one of two forms, depending on why they are written.

The short form is used to record statistical data for the devices (except tape drives) whose statistical data counters are in "core" rather than in control-unit buffers (see "Miscellaneous Data (MDR) Record" on page 10-35). It is also written in response to the same operator-initiated and program-initiated actions that can trigger an MDR record. When statistical data is written to the ERDS before EREP begins to retrieve records for a report, the data is in short OBR records or MDR records, depending on the devices involved. (For tape devices, statistical data is in long OBR records.)

The long form of the OBR record includes, in addition to the information contained in the short form, documentation of permanent unit checks (I/O errors that the system's error recovery program could not correct). The long form also documents some temporary unit checks and statistical data for devices with in-core counters.

MVS and MVS/XA write a long OBR record when the dynamic pathing availability facility encounters an error while changing the state of a path group.

For teleprocessing (TP) devices and controllers, the device-dependent data in the OBR record is detailed in an IBM International Systems Center Bulletin entitled *Network Error Data.*

Figure 10-19 on page 10-42 shows the format of short OBR records; Figure 10-20 on page 10-44 shows the format of long OBR records.

Short OBR Record Format

Offset Dec(Hex)	Size (bytes) Alignment (bits)	Field Name	Description
0(0)	1	OBRKEY1	Class/Source:
	..11		OBR (unit check) record; type = X'30'.
	..11 .1..		TP access method (TCAM (VS)/BTAM (VSE)) OBR record; type = X'34'.
	..11 .11.		TP access method (VTAM) OBR record; type = X'36'.
	xx.. x..x		Not used for short OBR record.
1(1)	1	OBRKEY2	System/release level:
	000.		OS/360.
	001.		DOS (VSE systems).
	010.		OS/VS1.
	011.		CP67, VM/370, and later VM systems.
	100.		OS/VS2 and later MVS systems.
	...x		Reserved.
	 xxxx		Release level (0-1F).
2(2)	1	OBRSMS	Record-independent switches:
	1...		More records follow.
	0...		Last record.
	.x..		Time-of-Day clock instruction issued (0 = IBM System/360, 1 = IBM System/370). Used in conjunction with date and time values at displacements 8 and 12.
	..1.		Record truncated.
	...1		370-XA mode record.
	.... 1...		TIME macro used (MVS).
	.... 0...		Time in timer units (VSE).
	xxx		Reserved.
3(3)	3		Record-dependent switches:
	Byte 0	OBRSW2	
	1...		SDR counters dumped at EOD.
	.1..		Temporary switch, counter overflow.
	..1.		Short OBR record.
	...x x...		Not used.
	.... .1..		Volume demount.
	xx		Reserved.
	Byte 1	OBRXASW	
	1...		CHPID is invalid.
	.x..		Sense unavailable.
	..xx		Reserved.
	 xxxx		Model/version number; used with device type to indicate model or version.
	Byte 2	OBRCSID	Channel set ID, for S/370. Unused for 370-XA.
6(6)	1	OBRRCDCT	Record count:
	xxxx		Sequence number of this physical record.
	 xxxx		Total number of physical records in this logical record.
7(7)	1		Reserved.
8(8)	8	OBRDT	System date and time of incident:
8(8)	4	OBRDATE	System date of failure.
12(C)	4	OBRTIME	System time of failure.
16(10)	8	OBRCPUID	CPU identification.
16(10)	1	OBRVER	Machine version code:
	xxxx xxx.		Reserved.
	0		Version I CPUs.
	1		Version II CPUs.
17(11)	3	OBRSER	CPU serial number.
20(14)	2	OBRMOD	CPU machine model number (3033, 4341, etc.).
22(16)	2	OBRCEL	Maximum length of machine- (CPU-) dependent machine-check extended logout area.
			END OF STANDARD HEADER

Figure 10-19 (Part 1 of 2). Outboard (OBR) Record Format (Short Form)

Offset Dec(Hex)	Size (bytes) Alignment (bits)	Field Name	Description
24(18)	4	OBRCODE	Code for the device class and type of the failing device. First two bytes are VSE PUB bytes 4 and 5; second two bytes are MVS UCB device class and type bytes 18 and 19 (decimal). See "Device Type Codes in the OBR Record" on page 12-13 for all the codes currently supported by EREP.
28(1C)	1	OBRLSDRC	Number of bytes of statistical data recorded in the statistical data area at offset 32.
29(1D)	3	OBRPCUA	Primary CUA of the device associated with the failing I/O operation.
32(20)	Variable	OBRSDINF	Statistical data area, containing statistical counter/indicator data from the device statistics table.
32(20)	1	OBRSRDS	SDR counts:
	xxxx	OBRSDRS1	Temporary reads
	 xxxx	OBRSDRS2	Temporary writes
	2	SRCTWD	Flag bytes from the record control table (RCT) used to create this record. (MVS and MVS/XA only.)
	4	OBRSIOCT	The number of STARTIOs to the device since the last recording. (VSE only.)

Figure 10-19 (Part 2 of 2). Outboard (OBR) Record Format (Short Form)

Long OBR Record Format

Offset Dec(Hex)	Size (bytes) Alignment (bits)	Field Name	Description
0(0)	1	OBRKEY1	Class/Source:
	..11		OBR (unit check) record; type = X'30'.
	..11 .1..		TP access method (TCAM (VS)/BTAM (VSE)) OBR record; type = X'34'.
	..11 .11.		TP access method (VTAM) OBR record; type = X'36'.
	..11 1.1.		Dynamic Pathing Availability OBR record; type = X'3A'
1(1)	1	OBRKEY2	System/release level:
	000.		OS/360.
	001.		DOS (VSE systems).
	010.		OS/VS1.
	011.		CP67, VM/370, and later VM systems.
	100.		OS/VS2 and later MVS systems.
	...x		Reserved.
	 xxxx		Release level (0-1F).
2(2)	1	OBRSMS	Record-independent switches:
	1...		More records follow.
	0...		Last record.
	.x..		Time-of-Day clock instruction issued (0 = IBM System/360, 1 = IBM System/370). Used in conjunction with date and time values at displacements 8 and 12.
	..1.		Record truncated.
	...1		370-XA mode record.
	.... 1...		TIME macro used (MVS).
	.... 0...		Time in timer units (VSE).
	xxx		Reserved.
3(3)	3		Record-dependent switches:
	Byte 0	OBRSW2	
	1...		Statistical data (SDR) counters dumped at EOD.
	.1..		Temporary switch, counter overflow.
	..0.		Short OBR record.
	...1 x...		Alternate channel path was retried.
	.... x...		Not used.
	.... .1..		Volume demount.
	.... ..1.		SVC requested (VSE only).
	1		Field OBRSECUA contains polling characters instead of CUA. (Set for BTAM and TCAM OBR records only).
	Byte 1	OBRXASW	
	1...		CHPID is invalid.
	.x..		Sense unavailable.
	..xx		Reserved.
	 xxxx		Model/version number; used with device type to indicate model or version.
	Byte 2	OBRCSID	Channel set ID, for S/370. Unused for 370-XA.
6(6)	1	OBRRCDCT	Record count:
	xxxx		Sequence number of this physical record.
	 xxxx		Total number of physical records in this logical record.
7(7)	1		Reserved.
8(8)	8	OBRDT	System date and time of incident:
8(8)	4	OBRDATE	System date of failure.
12(C)	4	OBRTIME	System time of failure.
16(10)	8	OBRCPUID	CPU identification.
16(10)	1	OBRVER	Machine version code:
	xxxx xxx.		Reserved.
	0		Version I CPUs.
	1		Version II CPUs.
17(11)	3	OBRSER	CPU serial number.
20(14)	2	OBRMOD	CPU machine model number (3033, 4341, etc.).
22(16)	2	OBRCEL	Maximum length of machine- (CPU-) dependent machine-check extended logout area.
			END OF STANDARD HEADER

Figure 10-20 (Part 1 of 2). Outboard (OBR) Record Format (Long Form)

Offset Dec(Hex)	Size (bytes) Alignment (bits)	Field Name	Description
24(18)	8	OBRJOBID	Jobname or userID.
32(20)	8	OBRFCCW	CCW being executed at time of failure.
40(28)	8	OBRCSW	Contents of CSW stored following detection of I/O error. Records indicating path failures handled by alternate path recovery are identified by the unique CSW status of Interface Control Check plus Channel Control Check. Bytes 4 and 5 are the "status byte." Not used for 370-XA.
48(30)	1	OBRDEVDC	Number of doublewords used for device-dependent data.
49(31)	3	OBRSECUA	For S/370, secondary channel and unit address (actual CUA) associated with final retry of failing I/O device; or TP polling characters (BTAM and TCAM only), right-justified. For 370-XA, byte 0 contains the channel path ID (CHPID), left-justified. Bytes 1 and 2 contain the channel-path polling characters: the unmodified unit address or device number (NOT the same as the device number that replaces the PCUA).
52(34)	1		Device type for the device associated with the error.
	Byte 0		
	1...		Byte 1 contains a control unit ID.
	.xxx xxxx		Reserved.
	Byte 1		Control unit ID if Byte 0, Bit 0 is '1'. Otherwise system dependent data unused by EREP.
	Byte 2		Device class code.
	Byte 3		Device type code. See "Device Type Codes in the OBR Record" on page 12-13 for all the codes currently supported by EREP.
56(38)	1	OBRLSDRC	Number of bytes of data recorded in the statistical data area (OBRSDINF). (370-XA only.)
57(39)	3	OBRPCUA	For S/370, primary CUA of device being used when failure occurred. For IBM 2314, 3330 or 3340 series of devices, this field contains the physical location (not address) of the failing unit. For 370-XA, this field contains the device number.
60(3C)	2	OBRRETRY	Number of I/O retries attempted for this error incident.
62(3E)	2	OBRSBCNT	Number of bytes of sense data in field OBRSENSE.
64(40)	Variable	OBRDEVDP	Device-dependent data.
Variable	Variable	OBRSDINF	Statistical data area, containing statistical counter/indicator data from device statistics table.
	Variable	OBRSENSE	Device-dependent sense information received from the device.
	16	OBRIRB	For 370-XA, information from the subchannel status word (SCSW) (and the extended status word (ESW), if available). Not used for S/370.
	2	OBRRCTWD	Flag bytes from the record control table (RCT) used to create this record. (MVS and MVS/XA only.) For VSE records: • The field is 4 bytes. • It is the SIO field. However, it is not present for some devices – specifically, tape and DASD that record the SIO counts either in the device dependent data or in MDR records.

Figure 10-20 (Part 2 of 2). Outboard (OBR) Record Format (Long Form)

Software (SFT) Error Record

MVS and MVS/XA produce a software record (type X'4X') as part of the system error recovery process. The record includes such software-specific information as the ERRORID and the system diagnostic work area (SDWA) control block and its extensions for the failing task or request block. The SDWA is mapped by IHASDWA and is fully documented in the *Debugging Handbook*. It identifies the error and includes information used by the recovery routines.

The MVS systems also write a software record at the request of the machine check handler (MCH) to provide program-damage assessment data in case of a machine check. They also produce a short form of the software record to indicate the number of records lost because the error-recording (LOGREC) buffer was full.

Under VS1, VTAM prepares software records to document program failures.

VM writes no type X'4X' record of its own. It reflects back to the guest system any SFT records it detects.

Figure 10-21 shows the format of the software record.

Offset Dec(Hex)	Size (bytes) Alignment (bits)	Field Name	Description
0(0)	1	SFTKEY1	Class/Source:
	.1..		Software-detected software error; type = X'40'.
	.1.. ..1.		Hardware-detected software error; type = X'42'.
	.1.. .1..		Operator-detected error; type = X'44'.
	.1.. 1...		Hardware-detected hardware error; type = X'48'.
	.1.. 1111		Lost-record summary record; type = X'4F'.
1(1)	1	SFTKEY2	System/release level:
	xxx.		OS/360.
	xxx.		DOS (VSE systems).
	010.		OS/VS1.
	xxx.		CP67, VM/370, and later VM systems.
	100.		OS/VS2 and later MVS systems.
	...x		Reserved.
	 xxxx		Release level (0-1F).
2(2)	1	SFTSMS	Record-independent switches:
	1...		More records follow.
	0...		Last record.
	.x..		Time-of-Day clock instruction issued (0 = IBM System/360, 1 = IBM System/370). Used in conjunction with date and time values at displacements 8 and 12.
	..1.		Record truncated.
	...1		370-XA mode record.
	.... 1...		TIME macro used (MVS).
	.... 0...		Time in timer units (VSE).
	xxx		Reserved.
3(3)	3		Record-dependent switches:
	Byte 0	SFTSW2	
	1...		Short record. (Set for type X'4F' records.)
	.1..		Record incomplete. (Record truncated because of lack of buffer space.)
	..1.		Record contains an ERRORID.
	...x xxxx		Not used.
	Byte 1		Reserved.
	Byte 2		Reserved.
6(6)	1	SFTRCDCT	Record count:
	xxxx		Sequence number of this physical record.
	 xxxx		Total number of physical records in this logical record.
7(7)	1		Reserved.
8(8)	8	SFTDT	System date and time, as:
8(8)	4	SFTDATE	System date of failure.
12(C)	4	SFTTIME	System time of failure.
16(10)	8	SFTCPUID	CPU identification.
16(10)	1	SFTVER	Machine version code:
	xxxx xxx.		Reserved.
	0		Version I CPUs.
	1		Version II CPUs.
17(11)	3	SFTSER	CPU serial number.
20(14)	2	SFTMOD	CPU machine model number (3033, 4341, etc.).
22(16)	2		Not used.
			END OF STANDARD HEADER
24(18)	8	SFTJOBID	Alphameric name assigned to job (as identified, for example, by a jobname on a JCL JOB statement) being executed and/or requesting service at time of failure. In the short software record, this byte contains the count of records lost because the LOGREC buffer was full.
32(20)	Variable	SFTSDWA	System diagnostic work area, detailed by IHASDWA mapping macro. The SDWA is filled in and used by the FRR or ESTAE/ESTAI in recovery processing. See your *Debugging Handbook* for the detailed format.
Variable	10	ERRORID	RTM-generated error identifier consisting of: 2-byte sequence number 2-byte CPU identifier 2-byte ASID 4-byte time stamp The error ID is always the last 10 bytes of the record.

Figure 10-21. Software Record Format

Subchannel Logout Handler (SLH) Record

In S/370-XA, SLH records are produced for subchannel-detected errors that do not terminate system operation. The SLH record, combined with the CRW record, corresponds to the CCH record written for S/370 channel checks. The record contains subchannel-dependent error information reflecting the type (storage or key) and location (CCW, IDAW, buffer) of the error, if available from the logout information in the extended status word (ESW).

Figure 10-22 shows the format of the SLH record.

Offset Dec(Hex)	Size (bytes) Alignment (bits)	Field Name	Description
0(0)	1	SLHKEY1	Class/Source:
	..1. ..11		SLH Record; type = X'23'.
1(1)	1	SLHKEY2	System/release level:
	xxx.		OS/360.
	xxx.		DOS (VSE systems).
	xxx.		OS/VS1.
	xxx.		CP67, VM/370, and later VM systems.
	100.		OS/VS2 and later MVS systems.
	...x		Reserved.
	 xxxx		Release level (0-1F).
2(2)	1	SLHSMS	Record-independent switches:
	1...		More records follow.
	0...		Last record.
	.x..		Time-of-Day clock instruction issued (0 = IBM System/360, 1 = IBM System/370). Used in conjunction with date and time values at displacements 8 and 12.
	..1.		Record truncated.
	...1		370-XA mode record.
	.... 1...		TIME macro used (MVS).
	.... 0...		Time in timer units (VSE).
	.... .1..	SLHPASS	Passed, recovery status. VM passed error to guest.
	xx		Reserved.
3(3)	2		Record-dependent switches:
	Byte 0	SLHSW2	Reserved.
	Byte 1	SLHSW3	Reserved.
5(5)	1	SLHSWS	
	xxxx xx..		Reserved.
		SLHSFTSW	Recovery status bits. See "3090 Processor" on page 20-7.
	00		Invalid or, for VM, passed to guest.
	01		Hard fail.
	10		Degrade fail.
	11		Soft fail.
6(6)	1	SLHRCDCT	Record count:
	xxxx		Sequence number of this physical record.
	 xxxx		Total number of physical records in this logical record.
7(7)	1		Reserved.
8(8)	8	SLHDT	System date and time, as:
8(8)	4	SLHDATE	System date of failure.
12(C)	4	SLHTIME	System time of failure.
16(10)	8	SLHCPUID	CPU identification.
16(10)	1	SLHVER	Machine version code:
	xxxx xxx.		Reserved.
	0		Version I CPUs.
	1		Version II CPUs.
17(11)	3	SLHSER	CPU serial number.
20(14)	2	SLHMOD	CPU machine model number (3033, 4341, etc.).
22(16)	2	SLHCEL	Maximum length of machine- (CPU-) dependent machine-check extended logout area.
			END OF STANDARD HEADER

Figure 10-22 (Part 1 of 2). Subchannel Logout (SLH) Record Format

Offset Dec(Hex)	Size (bytes) Alignment (bits)	Field Name	Description
24(18)	8	SLHJOBID	Identity of job executing at time of failure.
32(20)	8	SLHFCCW	Last CCW executed before failure.
40(28)	4	SLHDEVT	Code for device class and type of failing device. The first two bytes are not used for this record. The second two bytes are the OS/VS UCB device class and type bytes 18 and 19 (decimal). See "Device Type Codes in the OBR Record" on page 12-13 for a list of the UCB codes.
44(2C)	8	SLHERPIB	ERP information block (Information from the ESW (extended status word)).
44(2C)	1	SLHESW01	Byte 0 of the ESW.
45(2D)	3	SLHRSVD1	Reserved.
48(30)	1	SLHFLG1	Flag byte: (Byte 0 of the ESW, modified)
	0...	SLHSSCH	No status stored after SSCH.
	.1..	SLHINT	Status stored after I/O interrupt.
	..0.	SLHTSCH	No status stored after TSCH.
	...0	SLHHSCH	No status stored after HSCH.
	.... x...		Reserved.
	.... .1..	SLHSENSE	Sense data was stored.
	.... ..1.	SLHCSWCT	CSW count is valid.
	1	SLHRETRY	Operation cannot be retried.
49(31)	1	SLHLPUM	Last-path-used mask (LPUM). (Byte 1 of the ESW)
50(32)	1	SLHVALID	Validity indicators: (Byte 2 of the ESW, modified.)
	x...		Reserved.
	.1..	SLHVLPUM	LPUM consistent with log indicator.
	..1.	SLHVTERM	Termination code valid.
	...1	SLHVSEQC	Sequence code valid.
	.... 1...	SLHVDVST	Device status valid.
	.... .1..	SLHVCCW	CCW address valid.
	11	SLHVDVNO	Device number valid.
51(33)	1	SLHTRMSQ	Termination and sequence codes: (Byte 3 of the ESW.)
	xx..	SLHTRMCD	Termination code:
	00..		Interface disconnect.
	01..		Stop, stack or normal termination.
	10..		Selective reset.
	..xx		Reserved.
	.... 1...	SLHIOALT	I/O error alert.
	xxx	SLHSEQCD	Sequence code:
	.... .000		Reserved.
	.... .001		Command sent but status not analyzed.
	.... .010		Command accepted by device but no data transferred.
	.... .011		At least one byte of data has been transferred.
	.... .100		Command not sent or sent but not yet accepted.
	.... .101		Command accepted but data transfer unpredictable.
	.... .110		Reserved.
	.... .111		Reserved.
52(34)	64	SLHIRB	IRB; includes the SCSW (subchannel status word), the ESW (extended status word), and the ECW (extended control word; model-dependent). See your *Debugging Handbook* for the detailed format of the IRB.
116(74)	4	SLHUCBAD	UCB or RDEV address.
120(78)	2	SLHDEVNO	Device number.
122(7A)	6	SLHVOLSR	Volume serial number.
128(80)	5	SLHUCBLV	UCB level byte and mask.
133(85)	2		Reserved.
135(87)	1	SLHCHPID	Channel path ID.
136(88)	4	SLHSID	Subchannel ID number.
140(8C)	4	SLHRSMAD	Absolute address of storage or key error, if available.
144(90)	2	SLHRSMRC	RSM return code for storage or key error.
146(92)	2	SLHRSMER	Error type:
	Byte 0		Reserved.
	Byte 1	SLHCKBYT	Last two bits indicate storage or key error:
	xxxx xx..		Reserved.
	01		Storage error.
	10		Key error.
	00		Other.
148(94)	4	SLHRSMST	RSM status information.

Figure 10-22 (Part 2 of 2). Subchannel Logout (SLH) Record Format

Chapter 11. EREP Messages

This section of the *EREP User's Guide and Reference* contains the messages issued by the IFCEREP1 program modules. These are the messages that appear as the TOURIST[1] output, either along with the printed report or on the terminal screen. Some of the messages listed here also appear in the report output.

Also included in this chapter are the messages issued by CMS and CPEREP during VM EREP processing. These messages can appear on the display terminal screen in the course of CPEREP processing, but are not really EREP messages. See the *Messages and Codes* manual for your VM system for general information about the messages issued by CMS on behalf of other programs. The CMS/CPEREP messages are listed and explained in "CPEREP Messages for VM Users" on page 11-48.

Introduction to IFC-prefixed Messages

Although all the IFC-prefixed EREP message numbers are followed by "I," meaning that they are informational, they can in fact indicate both the status of EREP processing and the occurrence of a problem with EREP processing or your EREP/system controls. When IFCEREP1 encounters a severe error, it stops. See "EREP Return Codes" on page 11-54 for the return codes issued by IFCEREP1.

EREP messages are prefixed with "IFC," the prefix for the program itself (IFCEREP1). They are listed here in ascending order of the numbers that follow the prefix. Note that not all the messages apply to all three operating systems. The explanation for each message indicates which system(s) the message is for.

In "Problem Determination Aids" on page 11-54 are three tables of recommended general problem determination actions. Many message descriptions include references to numbered items in these tables to help you get started on diagnosing a problem.

[1] For VSE systems, the messages and the report output are written to the SYSLST logical unit.

Message Format

The following table summarizes the source and format of the IFC-prefixed EREP messages as presented in this book:

Component prefix:	IFC
Program producing messages:	IFCEREP1
Where produced:	TOURIST data set or SYSLST; with report output or at terminal.
Audience:	System programmer
Message format:	IFCnnnI text nnn is the message serial number. text is the text of the message.

Following the text of each message are other kinds of information:

Explanation:	Under which system(s) the message appears, and what the message means.
System Action:	What happens when the message is issued.
Programmer Response:	What the system programmer should do when the message appears.
Problem Determination:	Items in the problem determination tables that may help document the problem.

EREP Messages for MVS, VM, and VSE Users

IFC101I

REQUEST FOR NON-EXISTENT I/O SERVICE

Explanation: (MVS, VM, and VSE). An internal request for I/O service specified an invalid request code.

System Action: The request is ignored. No further input is processed.

Programmer Response: Make sure the system controls are correct, then rerun the job. If the problem persists, perform problem determination.

Problem Determination: Table I, items 2, 4, 29.

IFC102I

ddname **OPEN REQUESTED, ALREADY OPEN**

Explanation: (MVS and VM). A second open was requested for a data set that is already open.

System Action: The request is ignored. No further input is processed.

Programmer Response: Make sure the DD statements or FILEDEFs are correct, then rerun the job. If the problem persists, perform problem determination.

Problem Determination: Table I, items 2, 4, 29.

IFC103I

ddname **DD STATEMENT MISSING OR INCORRECTLY CODED**

Explanation: (MVS and VM). The named data set could not be opened because the required DD statement or FILEDEF is missing or invalid. For an existing data set, the DD statement or FILEDEF may be correct but the attributes (RECFM, BLKSIZE) invalid.

System Action: EREP terminates.

Programmer Response: Add or correct the indicated system control and rerun the job.

Problem Determination: Table I, items 2, 4, 29.

IFC104I

ddname **NOT OPEN WHEN {READ|WRITE} REQUESTED**

Explanation: (MVS and VM). The named data set was not open when a read or write was requested.

System Action: The request is ignored. No further input is processed.

Programmer Response: Make sure the DD statements and FILEDEFS are correct, then rerun EREP. If the problem persists, perform problem determination.

Problem Determination: Table I, items 2, 4, 29.

IFC105I

RECORD IGNORED; *ddname* **READ ERROR**

Explanation: (MVS and VM). A permanent I/O error has occurred on the named data set.

System Action: Processing continues. The physical record that caused the error is ignored.

Programmer Response: Move the volume containing the data set to another device, or move the data set to another volume, to determine if the problem was caused by a hardware malfunction.

Warning: Move the suspect volume only once to ascertain a fault. Indiscriminate mounting and demounting of the disk pack could cause the destruction of packs and drives.

For MVS systems: If the message does not recur, there probably is a hardware error on the device (or volume) originally used. If the error persists, execute the SPZAP (VS2), or HMASPZAP (VS1) service aid program to obtain a dump of the data set on which the input error occurred. If the error occurred on SYS1.LOGREC, execute IFCDIP00 to reinitialize the data set.

For VM systems: If the error occurred in the error-recording area, issue the CPEREP command, with the CLEAR/CLEARF operand, to reinitialize the cylinders.

Problem Determination: Table I, items 2, 4, 29, or 30.

IFC106I

ddname **CLOSE REQUESTED,** *ddname* **NOT OPEN**

Explanation: (MVS and VM). The ddname data set was not open when a close was requested.

System Action: The request is ignored.

Programmer Response: Make sure the system controls are correct, then rerun the job. If the problem persists, perform problem determination.

Problem Determination: Table I, items 2, 4, 29.

IFC107I

ACCIN RECORD FORMAT NOT V OR VB

Explanation: (MVS and VM). The ACCIN DD statement or FILEDEF that defines the history input data set either:

- does not specify RECFM, or
- does not specify the RECFM as V or VB, or
- specifies a volume or CMS file that does not contain variable format records.

System Action: The job step terminates.

Programmer Response: Verify that the record format of the data set is V or VB and is properly specified on the DD statement or FILEDEF.

IFC108I

ATTEMPTED TO READ OUTSIDE SERLOG EXTENT

Explanation: (MVS). IOS indicates an attempt was made to read outside the extent on SERLOG (SYS1.LOGREC). The LOGREC header may be bad.

System Action: EREP continues processing. The record that caused the input error is ignored. SYS1.LOGREC is not cleared.

Programmer Response: Execute the SPZAP (VS2), or HMASPZAP (VS1) service aid program to obtain a dump of the data set on which the input error occurred. Determine if the problem was caused by a hardware malfunction. If the message does not recur, there probably is a hardware error on the device (or volume). Otherwise, it is probably a programming error. Execute the IFCDIP00 program to reinitialize SYS1.LOGREC.

IFC109I

SERLOG HEADER CANNOT BE READ

Explanation: (MVS). The header record on the SYS1.LOGREC data set could not be read.

System Action: The job step terminates.

Programmer Response: Execute the SPZAP (VS2), or HMASPZAP (VS1) service aid program to obtain a dump of the SYS1.LOGREC data set. Then execute the IFCDIP00 program to reinitialize the SYS1.LOGREC data set.

IFC110I

SERLOG HEADER CHECK BYTE INCORRECT

Explanation: (MVS). A validity check of the header record on SYS1.LOGREC has uncovered an error.

System Action: EREP terminates.

Programmer Response: Execute the SPZAP (VS2), or HMASPZAP (VS1) service aid program to obtain a dump of the SYS1.LOGREC data set to verify the output from the EREP program. Then execute the IFCDIP00 program to reinitialize the SYS1.LOGREC data set.

Problem Determination: Table I, items 2, 4, 29.

IFC111I

OPEN REQUESTED, DATA SET NOT SPECIFIED

Explanation: (MVS, VM and VSE). An OPEN was requested but the data set to be opened was not indicated.

System Action: EREP terminates.

Programmer Response: Make sure the DD statements or FILEDEFS are correct, then rerun the job. If the problem persists, perform problem determination.

Problem Determination: Table I, items 2, 4, 29.

IFC112I

READ REQUESTED, NO DATA SET OPEN

Explanation: (MVS, VM, and VSE). EREP cannot perform the requested read operation because no data set is open.

System Action: EREP terminates.

Programmer Response: Make sure the DD statements or FILEDEFS are correct, then rerun the job. If the problem persists, perform problem determination.

Problem Determination: Table I, items 2, 4, 29.

IFC113I

RECORDS IGNORED, INSUFFICIENT SPACE ON DIRECTWK

Explanation: (MVS and VM). Not enough space was allocated to the DIRECTWK data set to allow EREP to process all the input records. Message IFC114I should follow this message.

System Action: Processing continues. Output will be based on the input read prior to the record that could not be written on DIRECTWK; no further input was processed.

Programmer Response:
For MVS: Increase the space allocation for DIRECTWK and rerun the job.

For VM: Erase unnecessary files on the disk; or access a larger disk, possibly a temporary disk. (See the CP DEFINE command and the CMS FORMAT command.) Then rerun CPEREP.

IFC114I

LAST RECORD PROCESSED WAS *text data...*

Explanation: This message follows IFC113I and provides a hexadecimal dump of the first 40 bytes of the last record processed before the space on DIRECTWK was exhausted.

System Action: None.

Programmer Response: None.

IFC116I

SYS1.LOGREC HEADER CANNOT BE RESET. USE IFCDIP00

Explanation: (MVS). The header record of the SYS1.LOGREC data set cannot be reset because of an uncorrectable output error.

System Action: The program terminates normally.

Programmer Response: Execute the IFCDIP00 program to reinitialize the SYS1.LOGREC data set.

Problem Determination: Table I, items 2, 4, 30.

IFC117I

SERLOG CLOSED PREMATURELY. USE IFCDIP00

Explanation: (MVS and VM). When EREP tried to check the ERDS header for records written while processing, it found that the data set was already closed.

System Action: The request is ignored; the ERDS is not cleared.

Programmer Response: If you got all the report output you expected, run IFCDIP00 or CPEREP with CLEAR/CLEARF to reinitialize LOGREC. Records written on SYS1.LOGREC during processing will be lost.

Problem Determination: Table I, items 2, 4, 29.

IFC118I

GETMAIN FAILURE WHILE CLEARING SYS1.LOGREC

Explanation: (MVS). While EREP was clearing LOGREC, it tried to obtain storage for the records written to LOGREC during EREP's previous processing, but the GETMAIN failed.

System Action: Processing continues. However, those records for which EREP could not obtain storage are lost.

Programmer Response: The next time EREP is executed, increase the region size. Investigate the possibility that a large number of error records were written on SYS1.LOGREC during EREP processing.

IFC119I

RECORDS IGNORED, TABSIZE ALLOCATION TOO SMALL

Explanation: (MVS, VM, and VSE). EREP's internal sort table, controlled by the TABSIZE parameter, is too small for this report.

System Action: Processing continues.

Programmer Response: Increase the value of the TABSIZE parameter, increase the region, virtual machine storage or partition size if necessary, and rerun the job step. If running IFCOFFLD, you need only increase the region, virtual machine storage or partition size.

IFC120I

nnnnnn **RECORDS THAT PASSED FILTERING SAVED FOR** *rrrrrrr*

Explanation: (MVS, VM, and VSE). Indicates the number of records that EREP selected and used to generate the requested report; *rrrrrrr* is one of the following:

SYSEXN
SYSUM PART 1
SYSUM PART 2
TREND PART 1
TREND PART 2

System Action: None.

Programmer Response: None.

IFC121I

GETMAIN FAILED FOR *ttttttt* **TABLE**

Explanation: (MVS and VM). EREP issued a GETMAIN for the amount of storage indicated by the TABSIZE parameter, but not enough storage was available; *ttttttt* is one of the following:

DASDID
LIMIT
SHARE
SORT
SUMM

System Action: EREP terminates.

Programmer Response:
For MVS: Increase the region size on the JOB and/or EXEC statement, and rerun the job.

For VM: Rerun CPEREP in a virtual machine having a larger virtual storage capacity; or, if the TABSIZE value was larger than necessary, rerun with a smaller value for the TABSIZE parameter.

IFC122I

nnnnnn **RECORDS IGNORED BECAUSE TRUNCATED BIT ON**

Explanation: (MVS, VM, and VSE). Indicates the number of records EREP found that had the truncated bit set on.

System Action: The records are ignored; when you code the TYPE parameter, EREP does not process truncated or unknown records.

Programmer Response: None.

IFC123I

nnnnnn **RECORDS IGNORED BECAUSE OF UNKNOWN TYPE**

Explanation: (MVS and VM). Indicates the number of records EREP found that were from an unsupported source.

System Action: The records are ignored; when you code the TYPE parameter, EREP does not process truncated or unknown records.

Programmer Response:
For MVS: Execute the SPZAP (VS2), or HMASPZAP (VS1) service aid program to obtain a dump of the output data set to verify the existence of the records of unknown type.

For VM: Try to determine which device triggered the error records.

IFC129I

nnnnnnnnn **RCDS IGNORED BECAUSE DIRECTWK READ ERRORS**

Explanation: (MVS and VM). Indicates the number of records EREP could not process because of I/O errors in reading the DIRECTWK data set.

System Action: Processing continues.

Programmer Response: Rerun the job. If the problem persists, check the DASD device or CMS disk on which the DIRECTWK data set resides.

Problem Determination: Table I, items 2, 4, 30.

IFC130I

UNABLE TO FIND MODULE SPECIFIED BY USERPGM

Explanation: **(MVS).** EREP was unable to find the program requested via the USERPRG parameter.

System Action: EREP terminates.

Programmer Response: Verify that the user program requested was correct, and that the program is in SYS1.LINKLIB.

IFC131I

SYNTAX ERROR AT *

Explanation: **(MVS and VM).** The EREP controls that appear above this message contain a syntax error. The error is in the keyword or operand above the asterisk. This message also appears when EREP encounters a device type on the DEV parameter that it does not recognize.

System Action: EREP terminates.

Programmer Response: Correct the parameter and rerun the job.

IFC132I

DUPLICATION AT *

Explanation: **(MVS and VM).** The EREP controls that appear above this message contain a duplicate keyword or operand. The duplicate is above the asterisk.

System Action: EREP terminates.

Programmer Response: Eliminate the duplicate keyword or operand, and rerun the job.

IFC133I

PARAMETER CONFLICTS - *parameter text*

Explanation: (MVS and VM). The EREP controls that appear above this message contain parameters, either specified or implied, that are mutually exclusive.

System Action: EREP terminates.

Programmer Response: Eliminate the conflicting parameters and rerun the job.

IFC134I

MORE THAN {10|16} CPUs
{SPECIFIED WITH SHARE CARDS|ENCOUNTERED}
nnnn **RECORDS DROPPED**

Explanation: (MVS, VM, and VSE).

- EREP found SHARE statements specifying more than 16 processors (CPUs), or
- the data set(s) being processed contained records from more than the indicated number of CPUs, and the EREP controls did not include the CPU or MOD selection parameter.

The System Summary report is limited to 10 processors; all other reports can show up to 16.

System Action: If it is a case of SHARE statements specifying more than 16 processors altogether, EREP terminates. Otherwise, processing continues, but the output does not show all possible processors, only the first 10 or 16.

Programmer Response: Use the CPU or MOD selection parameter to restrict the number of processors whose records are to be processed, and rerun the job.

IFC135I

PROCESSING TERMINATED, *ddname* **{READ|WRITE} ERROR**

Explanation: (MVS and VM). A permanent I/O error has occurred on the *ddname* data set.

VM note: If ddname is ACCDEV, the following may have occurred: the user did not want the records accumulated, but failed to code ACC=N, so the default of ACC=Y is in effect. If tape 181 is not attached to the virtual machine, this I/O error results.

System Action: EREP terminates; the records are not accumulated.

Programmer Response:
For VM: If the situation described in the note applies, rerun the job with ACC=N. Otherwise, move the volume containing the data set to another volume, to determine if the problem was caused by a hardware malfunction.

For MVS: Move the volume or data set to determine if the problem was caused by a hardware malfunction. If the message recurs, execute the SPZAP (VS2), or HMASPZAP (VS1) service aid program to obtain a dump of the data set on which the input error occurred. If the error occurred on SYS1.LOGREC, run the IFCDIP00 program to reinitialize the data set.

Problem Determination: Table I, items 2, 4, 29, 30.

Warning: Move the suspect volume only once to ascertain a fault. Indiscriminate mounting and demounting of the disk pack could cause the destruction of packs and drives.

IFC136I

CLOSE REQUESTED, NO DATA SET OPEN

Explanation: (MVS, VM, and VSE). EREP received a request for the CLOSE of a data set, but no data set is open.

System Action: EREP terminates.

Programmer Response: Make sure the system controls are correct, then rerun the job. If the problem persists, perform problem determination.

Problem Determination: Table I, items 2, 4, 29.

IFC137I

RECORD WITHOUT CPU SERIAL NUMBER ENCOUNTERED

Explanation: (MVS, VM, and VSE). EREP encountered a record with a processor serial number of 000000.

System Action: The record is ignored.

Programmer Response: None.

IFC138I

FRAME LOST WHILE WRITING TO ACCDEV

Explanation: (MVS and VM). EREP encountered an error when writing a frame record to the ACCDEV data set.

System Action: EREP terminates.

Programmer Response: Rerun the job. If the problem persists, check the device on which the data set resides.

IFC139I

{MCF | CCF} FRAME *xx* MISSING – MOD *yyyy* SER *zzzzzz*

Explanation: (MVS and VM). EREP could not find the expected frame record.

System Action: Processing continues. However, part of the data record will not be edited. Additional messages may appear in the report output.

Programmer Response: Reinitialize the ERDS for the processor with the indicated serial number, using IFCDIP00 with PARM='FRAMES' or CPEREP with CLEARF. Then rerun the job on that CPU with the MERGE parameter included in the EREP controls.

Problem Determination: Save all associated output for analysis.

IFC140I

FRAME CPU-SERIAL-NUMBER TABLE OVERFLOWED

Explanation: (MVS, VM, and VSE). EREP has encountered more processors than the frame table can hold (16).

System Action: Processing continues, but some CCH or MCH records may not be edited with frames.

Programmer Response: Rerun the job and restrict the number of processors by using the CPU selection parameter.

IFC141I

CORE NOT AVAILABLE FOR FRAME PROCESSING

Explanation: **(MVS, VM, and VSE).** A GETMAIN or GETVIS for additional storage failed.

System Action: EREP terminates.

Programmer Response: Increase the amount of virtual storage available to EREP and rerun the job.

IFC142I

nnnnnn **RECORDS FOUND WITH INVALID DATE FIELD**

Explanation: **(MVS, VM, and VSE).** EREP has encountered one or more records with an invalid date field. The last half byte was not an X'F'.

System Action: The record is ignored and processing continues.

Programmer Response: None.

IFC143I

NUMBER OF ***xxx*** **TYPE RECORDS READ WAS** ***nnnnnn***

Explanation: **(MVS, VM, and VSE).** Indicates the number of records of each type that EREP has processed for a Detail Edit or Summary (PRINT) report.

System Action: None.

Programmer Response: None.

IFC144I

SCAN ERROR CODE AT ***

Explanation: **(MVS and VM).** A scan command in a frame record was found for which there is no defined action.

System Action: Processing continues; the frame is dumped in hexadecimal format to the EREPPT data set. '***' is placed in the normal print line in the position corresponding to the location in the frame where the error occurred.

Programmer Response: Reinitialize the ERDS, using IFCDIP00 with PARM='FRAMES' or CPEREP with CLEARF, then rerun the job using the MERGE parameter.

Problem Determination: Save all the associated output for analysis.

IFC145I

FRAME SET MISSING *yyyy zzzzzz*

Explanation: **(MVS and VM).** EREP has identified a missing frame set for processor model *yyyy* and serial *zzzzzz*.

System Action: MCH and CCH records for processor model *yyyy*, serial *zzzzzz* will not be edited correctly because the frame set needed to edit them is missing.

Programmer Response: If the ERDS was the input data set, you may need to reinitialize it (IFCDIP00 with PARM='FRAMES' or CPEREP with CLEARF) to make sure all the frames are there.

IFC146I

NO FRAMES AVAILABLE {MCH|CCH} MOD *yyyy* **SER** *zzzzzz*

Explanation: **(MVS and VM).** No frames are available for processing the MCH or CCH error record with this model and serial number.

System Action: Processing continues. The error record is not edited, or is edited with frames for the same model number only.

Programmer Response: Reinitialize the ERDS using IFCDIP00 with PARM='FRAMES' or CPEREP with CLEARF, then rerun the job using the MERGE parameter.

IFC147I

LOG ERR {MCF|CCF} FRAME *xx* **MOD** *yyyy* **SER** *zzzzzz.*

Explanation: **(MVS and VM).** EREP detected an invalid log type scan code in the frame.

System Action: Processing continues. This frame is not used; part of the error record is not summarized.

Programmer Response: Reinitialize the ERDS using IFCDIP00 with PARM='FRAMES' or CPEREP with CLEARF, then rerun the job using the MERGE parameter.

IFC148I

xxx **RECORDS REQUESTED BUT NOT FOUND**

Explanation: (MVS, VM, and VSE). You requested Detail Edit or Summaries for record type *xxx*. Either EREP did not find any records of that type on the input data set, or none of the records passed filtering. (By date, for example.)

System Action: Processing continues.

Programmer Response: If you want to see records of this type, modify the selection parameters in the EREP controls for this report.

IFC149I

nnnnnn **DIRECTWK READ FAILURES**

Explanation: (MVS and VM). Indicates the number of records that were lost while reading from the DIRECTWK data set.

System Action: Processing continues.

Programmer Response: Rerun the job. If the problem persists, check the direct access device on which the data set resides.

Problem Determination: Save the console spool file. Contact IBM for hardware support.

IFC150I

nnnnnn **RECORDS READ FROM INPUT SOURCE**

Explanation: (MVS, VM, and VSE). Indicates the number of records EREP read for the report.

System Action: None.

Programmer Response: None.

Problem Determination: None.

IFC152I

nnnnnn **RECORDS FOUND WITH A ZERO VOLID**

Explanation: (MVS, VM, and VSE). Indicates the number of records EREP found that contained a volume serial of 000000.

System Action: None.

Programmer Response: None.

Problem Determination: None.

IFC153I

GETMAIN FAILED FOR MODULE *mmmmmmmm*

Explanation: (MVS and VM). The region or storage size is too small to contain the tables for this module.

System Action: EREP terminates.

Programmer Response: Increase the region size or the virtual machine storage size and rerun the job.

IFC154I

SORTBREAK FORCED DUE TO EXCESSIVE FAULT CODES

Explanation: (MVS, VM, and VSE). EREP has encountered more different fault symptom codes than the symptom code table can hold.

System Action: The DASD device summary for this channel/control unit will be two (or more) reports rather than one.

Programmer Response: Increase the region/partition or virtual machine storage size. If the problem continues, limit the amount of data by use of selection parameters.

IFC165I

SORTBREAK FORCED DUE TO EXCESSIVE VOLIDS

Explanation: (MVS, VM, and VSE). EREP has encountered more unique volume identifiers than the VOLID table can hold.

System Action: The DASD Detail Summary for this channel/control unit will be two (or more) reports rather than one.

Programmer Response: Increase the region/partition or virtual machine storage size. If the problem persists, restrict the amount of data by use of selection parameters.

IFC166I

ttttttt **TABLE FULL; INCREASE TABSIZE**

Explanation: (MVS, VM, and VSE). The area allocated to the specified table has been filled; *ttttttt* is one of the following:

DASDID
LIMIT
SHARE
CONTROLLER
SUMM

System Action: EREP terminates.

Programmer Response: Increase the TABSIZE value and, if necessary, the region/partition or virtual machine storage size as well. Then rerun the job.

IFC167I

CUA RANGE IS INVALID ON A SHARE/CONTROLLER CARD

Explanation: (MVS, VM, and VSE). The range specified on the SHARE or CONTROLLER statement either exceeds the 32-address limit, or crosses an invalid control unit boundary. For example, the range on SHARE=(. . . 130-14F) crosses from an odd to an even CUA and is invalid.

System Action: EREP terminates.

Programmer Response: Correct the SHARE/CONTROLLER statement and rerun the job.

IFC168I

CUA OVERLAPS WITH ANOTHER SHARE/CONTROLLER ENTRY

Explanation: (MVS, VM, and VSE). The address range on one SHARE or CONTROLLER statement overlaps the range on another SHARE or CONTROLLER statement.

System Action: EREP terminates.

Programmer Response: Correct the SHARE or CONTROLLER statement(s) and rerun the job.

IFC169I

nnnn **RECORDS NOT USED BY** *module name* **FOR CUX** *xxx*

Explanation: (MVS, VM, and VSE). Indicates why the number of records used to build the maintenance device code does not equal the number of records present for this channel/control unit: all MDR and OBR records are passed to EREP, but only OBR records with particular fault symptom codes are used for the Data Reduction report.

System Action: Processing continues.

Programmer Response: None.

Problem Determination: None.

IFC170I

GETVCE FAILURE. LOGICAL UNIT SYS*xxx*

Explanation: (VSE). The get-device-characteristics SVC has failed. The device type needed to open SYS*xxx* cannot be obtained.

System Action: The job step terminates.

Programmer Response: Correct or add the // ASSGN statement for the appropriate logical unit.

IFC171I

INVALID DEVICE TYPE SYS*xxx*

Explanation: (VSE). The device assigned to logical unit SYS*xxx* is invalid for the type of processing that must be performed.

System Action: The job step terminates.

Programmer Response: Correct the // ASSGN statement for SYS*xxx*.

IFC172I

SEGMENTED RECORD INCOMPLETE (24-byte header)

Explanation: (VSE). A segment of a logical record on SYSREC is missing or incorrect. The first 24 bytes of the record are included in the message.

System Action: Not all of the record's segments are processed. If the segment involved belongs to a frame or to SYSREC, the entire frame set is deleted, so some MCH and CCH records might not be processed.

Programmer Response: Check for a succeeding read error message. You may have to reallocate and reinitialize IJSYSRC. An error-recording transient may be executing incorrectly. Call IBM programming support.

IFC173I

ERROR READING SYSREC, RECORD SKIPPED

Explanation: (VSE). A read error occurred on SYSREC.

System Action: Processing continues.

Programmer Response: Reallocate IJSYSRC and reinitialize SYSREC using the SET RF=CREATE IPL command.

IFC174I

***nnnn* RECORDS WITH SENSE BYTES 3 & 4 EQUAL TO SENSE BYTES 8 & 9**

Explanation: (MVS, VM, and VSE). OBR records with fault symptom code 191A should not have sense bytes 3 and 4 equal to sense bytes 8 and 9. This message indicates the number that do, nevertheless.

System Action: Processing continues. However, these records are not used to determine the maintenance device code.

Programmer Response: A hardware problem; notify your CE or other maintenance person.

Problem Determination: Table III, item 30.

IFC175I

logical unit **OPEN REQUESTED, ALREADY OPEN**

Explanation: (VSE). A second open was requested for a data set that is already open.

System Action: The request is ignored. No further input is processed.

Programmer Response: Make sure the system controls are correct, then rerun the job. If the problem persists, perform problem determination.

Problem Determination: Table III, items 2, 4, 29.

IFC176I

logical unit **FAILED TO OPEN**

Explanation: (VSE). The specified data set could not be opened.

System Action: The job step terminates.

Programmer Response: Add or correct the // ASSGN statement for the specified data set and rerun the job.

IFC177I

logical unit **NOT OPEN WHEN {READ|WRITE} REQUESTED**

Explanation: (VSE). The specified data set was not open when a read or write was requested.

System Action: The request is ignored. No further input is processed.

Programmer Response: Make sure the system controls are correct, then rerun the job. If the problem persists, perform problem determination.

Problem Determination: Table III, 2, 4, 29.

IFC178I

RECORDS IGNORED; *logical unit* **READ DIRECT ERROR**

Explanation: (VSE). A permanent I/O error has occurred on the specified data set. EREP has ignored one or more records.

System Action: Processing continues. The physical record that caused the error is ignored.

Programmer Response: Move the volume containing the data set to another device, or move the data set to another volume, to determine if the problem was caused by a hardware malfunction. If the message does not recur, there probably is a hardware error on the device (or volume) originally used. If the error persists, execute a utility to obtain a dump of the data set on which the error occurred. If the error occurred on SYSREC, re-IPL and SET RF=CREATE to reinitialize the data set.

Warning: Move the suspect volume only once to ascertain a fault. Indiscriminate mounting and demounting of the disk pack could cause the destruction of packs and drives.

Problem Determination: Table III, items 2, 4, 29, 30.

IFC179I

logical unit **CLOSE REQUESTED,** *logical unit* **NOT OPEN**

Explanation: (VSE). The specified data set was not open when a close was requested.

System Action: The request is ignored.

Programmer Response: Make sure the system controls are correct, then rerun the job. If the problem persists, perform problem determination.

Problem Determination: Table III, items 2, 4, 29.

IFC180I

SYSREC HEADER CANNOT BE READ

Explanation: (VSE). EREP could not read the header record on SYSREC.

System Action: The job step terminates.

Programmer Response: Execute a utility to obtain a dump of SYSREC. Then re-IPL and SET RF=CREATE to reinitialize the recorder file (SYSREC).

IFC181I

SYSREC HEADER CHECK BYTE INCORRECT

Explanation: (VSE). A validity check of the header record on SYSREC has uncovered an error.

System Action: The EREP program terminates.

Programmer Response: Execute a utility to obtain a dump of SYSREC. Then re-IPL and SET RF=CREATE to reinitialize the recorder file (SYSREC).

Problem Determination: Table III, items 2, 4, 29.

IFC182I

RECORDS IGNORED; INSUFFICIENT SPACE ON SYS001

Explanation: (VSE). Not enough space was allocated on SYS001 to process all input records. Message IFC183I should follow this message.

System Action: Processing continues. The report output includes only the records read prior to the record that could not be written on SYS001. EREP reads no more records for the report.

Programmer Response: Increase the space allocation for SYS001 and rerun the job.

IFC183I

LAST RECORD PROCESSED WAS *text data ...*

Explanation: (VSE). This message follows IFC1821 and provides a hexadecimal dump of the first 40 bytes of the last record processed before the space on SYS001 was exhausted.

System Action: None.

Programmer Response: None.

IFC184I

RECORDER FILE HEADER CANNOT BE RESET. RE-IPL AND SET RF=CREATE

Explanation: (VSE). The header record of SYSREC cannot be reset because of an uncorrectable output error.

System Action: The program terminates normally.

Programmer Response: Re-IPL and issue SET RF=CREATE to reinitialize SYSREC.

Problem Determination: Table III, items 2, 4, 30.

IFC185I

GETVIS FAILED FOR *ttttttt* **TABLE**

Explanation: (VSE). A GETVIS was issued for the value indicated by parameter TABSIZE and the partition GETVIS area was too small; *ttttttt* is one of the following:

DASDID
LIMIT
SHARE
SORT
SUMM
ALIAS LIST
CI BUFFER
HEADER BUFFER

System Action: The job step terminates.

Programmer Response: Alter the SIZE parameter on the // EXEC statement to increase the partition size, then rerun the job.

IFC186I

nnnnnn **RECORDS IGNORED BECAUSE OF UNKNOWN TYPE**

Explanation: (VSE). EREP has encountered records from an unsupported device.

System Action: The records are ignored; not used for the report.

Programmer Response: Execute a utility to obtain a dump of the output data set to verify the existence of the unknown records.

IFC187I

nnnnnn **RCDS IGNORED BECAUSE SYS001 READ ERRORS**

Explanation: **(VSE).** The message indicates the number of records EREP could not process because of I/O errors in reading the SYS001 data set.

System Action: Processing continues.

Programmer Response: Rerun the job. If the problem persists, check the direct access device on which the data set resides.

Problem Determination: Table III, items 2, 4, 30.

IFC188I

UNABLE TO FIND MODULE SPECIFIED BY USERPGM

Explanation: **(VSE).** EREP was unable to find the program requested via the USERPGM parameter.

System Action: EREP terminates.

Programmer Response: Verify that the user program requested was correct and that the program is on the core image library.

IFC189I

SYNTAX ERROR AT *

Explanation: **(VSE).** The EREP controls that appear above this message contain a syntax error. The error is in the keyword or operand above the asterisk. This message also appears when the DEV parameter includes a device type EREP does not recognize.

System Action: The job step terminates.

Programmer Response: Correct the parameter and rerun the job step.

IFC190I

DUPLICATION AT *

Explanation: (VSE). The EREP controls that appear above this message contain a duplicate keyword or operand. The duplicate is above the asterisk.

System Action: The job step terminates.

Programmer Response: Eliminate one of the duplicates and rerun the job step.

IFC191I

PARAMETER CONFLICTS - *parameter text*

Explanation: (VSE). The EREP controls include parameters that are mutually exclusive.

System Action: The job step terminates.

Programmer Response: Eliminate the conflicting parameters and rerun the job step.

IFC192I

PROCESSING TERMINATED; *logical unit* **{READ|WRITE} ERROR**

Explanation: (VSE). A permanent I/O error has occurred on the specified data set.

System Action: The job step terminates; SYSREC is not cleared.

Programmer Response: Move the volume containing the data set to another device, or move the data set to another volume, to determine if the problem was caused by a hardware malfunction. If the message does not recur, there is probably a hardware error on the device (or volume) originally used. If the error persists, execute a utility to obtain a dump of the data set on which the input error occurred. If the error occurred on SYSREC, re-IPL and issue SET RF=CREATE to reinitialize the data set.

Warning: Move the suspect volume only once to ascertain a fault. Indiscriminate mounting and demounting of the disk pack could cause the destruction of packs and drives.

Problem Determination: Table III, items 2, 4, 29, 30.

IFC193I

FRAME LOST WHILE WRITING TO SYS009

Explanation: (VSE). EREP encountered an error when writing a frame record to the SYS009 data set.

System Action: The job step terminates.

Programmer Response: Rerun the job. If the problem persists, check the device on which the data set resides.

IFC194I

{MCF|CCF} FRAME *xx* **MISSING FOR MOD** *yyyy* **SERIAL** *zzzzzz*

Explanation: (VSE). EREP did not find the expected frame record.

System Action: Processing continues; part of the data record is not be edited. Additional messages may appear in the report output.

Programmer Response: Reinitialize the recorder file (SYSREC) of the processor with the serial number in the message. Then rerun the job on that CPU with the EREP parameter MERGE included.

Problem Determination: Table III, item 13.

IFC195I

SCAN ERROR CODE AT ***

Explanation: (VSE). A scan command in a frame record was found for which no action is defined.

System Action: Processing continues and the frame is dumped in hexadecimal format to SYSLST. '***' appears in the normal print line in the position corresponding to the location in the frame where the error occurred.

Programmer Response: Reinitialize SYSREC, then rerun the job step using the MERGE parameter. probd.Table III, item 13.

IFC196I

FRAME SET MISSING *yyyy zzzzzz*

Explanation: (VSE). EREP has identified a missing frame for processor model *yyyy* and serial *zzzzzz*.

System Action: MCH and CCH records for this processor are not edited correctly because the frame set needed to edit them is missing.

Programmer Response: If SYSREC was the input data set, it may be necessary to reinitialize it to make sure that all frames exist.

IFC197I

NO FRAMES AVAILABLE {MCH|CCH} MOD *yyyy* **SERIAL** *zzzzzz*

Explanation: (VSE). EREP could not find the frames needed to process the MCH or CCH record with this model and serial number.

System Action: Processing continues. The error record is not edited, or is edited with frames for the same model number only.

Programmer Response: Reinitialize SYSREC, then rerun the job using the MERGE parameter.

IFC198I

LOG ERR {MCF|CCF} FRAME *xx* **MOD** *yyyy* **SERIAL** *zzzzzz*

Explanation: (VSE). EREP detected an invalid log type scan code in the frame.

System Action: This frame is not used. Part of the error record is not edited. Processing continues.

Programmer Response: Reinitialize SYSREC, then rerun the job using the MERGE parameter.

IFC199I

nnnnnn **DIRECTWK READ FAILURES**

Explanation: (VSE). EREP lost *nnnnnn* records while reading from SYS001.

System Action: Processing continues.

Programmer Response: Rerun the job. If the problem persists, check the direct access device on which the data set resides.

Problem Determination: Table III, items 2, 4, 30.

IFC200I

NUMBER OF BYTES REPORTED DIFFERS FROM RECORD COUNT

Explanation: (MVS, VM, and VSE). The number of sense bytes, or bytes of statistical data, expected is not the same as the number of sense bytes recorded by the device and specified in the OBR record. EREP formats sense bytes according to the original engineering requirements for a device's EREP support.

System Action: None. EREP has formatted the number of sense bytes it expected to find in the record.

Programmer Response: This message could appear in the report output when either:

- The number of bytes formatted is less than the total number of bytes the device actually recorded in the OBR record. In this case, the message is informational; the unformatted sense bytes are not relevant to the EREP report.

  OR

- The number of bytes formatted is greater than the number of bytes the device actually recorded in the OBR record, implying that the byte counts (statistical or sense) were recorded erroneously. In this case, the message indicates a problem.

If you suspect that the second case applies, perform problem determination, focusing on the device as well as on the system recording process.

Problem Determination: Table I, items 13, 29.

IFC201I

nnnn **RECORDS IGNORED DUE TO EXCESSIVE CPUs**

Explanation: (MVS, VM, and VSE). EREP encountered more than 16 unique CPUs in the input data.

System Action: Processing continues.

Programmer Response: Check the report output to see if the records from the CPUs you are interested in were processed. You may need to restrict the number of records by use of the CPU or MOD selection parameter.

Problem Determination: Table I, items 13, 22, 29.

IFC214I

CANNOT PROCESS RECORD: TYPE OR LOGOUT LENGTH INVALID

Explanation: **(MVS, VM, and VSE).** EREP encountered an MCH or CCH record with a logout-length field of zero, or a CCH record produced by a non-IBM system or a system other than MVS, VM or VSE.

System Action: This record is not included in the summary.

Programmer Response: Check the input record and rerun the job. If the error persists, perform problem determination.

Problem Determination: Table I, items 13, 22, 29.

IFC215I

FRAME READ ERROR: MOD *yyyy* **SER** *zzzzzz*

Explanation: **(MVS, VM, and VSE).** EREP's I/O handler could not read a frame record because of an I/O error.

System Action: Processing continues with the next record.

Programmer Response: If possible, remount the input volume on another drive and rerun the job. If the error persists, perform problem determination.

Problem Determination: Table I, items 13, 22, 29.

IFC216I

UNIDENTIFIED FRAME TYPE *xx*: **MOD** *yyyy* **SER** *zzzzzz*

Explanation: **(MVS, VM, and VSE).** During a 303X Detail Summary, EREP encountered a frame record type other than the expected MCF or CCF.

System Action: Processing continues, but this record is not used.

Programmer Response: Rerun the job. If the error persists, perform problem determination.

Problem Determination: Table I, items 13, 22, 29.

IFC217I

303X LOAD LIST IS FULL

Explanation: (MVS, VM, and VSE). EREP found the 303X load list in the summary-table module already full.

System Action: EREP terminates summary processing.

Programmer Response: Rerun the job. If the error persists, perform problem determination. This could be a hardware or IBM software problem.

Problem Determination: Table I, items 13, 22, 29.

IFC218I

303X DEFAULT SUMMARY TABLE MODULE *mmmmmmmm* USED

Explanation: (MVS, VM, and VSE). EREP used default module *mmmmmmmm* in place of the missing summary module identified in the previously issued IFC219I message.

System Action: EREP continues summary processing using the default summary table module named in the message.

Programmer Response: Make sure the latest release of EREP is installed on your system and rerun the job. If the error persists, perform problem determination.

Problem Determination: Table I, items 13, 22, 29.

IFC219I

303X SUMMARY MODULE *mmmmmmmm* NOT FOUND

Explanation: (MVS, VM, and VSE). EREP could not find the selected *mmmmmmmm* summary module.

System Action: EREP omits this record from the summary and continues summary processing using the default summary module named in message IFC218I. If the default summary-table module is missing, EREP terminates summary processing and issues message IFC220I.

Programmer Response: If message IFC218I immediately follows this message, see the programmer response for that message. If message IFC220I immediately follows, the proper level of EREP is probably not installed. Check with your PSR.

IFC220I

SEVERE ERROR. SUMMARY TERMINATED FOR THIS MODEL

Explanation: (MVS, VM, and VSE). The error mentioned in the immediately preceding message caused EREP to terminate the summary.

System Action: EREP terminates summary processing.

Programmer Response: See the message immediately preceding this message for programmer response.

IFC221I

NO SHARE CARD

Explanation: (MVS, VM, and VSE). EREP found records for more than one processor in the input but found no SHARE statements.

System Action: EREP continues processing; however, the probable failing unit could be incorrect for tape devices.

Programmer Response: Provide SHARE statements for tape devices.

IFC223I

ttttttt **TABLE ERROR**

Explanation: (MVS, VM, and VSE). The table contains a value or other data that EREP does not recognize, or does not contain the data EREP expects; *ttttttt* is one of the following:

SELECTION CRITERIA
THRESHOLD

System Action: EREP stops processing DASD records.

Programmer Response: The table either is incorrect or has been overlaid. Make sure the latest level of EREP is installed and includes all the applicable APAR/PTFs.

If the table has been replaced by PTF, remove the PTF and rerun the job.

In either case, contact your IBM PSR.

IFC225I

SCAN CODE ERROR *xxxxxx*, **MOD** *yyyy* **SER** *zzzzzz*

Explanation: (MVS, VM, and VSE). During a 303X MCH/CCH Detail Summary, EREP found an invalid scan code in a frame record.

System Action: Processing continues. However, instead of summarizing the indicators referenced by this frame code, EREP flags them with '***' in the report. EREP also issues message IFC226I to further identify the problem.

Programmer Response: Perform problem determination.

Problem Determination: Table I, items 13, 22, 29.

IFC226I

SUMMARY IN ERROR: FRAME TYPE {MCF|CCF} FRAME ID *xx*

Explanation: (MVS, VM, and VSE). The extension of the preceding message, IFC225I.

System Action: See IFC225I.

Programmer Response: See IFC225I.

IFC227I

NO DASDID CARD FOR ENTRIES FLAGGED WITH *

Explanation: (MVS, VM, and VSE). EREP found records for DASD devices for which there were no DASDID statements. The '*'-flagged entries are on the DASD Subsystem Exception report.

System Action: EREP continues processing; however, probable failing unit analysis might be incorrect.

Programmer Response: Include DASDID statements for your DASD that do not provide their own physical IDs, and rerun the job.

IFC228I

DASD RECORD THAT RESULTED IN UNKNOWN PFU FOLLOWS

Explanation: (MVS, VM, and VSE). The following record does not match the selection criteria. An error may have occurred in building the record or in the selection criteria table; or the record may be from a non-IBM DASD.

System Action: Processing continues.

Programmer Response: If the record is from an IBM DASD, contact field support to determine where the error occurred.

Problem Determination: Obtain the following documentation:

- the record following this message
- the level of EREP on your system, including APAR/PTFs.

IFC229I

MODULE *mmmmmmmm*, **RPA =** *aaaaaaaa*, **REQUESTED AN UNSUPPORTED SERVICE FUNCTION; FRF =** *bbbbbbbb*, **FCF =** *cccccccc*

Explanation: (MVS, VM, and VSE). The named module made a service request that contained an invalid or unsupported code in the function request flag (FRF) or the function control flag (FCF).

System Action: EREP ignores the request and returns control to the calling module at the specified return-point address (RPA). Register 15 contains the return code.

Programmer Response: There is an error either in the product-dependent exit module or in the product control table (PCT) for the product. Make sure EREP support is installed for the product(s) included in the module name.

Problem Determination: Save any output for analysis.

IFC230I

UNABLE TO TRANSFER CONTROL TO {MOD = *mmmmmmmm*| **PROC** *pppppppp*}; **IFCXCST OVERFLOW — CRITICAL ERROR**

Explanation: (MVS, VM, and VSE). The transfer-of-control stack table, IFCXCST, is full; EREP cannot transfer control to the named module or procedure as requested.

System Action: EREP ignores the request and returns control to the calling module. Register 15 contains the return code.

Programmer Response: Call IBM Level Two service.

IFC231I

UNABLE TO LOAD MODULE *mmmmmmmm* **FOR MODULE** *xxxxxxxx*; **LMAT OVERFLOW — CRITICAL ERROR**

Explanation: (MVS, VSE, and VM). Module *xxxxxxxx* requested, via the IFCLOAD or IFCCALL macro, that EREP load module *mmmmmmmm*. EREP cannot satisfy the request because the load-module-address table (LMAT) is full.

System Action: EREP ignores the request and returns control to the calling module. Register 15 contains the return code.

Programmer Response: Call IBM Level Two service.

IFC232I

UNABLE TO GET VIRTUAL STORAGE FOR MODULE *mmmmmmmm*; **VSAT OVERFLOW — CRITICAL ERROR**

Explanation: (MVS, VM, and VSE). The named module requested virtual storage via the IFCGETM macro. EREP cannot satisfy the request because its virtual storage address table (VSAT) is full.

System Action: EREP ignores the request and returns control to the calling module. Register 15 contains the return code.

Programmer Response: Call IBM Level Two service.

IFC233I

INVALID FUNCTION — STE BUILD MODULE *mmmmmmmm*

Explanation: (MVS, VM, and VSE). The named module was asked to do something it cannot do.

System Action: Processing continues; EREP does not include this record in the System Exception reports.

Programmer Response: There is an error either in the product-dependent exit module or in the product control table (PCT) for the product. Make sure EREP support is installed for the product(s) included in the module name.

Problem Determination: Save any output for analysis.

IFC234I

GETMAIN FOR EVTABLE FAILED

Explanation: (MVS and VM). EREP was unable to obtain virtual storage for the table of valid CPU serial numbers needed for the Event History report.

System Action: EREP terminates.

Programmer Response: Increase the region or virtual storage size and rerun the job.

IFC235I

GETVIS FAILED FOR EVTABLE

Explanation: (VSE). EREP was unable to obtain virtual storage for the table of valid CPU serial numbers needed for the Event History report.

System Action: EREP terminates.

Programmer Response: Increase the partition size and rerun the job.

IFC236I

GETMAIN FAILED FOR TREND TABLE PART 1

Explanation: (MVS and VM). EREP was unable to obtain virtual storage for the table needed to build Part 1 of the Trends report.

System Action: No more records are processed; EREP produces a partial report.

Programmer Response: Increase the region or virtual storage size and rerun the job.

IFC237I

GETVIS FAILED FOR TREND TABLE PART 1

Explanation: (VSE). EREP was unable to obtain virtual storage for the table needed to build Part 1 of the Trends report.

System Action: No more records are processed; EREP produces a partial report.

Programmer Response: Increase the partition size and rerun the job.

IFC238I

GETMAIN FAILED FOR PHYID TABLE

Explanation: **(MVS and VM).** EREP was unable to obtain virtual storage for the table of physical IDs.

System Action: Processing continues; this record is excluded from the report.

Programmer Response: Increase the region or virtual storage size and rerun the job.

IFC239I

GETVIS FAILED FOR PHYID TABLE

Explanation: **(VSE).** EREP was unable to obtain virtual storage for the table of physical IDs.

System Action: Processing continues; this record is excluded from the reports.

Programmer Response: Increase the partition size and rerun the job.

IFC240I

GETMAIN FAILED FOR ACLAS TABLE

Explanation: **(MVS and VM).** EREP was unable to obtain virtual storage for the additional-classification table used in building the System Summary and Trends reports.

System Action: Processing continues; EREP does no additional classification of this record.

Programmer Response: Increase the region or virtual storage size and rerun the job.

IFC241I

GETVIS FAILED FOR ACLAS TABLE

Explanation: (VSE). EREP was unable to obtain virtual storage for the additional-classification table used in building the System Summary and Trends reports.

System Action: Processing continues; EREP does no additional classification of this record.

Programmer Response: Increase partition size and rerun the job.

IFC242I

EXIT MOD ***mmmmmmmm*** **COULD NOT OBTAIN ERROR CLASS**

Explanation: (MVS, VM, and VSE). Either the named module could not load the PCT containing the product-dependent data for this record, or the PCT did not contain the expected error class.

System Action: Processing continues; this record is excluded from the report.

Programmer Response: There is an error either in the product-dependent exit module or in the product control table (PCT) for the product. Make sure EREP support is installed for the product(s) included in the module name.

IFC243I

EXIT MOD ***mmmmmmmm*** **COULD NOT OBTAIN PHYSICAL ID**

Explanation: (MVS, VM, and VSE). Either the named module could not load the PCT containing the product-dependent data for this record, or the PCT did not contain the expected physical ID.

System Action: Processing continues; this record is excluded from the report.

Programmer Response: There is an error either in the product-dependent exit module or in the product control table (PCT) for the product. Make sure EREP support is installed for the product(s) included in the module name.

IFC244I

EXIT MOD *mmmmmmmm* COULD NOT OBTAIN VOLID

Explanation: (MVS, VM, and VSE). Either the named module could not load the PCT containing the product-dependent data for this record, or the PCT did not contain the expected volume serial number.

System Action: Processing continues; this record is excluded from the report.

Programmer Response: There is an error either in the product-dependent exit module or in the product control table (PCT) for the product. Make sure EREP support is installed for the product(s) included in the module name.

IFC245I

EXIT MOD *mmmmmmmm* COULD NOT OBTAIN SYMCDE

Explanation: (MVS, VM, and VSE). Either the named module could not load the PCT containing the product-dependent data for this record, or the PCT did not contain the expected fault symptom code

System Action: Processing continues; this record is excluded from the report.

Programmer Response: There is an error either in the product-dependent exit module or in the product control table (PCT) for the product. Make sure EREP support is installed for the product(s) included in the module name.

IFC246I

EXIT MOD *mmmmmmmm* COULD NOT OBTAIN TERMINAL NAME

Explanation: (MVS, VM, and VSE). Either the named module could not load the PCT containing the product-dependent data for this record, or the PCT did not contain the expected terminal name.

System Action: Processing continues; this record is excluded from the report.

Programmer Response: There is an error either in the product-dependent exit module or in the product control table (PCT) for the product. Make sure EREP support is installed for the product(s) included in the module name.

IFC247I

EXIT MOD *mmmmmmmm* COULD NOT OBTAIN LIA/LIBADR

Explanation: (MVS, VM, and VSE). Either the named module could not load the PCT containing the product-dependent data for this record, or the PCT did not contain the expected line interface base address.

System Action: Processing continues; this record is excluded from the report.

Programmer Response: There is an error either in the product-dependent exit module or in the product control table (PCT) for the product. Make sure EREP support is installed for the product(s) included in the module name.

IFC248I

GETMAIN FAILED FOR SYSUM TABLE PART 1

Explanation: (MVS and VM). EREP was unable to obtain virtual storage for the table needed to build Part 1 of the System Summary.

System Action: No more records are processed; EREP produces a partial report.

Programmer Response: Increase region or virtual storage size and rerun the job.

IFC249I

GETVIS FAILED FOR SYSUM TABLE PART 1

Explanation: (VSE). EREP was unable to obtain virtual storage for the table needed to build Part 1 of the System Summary.

System Action: No more records are processed;.EREP produces a partial report.

Programmer Response: Increase partition size and rerun the job.

IFC250I

EXIT MOD *mmmmmmmm* COULD NOT OBTAIN SFT DATA

Explanation: (MVS, VM, and VSE). The named module supplies product-dependent data for the Event History report. It was unable to find the data for this software (SFT) record.

System Action: Processing continues; however, the entry for this record will not include the product-dependent data.

Programmer Response: There is an error either in the exit module or in the product control table (PCT) for the product. Make sure EREP support is installed for the product(s) included in the module name.

IFC251I

EXIT MOD *mmmmmmmm* COULD NOT OBTAIN OBR DATA

Explanation: (MVS, VM, and VSE). The named module supplies product-dependent data for the Event History report. It was unable to find the data for this OBR record.

System Action: Processing continues; however, the entry for this record will not include the product-dependent data.

Programmer Response: There is an error either in the exit module or in the product control table (PCT) for the product. Make sure EREP support is installed for the product(s) included in the module name.

IFC252I

EXIT MOD *mmmmmmmm* COULD NOT OBTAIN CCH DATA

Explanation: (MVS, VM, and VSE). The named module supplies product-dependent data for the Event History report. It was unable to find the data for this CCH record.

System Action: Processing continues; however, the entry for this record will not include the product-dependent data.

Programmer Response: There is an error either in the exit module or in the product control table (PCT) for the product. Make sure EREP support is installed for the product(s) included in the module name.

IFC253I

EXIT MOD *mmmmmmmm* COULD NOT OBTAIN MDRDASD DATA

Explanation: (MVS, VM, and VSE). The named module supplies product-dependent data for the Event History report. It was unable to find the DASD-specific data for this MDR record.

System Action: Processing continues; however, the entry for this record will not include the product-dependent data.

Programmer Response: There is an error either in the exit module or in the product control table (PCT) for the product. Make sure EREP support is installed for the product(s) included in the module name.

IFC256I

UNABLE TO LOAD MODULE ***mmmmmmmm*** **FOR MODULE IFCZIMGR**

Explanation: (MVS, VM, and VSE). During initialization of the EREP run, the named service module could not be found or loaded.

System Action: EREP terminates.

Programmer Response: Make sure the named module is included in the library being searched during initialization and try again to run EREP.

IFC257I

UNABLE TO INITIALIZE IFCZIMGR FOR ***mmmmmmmm***

Explanation: (MVS, VM, and VSE). EREP could not initialize its system interface manager (IFCZIMGR) for the named module. Either it could not load a needed service module or it could not open the TOURIST/SYSLST data set. The reason is indicated in the preceding message.

System Action: EREP terminates.

Programmer Response: Take the action recommended for the preceding message and try again.

IFC258I

EXIT MOD ***mmmmmmmm*** **COULD NOT FORMAT REPORT FOR** ***ssrr***

Explanation: (MVS, VM, and VSE). The named module produces the product-dependent Detail Summary Report. It was unable to produce the report for this SCP (*ss*) and record type (*rr*). The record type is byte 0 of the record. For a description of the various record types see Figure 10-6 on page 10-12. The SCP is one of the following:

- VM
- VE (VSE)
- V2 (MVS)

System Action: Processing continues; however, the Detail Summary Report for this SCP and record type will not be produced.

Programmer Response: There is an error either in the exit module or in the product control table (PCT) for the product. Make sure EREP support is installed for the product(s) included in the module name.

IFC259I

EXIT MOD ***mmmmmmmm*** **COULD NOT OBTAIN DATA FOR** ***ssrr***

Explanation: (MVS, VM, and VSE). The named module supplies product-dependent data for the Event History Report. It was unable to produce the report for this SCP (*ss*) and record type (*rr*). The record type is byte 0 of the record. For a description of the various record types see Figure 10-6 on page 10-12. The SCP is one of the following:

- VM
- VE (VSE)
- V2 (MVS)

System Action: Processing continues; however, the entry for this record will not include the product-dependent data.

Programmer Response: There is an error either in the exit module or in the product control table (PCT) for the product. Make sure EREP support is installed for the product(s) included in the module name.

IFC260I

USER EXIT MOD ***mmmmmmmm*** **COULD NOT BE LOADED FOR EREP**

Explanation: (MVS, VM, and VSE). The named module supplies product-dependent data for the Event History Report. EREP was unable to load it.

System Action: Processing continues; however, the entry for this record will not include the product-dependent data.

Programmer Response: There is an error in the product control table (PCT) for the product. Make sure EREP support is installed for the product(s) included in the module name.

Introduction to DMS-prefixed EREP Messages

The CPEREP command processor can issue messages to the VM user that have to do with CPEREP rather than EREP itself. Because CPEREP runs under the VM/SP Conversational Monitor System (CMS), its messages carry the DMS prefix in addition to EREP's IFC.[2]

The following table summarizes the source and format of the DMS-prefixed EREP messages as presented in this book:

Component prefix:	DMSIFC[2]
Program producing messages:	CPEREP (CMS)
Where produced:	On the output device; usually, the display terminal screen.
Audience:	System programmer
Message format:	**DMSIFC***nnn***E|I|S|W** *text*

nnn is the message serial number.

E indicates an error message

I indicates an informational message

S indicates a severe error

W indicates a warning message

text is the text of the message.

2 Message 830E is prefixed by DMSREA because it is issued by DMSREA, the CPEREP read module, instead of DMSIFC.

CPEREP Messages for VM Users

DMSIFC002E

[INPUT|OVERLAY] {FILE(S)|DATA SET|NOTE} ***fn ft fm*** **NOT FOUND**

Explanation: The specified file was not found on the accessed disk(s). Either:

- the file does not reside on this disk, or
- the file was identified incorrectly, or
- the system disk was not accessed as a read-only extension of the A-disk.

See the *VM/SP CMS Command and Macro Reference*, and "Defining Files for CPEREP" in this book for descriptions of the file identification required and the search procedure used.

Return Code: 28

System Action: Execution halts. System status remains the same.

User Response: Find or create the desired file. To make sure that the file exists, issue `STATE fn ft *` or `LISTFILE fn ft *`. Correct and reissue the command.

DMSIFC007E

FILE ***fn ft fm*** **[IS] NOT FIXED, 80 CHAR. RECORDS**

Explanation: The specified file must have fixed-length, 80-character records in order for the command to be executed.

Return Code: 32

System Action: Execution halts. System status remains the same.

User Response: It is possible that an incorrect fileID was specified in the command line. In this case, reissue the command. If, however, the fileID was correct but the file is in the wrong format or does not contain 80-character records, change the file's format and/or record length with the `COPYFILE` or `EDIT` command.

DMSIFC023E

NO FILETYPE SPECIFIED

Explanation: When CPEREP operands are in a separate file, you must include both the filename and the filetype on the command.

Return Code: 24

System Action: Execution halts. System status remains the same.

User Response: Reenter the command, specifying the filename and filetype.

DMSIFC070E

INVALID {PARAMETER *parameter*|**ARGUMENT** *argument*}

Explanation: An invalid operand was specified for CPEREP. If the operands are in a separate file, there may be too many on a line.

Return Code: 24

System Action: Execution halts. System status remains the same.

User Response: Correct the operand and reissue the CPEREP command.

DMSIFC104S

ERROR *xx* **READING FILE** *fn ft fm* **FROM DISK**

Explanation: An unrecoverable error occurred while reading the file from disk. *xx* indicates the nature of the error, and can be one of many codes. See *VM/SP System Messages and Codes* for the possible codes and their meanings.

Return Code: 100 or 1*xx*; same as the code in the message.

System Action: Execution halts. The system status remains the same as before the command was issued.

User Response: Retry the command. If the problem persists, call your system support personnel.

If the code is a 3 (permanent disk read error), it could be the result of your having detached a virtual disk without releasing it. CMS assumed the disk is still active and encountered an error when it tried to read the file.

DMSIFC825E

***CLEAR* IS VALID ONLY WHEN SPECIFIED BY ITSELF**

Explanation: CLEAR or CLEARF was specified along with other parameters. This is prohibited. The CLEAR parameter must be specified by itself, with no reports requested.

Return Code: 12

System Action: Execution halts. System status remains the same. No clearing takes place. No report is printed.

User Response: If you want the report, reissue the CPEREP command requesting the report without the CLEAR parameter. Include the ZERO parameter to clear the error-recording area after the report is completed. If you want only to clear the ERDS, reissue CPEREP specifying only the CLEAR/CLEARF operand. (However, see message DMSIFC829W.)

DMSIFC826E

EREP TXTLIBS NOT FOUND

Explanation: In attempting to search the EREP TXTLIBs, DMSIFC found that the pointer to the first TXTLIB contained zeros.

Return Code: 56

System Action: Execution halts. System status remains the same.

User Response: Issue a GLOBAL TXTLIB command listing the applicable EREP TXTLIBs in the proper search order. If no local libraries exist, the command should be:

```
GLOBAL TXTLIB ERPTFLIB EREPLIB
```

Reissue the CPEREP command. If the problem persists, call your system support personnel.

DMSIFC828I

CPEREP ZERO OR CLEAR HAS BEEN COMPLETED

Explanation: CLEAR/CLEARF or ZERO was specified by the user, or other parameters caused ZERO to be requested by default. The VM error-recording cylinders have been erased. If CLEARF was specified, the 303X MCH and CCH frame records were updated.

Return Code: 0

System Action: Control returns to CMS.

User Response: None required.

DMSIFC829W

ATTEMPTED 'ZERO' WAS SUPPRESSED. REQUIRES PRIVILEGE CLASS F

Explanation: CLEAR or ZERO was specified by the user, or other parameters caused ZERO to be requested by default. The VM/370 error-recording cylinders were not erased because the user was not authorized to do so. Only class F users can erase the error-recording area.

Return Code: 88 or 0

System Action: If the CLEAR function failed, the return code is 88. If the ZERO function failed, the return code will be 0. Reports (if requested) have already been generated. Control returns to CMS.

User Response: None required if ZERO was requested by mistake or default. If you need to erase the error-recording cylinders, see your system support personnel to get a class F directory entry.

DMSIFC831E

MORE THAN 100 CHARS OF OPTIONS SPECIFIED

Explanation: The maximum number of characters that can be used to specify CPEREP operands is 100. More than 100 characters were used.

Return Code: 62

System Action: Execution halts. System status remains the same.

User Response: Check the valid command options. Reissue the command using fewer than 100 characters to specify the options.

DMSIFC832S

SOFTWARE INCOMPATIBILITY AT THE CPEREP-EREP INTERFACE. CODE=*xxx*

Explanation: CPEREP is MVS EREP running under CMS with CPEREP providing interface code between MVS EREP and CMS. Some change has been made to MVS EREP (via PTF, or a new release) that has made it incompatible with the interface provided by CPEREP. *xxx* is one of the following reason codes:

Code	*Meaning*
001	An EXCP was attempted with a DCB other than that of the SYS1.LOGREC data set.
002	MVS EREP is expected to use only one IOB and one channel program when it uses EXCP to access the SYS1.LOGREC data set. But it has attempted to use IOBs or channel programs at more than one location in storage.
003	The expected read/write command in the channel program for accessing SYS1.LOGREC contains an unexpected op code.
004	While reading error records (with EXCP) from (simulated) SYS1.LOGREC, MVS EREP made an attempt to read nonsequentially prior to completion of the sequential reading phase.
005	An attempt was made to read record 2 of SYS1.LOGREC (the time stamp record), which CPEREP does not simulate.
006	The first EXCP to SYS1.LOGREC was not the expected read of the SYS1.LOGREC header record.
007	The channel program for accessing SYS1.LOGREC does not have the expected format.
008	An invalid disk address (CCHHR) was used while attempting to access SYS1.LOGREC.
009	There are no error records and yet MVS EREP attempted to read error records.
010	An invalid record length was encountered while reading SYS1.LOGREC. This may be due to error records being overlaid on the error cylinders.

Return Code: 104

System Action: CPEREP terminates with EREP message(s) IFC135I or IFC149I.

User Response: Reissue the command, or have your system programmer try it. If the problem persists, call your system support personnel.

DMSREA830E

I/O ERROR READING A BLOCK OF RECORDS FROM THE ERROR RECORDING CYLINDERS

Explanation: DMSREA, the CPEREP read module, encountered a permanent input/output error while attempting to read a 4K block of records from the error-recording area. Probable hardware error.

Return Code: 60

System Action: Execution halts. System status remains the same.

User Response: Execute the DDR service program to obtain a dump of the error recording cylinder on which the input error occurred. Reconstruct the data on the error-recording cylinders. If the reconstruction process is successful, try the CPEREP operation again. If the error recurs, call your system support personnel.

Problem Determination Aids

Sometimes, you must go through the process of problem determination in order to identify a failing hardware unit or program and determine who is responsible for fixing it. This section of the *EREP User's Guide* presents three different aids to problem determination:

1. the EREP return codes,
2. general problem determination tables and
3. the EREP DEBUG parameter.

EREP Return Codes

In addition to the IFC**** messages, EREP issues the following return codes when it stops processing:

(Decimal) Return Code	***Meaning***
00	No errors
04	Warning
08	Severe error (non-terminating)
10	Severe error (non-terminating)
12	Severe error (terminating)
16	Catastrophic error

A return code of 12 or greater means that EREP has terminated abnormally; it cannot complete the report. With a return code of 04 processing continues; the report will be complete but might not contain all possible records. With return codes 08 and 10, processing may or may not continue, depending on the kind of error EREP has encountered. If processing does continue, the report will likely be incomplete.

EREP (IFCEREP1) issues at least one IFC**** message for every return code greater than 04; it also issues messages for some situations that produce return codes of 04. The messages could appear in the TOURIST output or in the body of the report output.

Problem Determination Tables

Problem determination includes using procedures specified by IBM to arrive at the probable cause of an error that resulted in a message; it is part of the system programmer's response to the message.

Many of the descriptions of messages in this book are followed by the numbers of tables and items (**Problem Determination:** Table I, items 2, 4, 29, for example). The numbers correspond to the tables, and items in them, that appear on the following pages.

The actions prescribed in the problem determination tables are those recommended to diagnose problems with an SCP; they may not all be applicable to your problem. However, they will help you document the problem in case you have to call IBM for help in fixing it.

Table I and Table II are meant for MVS users; they include standard problem determination procedures for an MVS system, and the use of the Generalized Trace Facility (GTF) to isolate and document an error. Table III presents the standard problem determination procedures for VSE systems.

VM users can perform problem determination online, using the VM control program to investigate the virtual machine's operating system, whether it be MVS or VSE. The version of the VM *OLTSEP and Error Recording Guide* that applies to your VM facility contains more information about using VM to diagnose problems.

TABLE I. STANDARD PROBLEM DETERMINATION FOR MVS SYSTEMS

1. If MSGLEVEL=(1,1) was not specified in the JOB statement, specify it and rerun the job.

2. Save the console sheet from the primary console. In systems with remote consoles, save the remote console sheet. In systems with Multiple Console Support (MCS), save a copy of the hard-copy log.

3. Save the jobstream associated with the job.

4. Save the system output (SYSOUT) associated with the job.

5. Make sure the failing job step includes a:

   a. SYSABEND DD statement.
   b. SYSUDUMP DD statement.
   c. PL1DUMP DD statement.
   d. SYSMDUMP DD statement.

6. Make sure the PARM parameter of the EXEC statement specifies the following:

   a. MAP
   b. LIST
   c. DIAG
   d. MSG-AP
   e. CORE (if applicable)
   f. XREF
   g. DUMP

7. If SMP is used to make all changes to the system, execute the LIST CDS and LIST PTFBY functions of SMP to obtain a list of the current maintenance from the SMP control data set (CDS). If any changes are made to the system without using SMP, execute the LISDIR function of the BLIST (VS2), or HMBLIST (VS1) service aid program to obtain a list of all members with a PTF or local fix, and save the output. Execute the program against the:

   a. SYS1.LINKLIB data set.
   b. SYS1.SVCLIB data set.
   c. Library containing the program that issued the message.
   d. SYS1.LPALIB data set.

8. Execute the IMCJOBJQD (stand-alone) or IMCOSJQD (system-assisted) service aid program to obtain a formatted copy of the contents of the SYS1.SYSJOBQUE or SYS1.SYSWADS data sets, SWADS or the resident job list.

9. Execute the BLIST (VS2), or HMBLIST (VS1) service aid program to obtain:

   a. an object module listing, specifying the LISTOBJ function.
   b. a load module map and cross-reference listing, specifying the OUTPUT BOTH option of the LISTLOAD function.

10. Have a copy of the Message Control Program (MCP) available.

11. Execute the SADMP (VS2), or HMDSADMP (VS1) service aid program to dump the contents of real storage and page data sets to magnetic tape.

    After restarting the system, execute the appropriate function of the PRDMP (VS2), or HMDPRDMP (VS1) service aid program to print the required portion of the dump tape produced by SADMP.

    Save the tape from SADMP (VS2), or HMDSADMP (VS1), in case further information from the tape is required, and the listing from the PRDMP.

12. Execute the SEREP program, and save the resulting output.

13. Save all the associated output.

14. The normal response to this message is for the programmer/operator to execute a specific program. Save all output from that program.

15. Save the program listing associated with the job.

16. Save the dump.

17. Have the system generation (SYSGEN) output available from:

    a. Stage I
    b. Stage II

18. Execute IFCEREP1 to dump the SYS1.LOGREC data set and save the resulting output.

    For MSS, execute the following program to dump the SYS1.LOGREC data set:

    a. Service aid IFCISDA0.
    b. Program ISDASDA0 with the DETAIL (ALL) parameter.

19. Save the assembly listing associated with the job.

20. Save the control cards associated with the job.

21. Save the compiler output associated with the job.

22. Save the source input associated with the job.

23. Save the source program listing associated with the job.

24. Run OLTEP diagnostics for the problem device and save the output.

25. Execute the IEHLIST system utility program to obtain a list of the:

    a. volume table of contents of the associated volume, specifying the FORMAT option.
    b. volume table of contents of the associated volume, specifying the DUMP option.
    c. Directory of the associated data set.
    d. System catalog.

26. Execute the IEBPTPCH data set utility to print the:

    a. directory of the applicable data set.
    b. applicable data set.
    c. applicable member.
    d. applicable procedure.

27. Have the linkage editor/loader map available.

28. Save the associated volume.

29. Contact IBM for programming support.

30. Contact IBM for hardware support.

31. Save the trace output data set.

32. Print the GTF trace data set, using the PRDMP (VS2), or HMDPRDMP (VS1) service aid program with the EDIT statement.

33. Print the associated SVC dump data set, using the PRDMP (VS2), or HMDPRDMP (VS1) service aid with the GO statement.

34. Execute the Access Method Services LISTCAT command to:

    a. list the contents of the applicable catalog.
    b. list the catalog entries for the applicable objects and any related objects.

35. Execute the following access method services command:

    a. MSS LISTMSF for mountable volumes.
    b. MSS LISTMSF with the CARTRIDGES parameter.
    c. PRINT to list the contents of the MSVC (Mass Storage Volume Control) inventory data set.
    d. LISTMSVI.
    e. LISTMSF with the ALL parameter.

36. Execute the Access Methods Services PRINT command to print the repair workfile.

37. Execute the SPZAP service aid program using the ABSDUMP statement to print the contents of the applicable:

    a. data set.
    b. track.

38. Execute the Access Method Services AUDITMSS command with the following parameter:

    a. CHECK
    b. MAP
    c. READLABEL

39. Execute the Access Method Services CHECKMSS command.

40. Execute the Access Method Services COMPARED command.

41. Execute the Access Method Services DUMPMSS command to dump the following:

    a. Formatted Mass Storage Control storage.
    b. Mass Storage Control main storage.
    c. Mass Storage Control extended storage.
    d. Formatted Staging Adapter storage.
    e. Staging Adapter main storage.
    f. Staging Adapter extended storage.
    g. Mass Storage Control tables.

42. Save the latest output from the Mass Storage Control Table Create program.

43. Display units for units associated with the problem area. If the specific unit(s) is now known, display the range of all virtual units. See your configuration path chart for address ranges.

44. Obtain the RACF profile of the associated data set, where applicable.

45. Stop one CPU and use the hardware ALTER/DISPLAY facility to display:

    a. all general-purpose registers
    b. the PSW
    c. main storage locations 0 through 200 (hexadecimal) and 7000 through 7080 (hexadecimal).

46. If the SADMP program resides on tape, save the tape. If the SADMP program resides on disk, use the DUMP feature of IEHDASDR to print the SYS1.PAGEDUMP data set and cylinder 0 track 0 of this residence disk.

47. Save the output (listings) of the stage 1 and stage 2 SADMP initialization jobs.

48. Follow the procedures for item 9b on page 11-57 of this table for load modules:

    a. A/HMDSAPGE
    b. A/HMDSAPRO
    c. A/HMDSALDR

    of SYS1.LINKLIB. Use IEBUPDTE or IEBPTPCH to print the A/HMDSADMP and A/MDSADM2 macros from SYS1.MACLIB.

49. Save the A/HMDSADMP dump output (tape or listing).

50. If the program seems to be looping, use the display PSW feature of the hardware ALTER/DISPLAY facility along with the hardware instruction Step facility to trace the loop, instruction by instruction.

51. If there is an error in the contents of a page data set dump, restart the system using a different page data set, then dump the original page data set using the DUMP feature of IEHDASDR.

52. Use IEBCOPY to unload SYS1.IMAGELIB to tape.

53. Have a list of RACF-defined entities available.

TABLE II. USING GTF FOR PROBLEM DETERMINATION

Format 1: Tracing Without Prompting for Event Keywords

Before reproducing the problem, have the system operator issue a START GTF command specifying tape output, MODE EXT and TIME=YES. In response to message AHHL100A (VS2), or HHL100A (VS1), he should type TRACE=*opt*, where *opt* is the trace option indicated for the particular message or code.

When data for the problem has been recorded, run the PRDMP (VS2), or HMDPRDMP (VS1) service aid program using the EDIT statement to format the trace output, specifying DDNAME (ddname of the trace data set).

Format 2: Tracing With Prompting for Event Keywords

Before reproducing the problem, have the system operator issue a START GTF command specifying tape output, MODE=EXT and TIME=YES. In response to the message AHHL100A (VS2), or HHL100A (VS1) he should specify the trace options indicated for the associated message or code within the text of his reply.

Then, in response to the message AHHL101A (VS2) or HHL101A (VS1), he should specify the event keywords also indicated with the associated message or code.

When data for the problem has been recorded, run the PRDMP (VS2), or HMDPRDMP (VS1) service aid program using the EDIT statement to format the trace output, specifying DDNAME (ddname of the trace data set).

Format 3: Specialized Tracing Action

Before reproducing the problem, have the system operator issue a START GTF command specifying tape output MODE=EXT and TIME=YES. In response to message AHHL100A (VS2), or HHL101A (VS1), he should type 'TRACE=SYS.USR'. The DD statement for a data set in error should specify DCB=DIAGNS=TRACE.

When data for the problem has been recorded, execute the EDIT function of PRDMP (VS2), or HMDPRDMP (VS1), specifying the options SYS and USR=FFF.

TABLE III. STANDARD PROBLEM DETERMINATION FOR VSE SYSTEMS

2. Save the console sheet from the operator console. In systems with a DOC, save a copy of the hard-copy log.

4. Save the system output (SYSLST) associated with the job.

13. Save all the associated output.

29. Contact IBM for programming support.

30. Contact IBM for hardware support.

Note: For a more complete description, refer to the Serviceability Aids and Debugging Procedures manual that applies to your system.

The EREP DEBUG Parameter

When you need to see the actual input to EREP — as is recommended in several messages — one way to look at the error records is to run EREP again, specifying one of the DEBUG parameter options. Other DEBUG options give access to the communication and data areas used by the modules that make up the EREP program, to help in diagnosing problems within EREP itself.

Note: You should undertake the debugging of the EREP program only under the direction of an IBM service representative. If you suspect a problem exists, your first action should be to call the IBM Service Center for your area.

Because this book is primarily for IBM customers, it includes only those DEBUG options available and recommended for customer use; your IBM service representative can advise you further, if necessary.

Syntax

The DEBUG parameter can be included in any EREP run. Its syntax is:

```
DEBUG=(nn[,nn] ...)
```

nn is the one- or two-digit decimal number assigned to an EREP DEBUG option.

Indicates: That EREP is to print as part of the report output the information indicated by the specified option(s).

Default: None.
Debugging information is not normally printed.

Coding: The same rules and conventions apply as for other EREP keyword parameters.

Parameter Conflicts: None.

Options

The following DEBUG options are available for customer use:

Option Number	*Meaning*
4	Print the name and compile date of all control modules. Print the start and stop times of each routine called by IFCEREP1. The information appears in the TOURIST output.
17	Print a hexadecimal dump of every record that passed filtering. The records appear in the Event History report, one following each normal data line.
45	Print a hexadecimal dump of a frame record from module IFCZFRME. The record appears only in an MCH or CCH Detail PRINT report.
49	Print a hexadecimal dump of all the frame records from module IFCZFST1. This option, too, is for MCH and CCH Detail PRINT reports only.

Chapter 12. Summary of Tables and Charts

The following pages contain miscellaneous charts and tables that summarize reference information about EREP in various ways. In a some cases, the charts in this chapter are duplicates of ones in Chapter 1; they are here so their information is readily available when you are using the book in reference mode.

Incorrect EREP Parameter Combinations

Figure 12-1 shows all the EREP record selection and processing parameters and indicates which are mutually exclusive. For example, ACC is valid with all other parameters except DEVSER; and ZERO is valid only with some processing parameters and MODE=ALL.

	ACC	CPU	CPUCUA	CUA	DATE	DEV	DEVSER	ERRORID	HIST	LIA/LIBADR	LINECT	MERGE	MOD	MODE	SHORT	SYMCDE	TABSIZE	TERMN	TIME	TYPE	VOLID	ZERO
ACC	NO						NO															
CPU		NO	NO										NO									NO
CPUCUA		NO	NO	NO			NO						NO									NO
CUA			NO	NO																		NO
DATE					NO																	NO
DEV						NO	1			2										3	4	NO
DEVSER	NO		NO			1	NO	NO		NO			NO		NO	NO		NO			6	NO
ERRORID							NO	NO														NO
HIST									NO			NO										NO
LIA/LIBADR						2	NO			NO						NO		NO			NO	NO
LINECT											NO											
MERGE									NO			NO										
MOD		NO	NO				NO						NO									NO
MODE														NO								5
SHORT							NO								NO							
SYMCDE							NO			NO						NO		NO			NO	NO
TABSIZE																	NO					
TERMN							NO			NO						NO		NO			NO	NO
TIME																			NO			NO
TYPE						3														NO		NO
VOLID						4	6			NO						NO		NO			NO	NO
ZERO		NO	NO	NO	NO	NO	NO	NO	NO	NO			NO	5		NO		NO	NO	NO	NO	NO

Figure 12-1. Incorrect EREP Selection/Processing Parameter Combinations

Notes:

1. *DEVSER is used for the Threshold Summary only, so the only valid devices are 3410, 3420, 8809 and 34XX.*

2. *LIA/LIBADR applies only to TP communication controllers, so the only valid devices are 3705 and 3725.*

3. *DEV is valid with only four record types: DDR (D), MIH (H), OBR (O) and MDR (M).*

4. *VOLID applies only to 33XX DASD and 34XX tape devices.*

5. *ZERO is valid if you code or default MODE=ALL.*

6. *In EREP 3.1 and later, this combination is meaningless but not invalid. In earlier versions of EREP, it is considered a parameter conflict.*

Error Records and the EREP Reports

The following figure shows the various record types processed by EREP, and the reports that use those record types. Note that there is some overlap among device type groups and record types.

REPORT	*RECORD TYPES*	*SOURCE*
System Summary	CCH/CRW/SLH IPL Termination (EOD) MCH MDR OBR SFT	Processors, channels SCP SCP Processors I/O devices, including SCUs, controllers MVS operating system
System Exception Reports: System Error Summary Parts 1 and 2 Subsystem Exception DASD Summaries Tape Summaries	 IPL/EOD MCH CCH DDR OBR CCH MCH OBR MDR A OBR MDR A OBR MDR	 SCP Processors Processors, channels I/O devices, including: channel SCU controller device volume Processors, channels Processors 33XX DASD, 34XX Tape SCUs, controllers, devices Controllers, devices
Trends Report	IPL CCH/CRW/SLH Software (SFT) OBR MDR	SCP Processors, channels SCP I/O devices
Event History	CCH/CRW/SLH DDR EOD IPL MCH MDR MIH OBR SFT A	All hardware, plus software errors and events
Threshold Summary	MDR OBR (including demounts)	 3410, 3420, 8809 tape devices
PRINT Reports	All but DASD MDR	All products/devices.

Figure 12-2. The Records in EREP Reports

Report Parameter Summary

The following figure contains the same information as Figure 1-5 in Chapter 1, "Introduction to EREP."

REPORT PARAMETERS	*WHAT THEY DO*
EVENT	Produces an Event History report, a chronological presentation of one-line abstracts from the selected records. It has two parts.
PRINT	Produces a series of Detail Edit and/or Summary reports for the selected record types. The number of reports depends on the input and selection parameters. *Note: PRINT is the default report parameter. The only way to run EREP without producing any report output is to code PRINT=NO.*
SYSEXN	Produces a System Exception report series, reports covering processors, channels, DASD and tape subsystems.
SYSUM	Produces a System Summary, a condensed two-part report of all errors for the principal system elements - CPU, channels, storage, SCP, I/O Subsystem.
THRESHOLD	Produces a summary of a tape subsystem, including media statistics and permanent errors that exceed the limits set on the parameter itself.
TRENDS	Produces an expanded version of the System Summary, presenting error records logged by or for the various system elements during 30 days, at most. Like the System Summary, it has two parts.

Figure 12-3. Report Parameters for EREP

Selection Parameter Summary

The following figure contains the same information as Figure 1-6 in Chapter 1, "Introduction to EREP."

SELECTION PARAMETERS	*WHAT THEY DO*
CPU	(Processor serial and machine type numbers). Tells EREP to use only the records associated with this particular processor.
CPUCUA	(Processor serial number and device address). Tells EREP to use only the records associated with this device attached to this processor.
CUA	(Device address or number). Tells EREP to use only the records associated with this particular device address or device number.
DATE	Tells EREP to use only the records created during this date range.
DEV	(Device type). Tells EREP to use only the records associated with this particular device type; or, conversely, not to use the records associated with this device type.
DEVSER	(Device serial number). Tells EREP to use only the 34XX tape records associated with this tape device serial number.
ERRORID	(Error identifier). Tells EREP to use only the MVS software records containing this particular error identifier.
LIA/LIBADR	(Line interface [base] address). Tells EREP to use only the 3705 or 3725 communication controller records containing this line interface address.
MOD	(Processor model). Tells EREP to use only the records containing this processor machine type (number).
MODE	(370 or 370-XA). Tells EREP to use only the records created in this operating mode.
SYMCDE	(Fault symptom code). Tells EREP to use only the 33XX DASD records containing this particular fault symptom code.
TERMN	(Terminal name). Tells EREP to use only the VTAM or TCAM OBR records containing this terminal name.
TIME	Tells EREP to use only the records created during this time range.
TYPE	(Record type). Tells EREP to use only the records of the specified type(s).
VOLID	(Volume serial number). Tells EREP to use only the 33XX DASD or 34XX tape records containing this volume serial number.

Figure 12-4. Selection Parameters for EREP

Processing Parameter Summary

The following figure contains the same information as Figure 1-7 in Chapter 1, "Introduction to EREP."

PROCESSING PARAMETERS	*WHAT THEY DO*
ACC	(Accumulate). Tells EREP to copy the records used for the report onto an output history data set.
HIST	(History). Tells EREP that its input consists of records on a history data set.
LINECT	(Line count). Tells EREP that each page of the report output is to contain this number of lines.
MERGE	(Merge). Tells EREP that its input consists of records from both the ERDS and a history data set.
SHORT	(Short OBR). Tells EREP to print out short-form OBR records in Detail Edit report output.
TABSIZE	(Table size). Tells EREP that the sort table it uses for internal processing is to be this big.
ZERO	(Zero ERDS). When this report is complete, EREP is to clear the error-recording data set (SYSREC, or SYS1.LOGREC, or the error recording area).

Figure 12-5. Processing Parameters for EREP

Control Statement Summary

The following figure contains the same information as Figure 1-8 in Chapter 1, "Introduction to EREP."

CONTROL STATEMENTS	*WHAT THEY DO*
CONTROLLER	Tells EREP to combine the error records associated with this particular control unit and its attached devices.
DASDID	Tells EREP that this is the configuration of the 33XX DASDs within each subsystem; identifies those that do not provide physical IDs for the System Exception report series. This control statement applies only to the System Exception Report series.
ENDPARM	Tells EREP that this is the end of the instream EREP parameters; the instream data that follows consists of EREP control statements.
LIMIT	Tells EREP not to produce any output for the System Exception reports when the number of errors is below the values specified here. This control statement applies only to the System Exception Report series.
SHARE	Tells EREP to combine the records for these devices that are shared between systems. This control statement applies to all the reports that generate I/O device summaries.

Figure 12-6. Control Statements for EREP

Default Values and Actions for EREP Parameters

When you do not include a parameter in the controls for an EREP run, EREP uses a default value for that parameter. Figure 12-7 on page 12-10 lists the default values EREP uses for each of the parameters.

If you run EREP using *no controls at all*, the following happens:

- EREP produces Detail Summary (and Data Reduction, if your installation includes 3370 DASD) reports of all the records on the ERDS.
- The reports do not combine the records from shared I/O devices, nor do they identify the records as being from shared devices.
- EREP writes the records to a history data set if one is available to receive them; if none is available, EREP issues an error message and the job or step abends.

PARAMETER	*DEFAULT VALUE*
ACC	Yes, except with THRESHOLD; then, it's No.
CPU	All. EREP uses the records from all processors.
CPUCUA	All. EREP uses the records regardless of the CPU or CUA.
CUA	All. EREP uses the records from all device addresses.
DATE	All. EREP uses all the records in the input data set, regardless of when they were created. See the TRENDS parameter in Chapter 7, "EREP Parameters" for an exception to this rule.
DEV	All. EREP uses the records from all the valid IBM device types in your installation.
DEVSER	All. EREP uses records for the Threshold Summary regardless of the device serial numbers they contain.
ERRORID	All. EREP uses MCH and MVS software records regardless of the ERRORIDs they contain.
HIST	No. EREP does not expect to find the records on a history data set.
LIA/LIBADR	All. EREP uses 3705 and 3725 TP communication controller records regardless of the line interface base address they contain.
LINECT	For MVS and VM, 50 lines per page; for VSE systems, the default is the number of lines per page set for SYSLST.
MERGE	No. EREP does not expect to merge the records from a history data set with those on the ERDS.
MOD	All. EREP uses records from all S/370 processors, regardless of their model types.
MODE	All. EREP uses all available records, regardless of whether they were recorded in 370 or 370-XA mode.
SHORT	No. EREP does not print out short OBR records for Detail Edit reports. It does print them out for Detail Summaries, however.
SYMCDE	All. EREP uses all OBR records, regardless of the fault symptom codes they contain.
TABSIZE	For MVS and VM, EREP's internal sort table is 24K bytes; for VSE systems, it is 4K bytes.
TERMN	All. EREP uses OBR records from TCAM- or VTAM-supported TP devices regardless of the terminal names they contain.
TIME	All. EREP uses all available records, regardless of the time they were created.
TYPE	All. EREP uses all types of records.
VOLID	All. EREP uses certain DASD and tape records regardless of the associated volume serial numbers.
ZERO	No. EREP does not clear the ERDS after completing the report. However, ZERO is tricky; see the ZERO parameter in Chapter 7, "EREP Parameters."

Figure 12-7. Default Values and Actions for EREP Parameters

DASD Storage Capacities

When allocating DASD storage for the work (DIRECTWK) data set, you need to know how many records a given device can hold. This table contains that information.

See *Data Management Services* (MVS/370 and MVS/XA) to compute DASD track and cylinder capacities for blocked records.

Device Type	Average Records per Track [1,4]	Average Records per Cylinder [1,4]	MCH Records per Track [2]	MCH Records per Cylinder [2]
2314 2319	29	580	4	81
2305 Mod. 1	56	448	7	63
2305 Mod. 2	58	464	8	64
3310 [3]				
3330 3330 Mod. II	52	988	7	139
3340 3344	33	401	4	48
3350	76	2,291	10	300
3370 [3]				
3375	78	936	18	216
3380	81	1,281	22	352

Figure 12-8. Storage Capacities of IBM DASD

Notes:

1. *Average record size, except for MCH records, is 70 bytes plus the inter-record gap.*

2. *Average MCH record size is 1600 bytes plus the inter-record gap.*

3. *Fixed-block devices, with 512 bytes per block, can hold 7.3 average records per block. Each MCH record requires 3.125 blocks.*

4. *Software records are much larger than average, because they include the SDWA.*

TOURIST Output: Messages and Controls

Figure 8-4, in "DASDID Control Statement," is one example of the TOURIST output. Figure 12-9 is the TOURIST output from an example of the prompting method of entering CPEREP operands ("Prompting Method" on page 9-10). It shows the TOURIST messages as they appear on the terminal screen.

```
INPUT PARAMETER STRING              PRINT=PS,DEV=(3420)

PARAMETER OPTIONS VALID FOR THIS EXECUTION
    RECORD TYPES (MCH,CCH,OBR,SOFT,IPL,DDR,MIH,EOD,MDR),
    PRINT(EDIT,SUMMARY),ACCUMULATE,LOGREC INPUT,
    DUMP SDR COUNTERS
    DATE/TIME RANGE - ALL
    TABLE SIZE - 024K, LINE COUNT - 050
DEVICE ENTRIES
    DEVICE TYPES (OBR,MIH,DDR) -3420(8005)
    DEVICE TYPES (MDR) -3420(**)
IFC120I       6   RECORDS THAT PASSED FILTERING

OBR RECORDS REQUESTED BUT NOT FOUND
SFT RECORDS REQUESTED BUT NOT FOUND
IPL RECORDS REQUESTED BUT NOT FOUND
DDR RECORDS REQUESTED BUT NOT FOUND
MIH RECORDS REQUESTED BUT NOT FOUND
EOD RECORDS REQUESTED BUT NOT FOUND

NUMBER OF MCH TYPE OF RECORDS READ WAS     1
NUMBER OF CCH TYPE OF RECORDS READ WAS     1
NUMBER OF MDR TYPE OF RECORDS READ WAS     4
```

Figure 12-9. TOURIST Output From a CPEREP Run

Device Type Codes in the OBR Record

At offset 52 (X'34') in the long outboard (unit-check) record[1] is a four-byte field that contains a code for the device class and type of the failing device. Called "OBRCODE" in this book, the field contains data gathered from different sources for different operating systems:

- For VSE systems, only the first two bytes are significant. They are the contents of the fields at offset 4 and 5 in the physical unit block (PUB) for the device. The field names are PUBDEVTY and PUBOPTN.
- For MVS systems, it is the second two bytes that are meaningful. They are bytes 18 and 19 (decimal) of the unit control block (UCB) for the device:
  - 18 (UCBTBYT3) is a code (bit mask) for the device **class**
  - 19 (UCBTBYT4) is a code for the device **type**.
- When VM writes an OBR record, it takes the device type code from either the PUB or the UCB, depending on which operating system is running in the virtual machine.

1 In the short form of the record, the field is at offset 24 (X'18'). The code is also in the CCH record, at offset 68 (X'44'), and in the SLH record, at offset 40 (X'28').

OBR Codes

The following tables give the OBR device class/type codes (also called the OBR codes) from the UCB as they appear to EREP. *The OBR codes are to the **left** of the equal signs.*

OBR Codes, Sorted by Code

0801 = 2540DD	0828 = 2956	1005 = 2280	4004 = 2955	4100 = CTCA
0802 = 2540DD	0829 = 1419	1006 = 2282	4005 = 3705	4201 = 1030
0803 = 1442	082A = 1275	1007 = 3278	4006 = 3705	4202 = 1050
0804 = 2501	082B = 1275	1008 = 3066	4009 = 3704	4203 = 1060
0805 = 2520	082C = 1275	1009 = 327D	400A = 3968	4204 = 2740
0806 = 3505	082D = 1419	100A = 3284	4011 = 2702	4205 = 2740
0807 = 3525	082E = 1419	100B = 3286	4013 = 2703	4206 = 2741
0808 = 1403	082F = 2495	100C = 3158	4014 = 7772	4207 = 226T
0809 = 3211	0830 = 3213	100D = 3036	4015 = 3705	4208 = 105T
080A = 1443	0831 = 1017	100E = 3138	4021 = 2702	4209 = 2760
080B = 3203	0832 = 1018	100F = 3148	4022 = 2701	420A = 83B3
080C = 3525	0833 = 3210	1013 = 5080	4023 = 2703	420B = 115A
080D = 3262	0834 = 3215	1014 = BA00	4025 = 3705	420F = 1130
080E = 3800-1	0835 = 1255	2001 = 2311	4031 = 2702	4210 = 2020
0810 = 2671	0836 = 1255	2002 = 2301	4032 = 2701	4211 = 2780
0811 = 4245	0837 = 1270	2003 = 2303	4033 = 2703	4212 = 2770
0812 = 1012	0838 = 1270	2005 = 2321	4035 = 3705	4213 = 2265
0813 = 4248	0839 = 2596	2006 = 2305	4041 = 2702	4214 = 2770
0814 = 2947	083D = 7443	2007 = 2305	4042 = 2701	4214 = 2930
0816 = 3890	0840 = 3890	2008 = 2314	4043 = 2703	4215 = 2972
0817 = 3886	0841 = 3886	2009 = 3330	4045 = 1060	4216 = 327T
0818 = 2495	0842 = 3850	200A = 3340	4051 = 2702	4217 = 2970
0819 = 3895	0844 = 3540	200B = 3350	4052 = 2701	4218 = 3735
081A = 1285	0846 = 2560	200C = 3375	4053 = 2703	4219 = 3945
081B = 1287	0847 = 3504	200D = 3330	4061 = 2702	421A = 2790
081C = 1288	0848 = 5425	200E = 3380	4062 = 2701	421B = 3670
081D = 1419	0849 = 3203	201E = 3380	4063 = 2703	4420 = 3700
081E = 1419	084C = 3838	202E = 3380	4071 = 2702	8001 = 2400
081F = 1275	084D = 5203	2101 = 3310	4072 = 2701	8003 = 3400
0820 = 1052	084E = 5203	2102 = 3370	4073 = 2703	8004 = 3420
0821 = 2150	0880 = 5424	2105 = 3370	4081 = 2702	8005 = 3410
0822 = 3210	0882 = 3848	2106 = 9335	4082 = 2701	8006 = 8809
0823 = 3215	08A0 = 3800-3	2107 = 9332	4083 = 2703	8007 = 3430
0824 = 2956	1001 = 1015	4000 = 7770	4091 = 2702	8008 = 7340
0825 = 2956	1002 = 2250	4001 = 2702	4092 = 2701	8009 = 9347
0826 = 2956	1003 = 226D	4002 = 2701	4093 = 2703	800A = 3422
0827 = 2956	1004 = 105D	4003 = 2703	40F1 = 3791	8080 = 3480

OBR Codes, Sorted by Device Type

1014 = BA00	1003 = 226D	4061 = 2702	0849 = 3203	4420 = 3700
4100 = CTCA	4207 = 226T	4071 = 2702	080B = 3203	4009 = 3704
0812 = 1012	4213 = 2265	4081 = 2702	0822 = 3210	4005 = 3705
1001 = 1015	1005 = 2280	4091 = 2702	0833 = 3210	4006 = 3705
0831 = 1017	1006 = 2282	4003 = 2703	0809 = 3211	4015 = 3705
0832 = 1018	2002 = 2301	4013 = 2703	0830 = 3213	4025 = 3705
4201 = 1030	2003 = 2303	4023 = 2703	0823 = 3215	4035 = 3705
1004 = 105D	2006 = 2305	4033 = 2703	0834 = 3215	4218 = 3735
4208 = 105T	2007 = 2305	4043 = 2703	080D = 3262	40F1 = 3791
4202 = 1050	2001 = 2311	4053 = 2703	1009 = 327D	080E = 3800-1
0820 = 1052	2008 = 2314	4063 = 2703	4216 = 327T	08A0 = 3800-3
4045 = 1060	2005 = 2321	4073 = 2703	1007 = 3278	084C = 3838
4203 = 1060	8001 = 2400	4083 = 2703	100A = 3284	0882 = 3848
420F = 1130	0818 = 2495	4093 = 2703	100B = 3286	0842 = 3850
420B = 115A	082F = 2495	4204 = 2740	2101 = 3310	0817 = 3886
0835 = 1255	0804 = 2501	4205 = 2740	2009 = 3330	0841 = 3886
0836 = 1255	0805 = 2520	4206 = 2741	200D = 3330	0816 = 3890
0837 = 1270	0801 = 2540DD	4209 = 2760	200A = 3340	0840 = 3890
0838 = 1270	0802 = 2540DD	4212 = 2770	200B = 3350	0819 = 3895
081F = 1275	0846 = 2560	4214 = 2770	2102 = 3370	4219 = 3945
082A = 1275	0839 = 2596	4211 = 2780	2105 = 3370	400A = 3968
082B = 1275	0810 = 2671	421A = 2790	200C = 3375	0811 = 4245
082C = 1275	4002 = 2701	4214 = 2930	200E = 3380	0813 = 4248
081A = 1285	4022 = 2701	0814 = 2947	201E = 3380	1013 = 5080
081B = 1287	4032 = 2701	4004 = 2955	202E = 3380	084D = 5203
081C = 1288	4042 = 2701	0824 = 2956	8003 = 3400	084E = 5203
0808 = 1403	4052 = 2701	0825 = 2956	8005 = 3410	0880 = 5424
081D = 1419	4062 = 2701	0826 = 2956	8004 = 3420	0848 = 5425
081E = 1419	4072 = 2701	0827 = 2956	800A = 3422	8008 = 7340
082D = 1419	4082 = 2701	0828 = 2956	8007 = 3430	083D = 7443
082E = 1419	4092 = 2701	4217 = 2970	8080 = 3480	4000 = 7770
0829 = 1419	4001 = 2702	4215 = 2972	0847 = 3504	4014 = 7772
0803 = 1442	4011 = 2702	100D = 3036	0806 = 3505	420A = 83B3
080A = 1443	4021 = 2702	1008 = 3066	0807 = 3525	8006 = 8809
4210 = 2020	4031 = 2702	100E = 3138	080C = 3525	2107 = 9332
0821 = 2150	4041 = 2702	100F = 3148	0844 = 3540	2106 = 9335
1002 = 2250	4051 = 2702	100C = 3158	421B = 3670	8009 = 9347

Device Type Codes in the MDR Record

The following tables give the MDR codes — one byte at offset 4 (X'4') in the miscellaneous data record. *The MDR codes are to the **left** of the equal signs.*

MDR Codes, Sorted by Code

```
01 = 3330
02 = 2305 MOD 2
03 = 3277
03 = 3286
03 = 3284 (non-NCP mode)
04 = 3211
05 = (non-NCP mode)
06 = 3670
07 = 3168
08 = 2715
09 = 3340
09 = 3344
0A = 3330 MOD 11
0B = 3277
0C = 3800 MOD 1
0D = 3895
0E = 3850
0F = IGAR Diskette
10 = 3203
10 = 3289
11 = 3350
12 = 2305 MOD 1
13 = 3277 (NCP mode)
14 = 3380
15 = 3705 (NCP mode)
16 = 3310
17 = 3370 MOD 1
18 = 3375
1A = 3370 MOD 2
1B = 3380 MOD E
1C = 3380 MOD D
1D = 9335
1E = 9332
1F = 9347
20 = 3800 MOD 3,8
25 = 3725
40 = 8809
41 = 3480
F0 = 2946
F1 = 2948
F3 = 2703
```

MDR Codes, Sorted by Device Type

```
12 = 2305 MOD 1
02 = 2305 MOD 2
F3 = 2703
08 = 2715
F0 = 2946
F1 = 2948
07 = 3168
10 = 3203
04 = 3211
03 = 3277
0B = 3277
13 = 3277 (NCP mode)
03 = 3284 (non-NCP mode)
03 = 3286
10 = 3289
16 = 3310
01 = 3330
0A = 3330 MOD 11
09 = 3340
09 = 3344
11 = 3350
17 = 3370 MOD 1
1A = 3370 MOD 2
18 = 3375
14 = 3380
1B = 3380 MOD E
1C = 3380 MOD D
41 = 3480
06 = 3670
05 = 3705 (non-NCP mode)
15 = 3705 (NCP mode)
25 = 3725
0C = 3800 MOD 1
20 = 3800 MOD 3,8
0E = 3850
0D = 3895
40 = 8809
1E = 9332
1D = 9335
1F = 9347
0F = IGAR Diskette
```

Part 4. Product-Dependent Information

How to Use Part 4

This part of the *EREP User's Guide and Reference* contains information about how EREP works with specific hardware and software products:

- devices
- processors
- control programs

The information is arranged within IBM product groups, according to the specific product identifiers.

For example, if you want to see the special considerations for EREP controls for 34XX tape drives, you can find the information under 34XX in the subsection for magnetic tape devices. For specific information about an EREP report for a 3705 communications controller, you would look under 3705 in the TP Devices subsection.

The product subsections are in the following order:

1. Card Readers and Punches
2. Consoles and Displays
3. Direct-Access Storage Devices (DASD)
4. Diskette Devices
5. Magnetic Tape Drives
6. OCR/MICR Devices
7. Printers
8. Processors (CPUs)
9. Punched Tape Devices
10. Teleprocessing Devices
11. Other Devices
12. System Control Programs (SCP)

Within each product group are the following kinds of information:

- Special considerations for EREP reports
- Sample reports, for some devices
- Special considerations for EREP controls (parameters and control statements)
- Other considerations, if any
- A list of the valid specifications for the relevant EREP parameters
- Where possible, a bibliography of other books that can be helpful.

Note that this section of the book is somewhat dynamic; that is, it is designed so new pages can be added to the individual product subsections when new EREP device support is released.

This means that the product subsections might be superseded by newer information added under specific product numbers. Make sure the product-specific information is kept up to date, and look for new information under specific product numbers within each product subsection.

Section Table of Contents

Chapter 13. Card Readers and Punches

The information in this subsection could be changed or superseded by newer information added under specific device types. If you are interested in a new device or machine number, look for the specific number.

Special Considerations for EREP Reports

Useful reports for these devices are:

SYSUM
EVENT
TRENDS
PRINT=PT or PS with DEV=nnnn and TYPE=OH (OBR and MIH records)

Care should be taken when requesting reports other than these as the results could be misleading.

Some of the devices listed below may produce different record types. In that case, request that record type when requesting Detail Edit and Summary (PRINT) reports.

Special Considerations for EREP Controls

None.

Devices Supported by EREP

These card devices are valid for DEV=

1442 card reader/punch
2501 card reader
2520 card reader/punch
2540 card reader/punch
2560 multi-function card machine
2596 card reader/punch
3504 card reader
3505 card reader
3525 card punch
5424 multi-function card machine
5425 multi-function card machine

Chapter 14. Consoles and Displays

The information in this subsection could be changed or superseded by newer information added under specific device types. If you are interested in a new device or machine number, look under the specific number.

Special Considerations for EREP Reports

Useful reports for these devices are:

SYSUM
EVENT
TRENDS
PRINT=PT or PS with DEV=nnnn and TYPE=OTH (OBR, MDR and MIH records)

Care should be taken when requesting reports other than these as the results could be misleading.

Some of the devices listed below may produce different record types. In that case, request that record type when requesting Detail Edit and Summary (PRINT) reports.

Special Considerations for EREP Controls

None.

Devices Supported by EREP

These console and display devices are valid for DEV=

1015 display unit
1052 console
2020 console
2150 console
2250 display unit
2260 display station
2265 display station
3036 console
3066 console
3138 console
3148 console
3158 console
3168 console
3210 console printer/keyboard
3213 console printer
3215 console printer/keyboard
3277 display station (terminal)
3278 display station (terminal)
5080 graphics systems workstation

Note: Although the 3279 display terminal is not valid for the DEV parameter, EREP does process its records — as 3277 records.

Chapter 15. Direct-Access Storage Devices (DASD)

The information in this subsection could be changed or superseded by newer information added under specific device types. If you are interested in a new device or machine number, look under the specific number.

Special Considerations for EREP Reports

- Useful reports for these devices are:

  SYSUM
  SYSEXN (33XX only)
  EVENT
  TRENDS
  PRINT = AL with DEV = nnnn and TYPE = DOTH (DDR, OBR, MDR and MIH records)

  Care should be taken when requesting reports other than these as the results could be misleading.

  Some of the devices listed below may produce different record types. In that case, request that record type when requesting Detail Edit and Summary (PRINT) reports.

- The 2305 is not supported by the System Exception report; EREP produces a Detail Summary for this device. Also, EREP produces a Data Reduction report for the 3370 only. To separate the reports for these devices and for dedicated DASD from the rest of the Detail PRINT output for I/O devices, run the following step *before* running any Detail PRINT reports for other I/O devices:

  ```
  PRINT=SD
  DEV=(2305,3370)
  TYPE=OT
  ```

- Detailed information about EREP Reports for these specific devices may be found in this chapter under:
  - "33XX DASD"
  - "3380 DASD"
  - "3850 Mass Storage System"
  - "9332 and 9335 DASD"

Special Considerations for EREP Controls

- Detailed information about EREP Controls for these specific devices may be found in this chapter under:
  - "33XX DASD"
  - "3380 DASD"
  - "3850 Mass Storage System"
  - "9332 and 9335 DASD"

Devices Supported by EREP

These direct-access storage devices are valid for DEV=

2301 drum storage
2303 drum storage
2305 fixed head storage
2311 disk storage
2314 disk storage
2321 data cell drive
23XX
3310 disk storage
3330 disk storage
3340 disk storage facility
3344 disk storage
3350 disk storage
3370 direct access storage
3375 direct access storage
3380 direct access storage
9332 direct access storage
9335 direct access storage
33XX
3850 mass storage system
3851 mass storage facility

Note: 23XX and 33XX are general device type designations that include families of direct-access storage devices.

33XX DASD

Special Considerations for EREP Reports

System Exception Reports

- The 33XX subsystems are the only DASD subsystems included in the System Exception report series. The subsystems comprise:

  - 3310-3380 DASD drives
  - 9332 and 9335 DASD drives
  - 3830 and 3880 DASD control units

  Besides the other report types listed for DASD, SYSEXN is the most valuable for 33XX products. The System Exception series is meant to replace Detail Summaries for these devices.

- Virtual 33XX drives on 3850 mass storage systems are not included in the System Exception series.

- In the various summary reports of the System Exception series, 3375 and 3380 errors can be reported differently from those of other DASD. Because the 3375 and 3380 can have two controllers at the head of the string, the PHYSICAL ID field contains device or volume failures associated with the *lowest* CTLRID for the string on which the device resides.

- See "LIMIT Control Statement" in this subsection for more about DASD and the System Exception series.

All Reports

- Some 33XX DASD identify themselves to EREP via **physical IDs**; the identifiers assigned to the *storage control unit* (SCU), the *controller* and the *device* when the device was installed. Other 33XX DASD are identified by the physical and logical controller-unit addresses (*CUAs*). The sources of these different identifiers are as follows:

  - The **physical ID** is in the sense records created for 3375 and 3380 devices and 3880 storage control units (storage directors).

  - The **secondary control unit address (SCUA)** is in the OBR or MDR record. It is the logical address from which the sense data was received.

  - The **primary control unit address (PCUA)** is the address of the physical device via the base (primary) channel. For 33XX devices, this is the position of the drive in its string. Note that the PCUA is also the physical address for *all* demountable DASD.

Some EREP reports show 33XX DASD by physical ID. In those that do not, the address shown is the PCUA. See the discussion of the DASDID control statement in Chapter 8 for more information about the physical ID.

- The only records used for 33XX DASD are OBR (long) and MDR records. OBR records indicate errors or single incidents. MDR records contain statistical data collected at the storage control unit for usage, errors, and overruns.

- With the advent of the System Exception report series, the combined OBR/MDR Detail Summary (PRINT=PS|SD|SU,TYPE=OT) has been eliminated, as have the MDR Detail Edit and Summary reports. The DASD summaries included in the System Exception series contain more information than was in the OBR/MDR summary, and present more meaningful usage data than appeared in the MDR Detail reports.

- In the System Summary and Trends reports, 33XX devices providing physical IDs are listed by those IDs only; CPU identifiers are omitted.

Special Considerations for EREP Controls

- Devices having physical IDs do not require DASDID or SHARE statements.

- The reports that require SHARE statements for non-physical ID devices are System Summary, Trends, and Data Reduction (PRINT=DR or SD).

DASDID Control Statement

The DASDID control statement applies only to the System Exception report series and, by implication, only to 33XX DASD. Because the syntax and values for the DASDID statement are the same for all device types, the description and explanation of the DASDID statement are in Chapter 8 under "DASDID Control Statement."

However, there are some notes about the DASDID statement that are product-specific:

3880 and DASDID: The 3880 control unit supplies its own physical ID. If your installation includes 3880s, note those physical IDs before assigning any IDs to control units. The physical ID for each control unit should have been set with hardware switches at the time it was installed.

LIMIT Control Statement

The LIMIT control statement applies only to the System Exception report series. It works differently for each of the product groups it is used for; following is a description of the way you use it for 33XX DASD. ***It is not, however, valid for 9332 and 9335 devices.***

The LIMIT statement has the following format for 33XX DASD:

```
LIMIT dasd,keyword[,keyword][,keyword] . . .
```

dasd can be one of the following generic product types:

33XX
3310
3330
3340
3350
3370
3375
3380
3830
3880

Note: 3340 includes 3344.

33XX is the general device type designation for all the listed direct access devices and control units. When you code 33XX on a LIMIT statement, you are requesting that the limits apply to all devices of the general type.

You can set minimum thresholds for eight different kinds of temporary errors or events recorded against DASD, using the LIMIT keyword values listed here.

To Set Limits For:	***Use Keyword:***
Seek errors	SKS=nnnn
Read errors	RD=nnnn
Bus out parity errors	B=nnnn
Equipment checks	EQUCHK=nnnn
Check data	C=nnnn
Invoked offsets	I=nnnn
Diskette checks	D=nnnn
Overruns	OVRN=nnnn
All not otherwise specified	ALL=nnnn

"nnnn" can range from 1 to 9999; it requires no leading zeros.

Be aware that:

1. *If you do not specify a number for "nnnn," EREP uses a default value of 01, applying no limits to temporary errors on DASD. In this case, the line in the DASD Subsystem Exception report showing the CURRENT LIMITS contains 00 for that keyword, and all errors of that type are included in the Subsystem Exception report.*

2. *When you set limits on temporary errors for DASD, EREP excludes from the Subsystem Exception report those errors that do not equal or exceed the LIMIT statement values. For example, if you code*

```
LIMIT 3830,EQUCHK=5,OVRN=10
```

the DASD Subsystem Exception report will show temporary equipment checks and overrun errors for a 3830 control unit only if there are 5 or more equipment checks and/or 10 or more overruns recorded against the device.

3. *Not all the keywords are valid for every product type. The table on page 15-7 shows the valid error type keywords for each of the 33XX DASD products.*

4. *When you specify 33XX or ALL on a LIMIT statement, EREP uses only the valid keywords for each device type included.*

5. *Use of the LIMIT control statement is invalid for 9332 and 9335 devices.*

6. *EREP ignores the ALL values on any LIMIT statements that follow a 33XX statement on which ALL is specified. For example:*

```
LIMIT 3330,SKS=5,ALL=10
LIMIT 33XX,ALL=15
LIMIT 3340,RD=5,ALL=20
```

EREP limits the 3330 using the values in the 3330 statement, and limits all other DASD using the value in the 33XX statement. It ignores the "ALL" value in the 3340 statement, because the 33XX statement takes precedence. If you need the "ALL" value for 3340s, put that LIMIT statement before the one for 33XX, as follows:

```
LIMIT 3330,SKS=5,ALL=10
LIMIT 3340,RD=5,ALL=20
LIMIT 33XX,ALL=15
```

Now EREP limits the 3330 using the values in the 3330 statement, the 3340 using the values in the 3340 statement, and all other DASD using the value in the 33XX statement.

7. *Only one LIMIT statement is allowed for the general device class of 33XX.*

DEVICE TYPE	Seek Errors: **SKS**	Read Errors: **RD**	Bus Out Errors: **B**	Equipment Checks: **EQUCHK**	Check Data: **C**	Invoked Offset: **I**	Diskette Check: **D**	Overrun: **OVRN**	All Errors: **ALL**
3310	X	X		X	X				X
3330	X	X		X					X
3340	X	X		X					X
3350	X	X		X					X
3370				X	X	X			X
3375				X		X			X
3380				X		X			X
3830			X	X			X	X	X
3880			X	X			X	X	X
33XX	X	X	X	X	X	X	X	X	X

Bibliography

- *IBM Disk Storage Management Guide – Error Handling*, GA26-1672, contains much valuable information about 33XX DASD, including detailed information about the System Exception reports.

3380 DASD

"3380" represents a family of direct-access storage devices that comprises several models of the 3380 type.

See the "Bibliography" on page 15-10 for the titles of hardware manuals containing detailed information about the various models of 3380.

Special Considerations for EREP Reports

System Exception Report

- In the DASD System Exception reports that show FAILURE AFFECTs or PROBABLE FAILING UNITs, the 3380 family of devices has an additional category — MULTIPLE. This describes errors that may affect more than one device but are not controller failures.

  The reports in which you can see MULTIPLE used are:

  System Error Summary Part 2
  Subsystem Exception Report
  Symptom Code Summary
  String Summary

- In the System Exception report series, 3380 models are identified as follows:

Identifier	3380 Models
3380-DE	AD4, AE4, BD4, BE4

Other Reports

- In the device-dependent section of the Detail Edit (PRINT) report, the 3380 models are identified as they are for the System Exception Report series (see above).

Special Considerations for EREP Controls

- 3380 is a 33XX device; see "33XX DASD" on page 15-3 for more information.

Other Considerations

- MDR codes (one byte at offset 4) for the 3380 are:

MDR Code	*Description*
X'14'	3380 models AA4, A04, B04, AD4, and BD4. – Sense byte 4, bit 1 is 0 for models AA4, A04, and B04. – Sense byte 4, bit 1 is 1 for models AD4 and BD4 operating as a 3380 model AA4, A04, or B04.
X'1B'	3380 models AE4 and BE4.
X'1C'	3380 models AD4 and BD4 with full command support provided by the system.

- OBR codes (two bytes at offset X'36')[1] are:

OBR Device Class/Type	*Description*
X'200E'	3380 models AA4, A04, B04, AD4, and BD4. – Sense byte 4, bit 1 is 0 for models AA4, A04, and B04. – Sense byte 4, bit 1 is 1 for models AD4 and BD4 operating as a 3380 model AA4, A04, or B04.
X'201E'	3380 models AD4 and BD4 with full command support provided by the system.
X'202E'	3380 models AE4 and BE4

[1] The field, named OBRCODE, really starts at offset X'34', in the long form of the OBR record. The two bytes at offset X'34' are PUB bytes 4 and 5, used by VSE as device class and type. The two bytes listed here are bytes 18 and 19 from the MVS UCB.

- Storage capacities of the various models compare as follows:

Model	Factor
A04	1x
B04	1x
AD4	1x
BD4	1x
AE4	2x
BE4	2x

where "x" is the storage capacity of a 3380 Model AA4.

Bibliography

- *IBM 3380 Direct Access Storage Description and Users Guide*, GA26-1664, provides a complete description of the various models of the 3380 direct access storage characteristics, features and capabilities. In addition, the configuration and attachment options are described.

- *IBM 3880 Storage Control Models 1, 2, 3 and 4 Description Manual*, GA26-1661, contains commands, controls, sense descriptions, and recovery actions for all drives connected to the 3880 storage controller.

3850 Mass Storage System

Special Considerations for EREP Reports

The System Exception reports do not include virtual 33XX DASD connected to a 3850 mass storage system.

Special Considerations for EREP Controls

None.

Bibliography

- *OS/VS Mass Storage System (MSS) System Data Analyzer*, GC35-0027, describes the ISDASDA0 support for the IBM 3850 Mass Storage System.

9332 and 9335 DASD

These devices are members of the "33XX" family of direct access storage devices.

See the Bibliography for the titles of hardware manuals containing detailed information about these devices.

Special Considerations for EREP Reports

System Exception Report

- As members of the "33XX" family, the 9332 and 9335 devices are included in the System Exception report series.

Detail Edit Report

- For both the 9332 and 9335 devices, seek information is displayed when appropriate.

Special Considerations for EREP Controls

- 9332 and 9335 are 33XX devices; see "33XX DASD" on page 15-3 for more information.
- Use of the LIMIT control statement for 9332 and 9335 is invalid.

Other Considerations

- MDR codes (one byte at offset 4) are:

MDR Code	*Meaning*
X'1D'	9335 DASD
X'1E'	9332 DASD

- OBR codes (two bytes at offset X'36')[1] are:

OBR Device Class/Type	*Meaning*
X'2106'	9335 DASD
X'2107'	9332 DASD

Bibliography

- *IBM 9332 Disk Unit Models 200/400 Analyzing Problems*, SA21-9837.
- *IBM 9332 Disk Unit Models 200/400 Reference Code Guide*, SA21-9836.
- *IBM 9332 Disk Unit Models 200/400 Service Guide*, SY31-9026.
- *IBM 9335 Direct-Access Storage Subsystem Guide to Unit Reference Codes*, SY33-0143.
- *IBM 9335 Direct-Access Storage Subsystem Service Guide*, SY33-0113.

Chapter 16. Diskette Unit

The information in this subsection could be changed or superseded by newer information added under specific device types. If you are interested in a new device or machine number, look under the specific number.

Special Considerations for EREP Reports

Useful reports for these devices are:

SYSUM
EVENT
TRENDS
PRINT=PT or PS with DEV=nnnn and TYPE=OH (OBR and MIH records)

Care should be taken when requesting reports other than these as the results could be misleading.

Special Considerations for EREP Controls

None.

Devices Supported by EREP

This diskette unit is valid for DEV=

3540 diskette I/O unit

Chapter 17. Magnetic Tape Drives

The information in this subsection could be changed or superseded by newer information added under specific device types. If you are interested in a new device or machine number, look under the specific number.

Special Considerations for EREP Reports

- Useful reports for these devices are:

  SYSUM
  SYSEXN (34XX only)
  EVENT
  TRENDS
  THRESHOLD (3410, 3420 and 8809 only)
  PRINT=PT or PS with DEV=nnnn and TYPE=DOTH (DDR, OBR, MDR and MIH records).

  Care should be taken when requesting reports other than these as the results could be misleading.

  Some of the devices listed below may produce different record types. In that case, request that record type when requesting Detail Edit and Summary (PRINT) reports.

- The **System Exception report series** includes records from the following magnetic tape devices:

  3410
  3420
  3430
  3480
  9347

  See the subsection on 34XX Tape Devices for more information.

- The **Threshold Summary** report for tape subsystems includes records from the following magnetic tape devices:

  3410
  3420
  8809

- The fields in the Volume Statistics section of the **Threshold Summary** are used differently by different device types:

  - 3410/3420 OBR records do not use the the SIO WRTS field, and use the IOS RDS field for TOTAL IOS.

  - 8809 MDR and OBR records do not use the following fields at all:

    - MDR:

      TU SERIAL
      PERM RDS
      PERM WRTS
      PROGRAM ID
      MOD #
      DENSITY
      HDR SER

    - OBR:

      TU SERIAL
      TEMP RDS
      TEMP WRTS
      RETRY
      ERASE GAP
      MOD #
      DENSITY
      HDR SER

Special Considerations for EREP Controls

- It is unneccessary to use the TYPE parameter for the Threshold summary, as the THRESHOLD report parameter itself forces the selection of 3410, 3420, and 8809 OBR (TYPE=O) records. You can, however, use the DEV parameter to select records from one or two of the device types instead of all three. You can also use the DEVSER or VOLID parameter for the Threshold Summary to select records according to the device serial number or volume serial number they contain.

- You code the LIMIT control statement differently depending on which tape device you are dealing with. See "34XX Tape Devices" on page 17-4 and "9347 Magnetic Tape Drive" on page 17-23 for details.

Devices Supported by EREP

These magnetic tape storage devices are valid for DEV=

2400 magnetic tape drive
3400 magnetic tape drive
3410 magnetic tape drive
3420 magnetic tape drive
3430 tape unit
3480 magnetic tape subsystem
34XX general device class
7340 hypertape drive
8809 magnetic tape drive
9347 magnetic tape drive

Note: 34XX is a general device type designation that includes 3410, 3420, 3430, and 3480.

34XX Tape Devices

Special Considerations for EREP Reports

System Exception Report

- Besides the other report types listed for tape, SYSEXN is the most valuable for 34XX products. The System Exception series is meant to replace the Threshold summary for these devices.
- The System Exception report covers the 34XX family of tape devices, where 34XX includes 3410, 3420, 3430, and 3480.
- The SYSEXN (System Exception report series) report parameter produces different sets of reports for different 34XX tape devices. If you have all of the 34XX tape devices, you get one set of exception reports and summaries for 3410/3420 products, another for 3430, and another for 3480.
- If your tape subsystem is a large one, you can greatly improve performance when running the System Exception report by moving the tape records to a history data set and then running SYSEXN against those records. To move the records, run EREP using PRINT=NO, DEV=(34XX) and ACC=Y. Make sure you have included a job control statement or FILEDEF for the output data set.

Special Considerations for EREP Controls

LIMIT Control Statement

The LIMIT control statement applies only to the System Exception report series. The limiting action of this EREP control works differently for tape devices than it does for DASD or processors. You need two sets of keyword parameters for each device type; and, for 34XX devices, you must specify which tape density the limits apply to.

Note: ***Because tape density is not a factor for the 3480 tape subsystem, the 3480 LIMIT statement differs significantly from what follows.*** *See "3480 Flexible Media Tape Subsystem" on page 17-11 for details.*

The LIMIT statement has the following format for 34XX tape devices other than 3480:

```
LIMIT tape,xxbpi=nnn(ct)[,xybpi=nnn(ct)] ...
```

tape	is one of the device type numbers listed under "Valid Keywords and Values for the LIMIT Statement."
xx and xy	are pairs of initials indicating the types of temporary errors to be limited. See "Valid Keywords and Values for the LIMIT Statement."
bpi	is the density (bits per inch) at which the device is operating. The possible values for bpi are **1600** and **6250**.
nnn	Is a three-digit decimal value representing the number of megabytes of data processed between errors (MBYTES/ERROR).
ct	Is a decimal value from 1 to 99 representing the number of errors encountered before the device or volume appears on the Subsystem Exception report.

Valid Keywords and Values for the LIMIT Statement

tape can be one of the following:

34XX
3410
3420
3430

34XX is the general device type designation that, in this case, includes 3410, 3420, and 3430. When you code 34XX on a LIMIT statement, you are requesting that the limits apply to the reports for all of these devices.

The valid LIMIT keywords for 34XX tape drives are:

- **1600 BPI Temporary Errors**

To Set Limits For:	*Use Keyword:*
Hardware read	HR1600=nnn(ct)
Hardware write	HW1600=nnn(ct)
Volume read	VR1600=nnn(ct)
Volume write	VW1600=nnn(ct)

- **6250 BPI Temporary Errors**

To Set Limits For:	*Use Keyword:*
Hardware read	HR6250=nnn(ct)
Hardware write	HW6250=nnn(ct)
Volume read	VR6250=nnn(ct)
Volume write	VW6250=nnn(ct)

EREP uses both the nnn (MBYTES/ERROR) and ct (total errors) values to establish thresholds for temporary errors. If the number of errors recorded against the device or volume is greater than or equal to the count (ct) value, AND the average number of megabytes of data processed between errors is equal to or less than the error frequency (nnn) value, then the device or volume will appear on the Subsystem Exception report.

Points to Remember:

- To cover all the possible sources of errors for a 34XX device, you must code LIMIT statements for both hardware and volume read and write errors.
- If you do not code LIMIT statements for a tape device or volume, the Subsystem Exception report includes only the permanent errors recorded against that device or volume. However, all temporary errors appear in the Temporary Error Summary.
- To force EREP to show all the temporary errors on the Subsystem Exception report, use 999(1) for the nnn(ct) variables on the LIMIT statement.
- You should specify all LIMIT values. Results are unpredictable if any values are omitted, or if a value is coded as 0.
- The density of 6250 BPI applies only to 3420 and 3430 drives. If coded on a LIMIT statement for 34XX, it is ignored for 3410 devices.
- If a tape drive is operating at a density other than 1600 or 6250 BPI, EREP uses the LIMIT values you specify for 1600 BPI.
- Only one LIMIT statement is allowed for the general 34XX type.
- You may not continue a LIMIT statement from one line to the next. Generally, you should use separate LIMIT statements to establish hardware and volume limits for a device. If the device operates at both 1600 and 6250 BPI, you *must* use separate statements. However, if only one tape density is involved, you can combine all four keywords on the same LIMIT statement. For example, you might want to see only some of the temporary errors for your 3410 and 3420 drives, operating at 1600 BPI density, as follows:

  - *Hardware*

    Read: 1 or more errors, at 25 megabytes/error
    Write: 15 or more errors, at 10 megabytes/error

  - *Volume*

    Read: 1 or more errors, at 25 megabytes/error
    Write: 15 or more errors, at 10 megabytes/error

To set these limits, you could code the following LIMIT statements:

```
LIMIT 3410,HR1600=025(1),HW1600=010(15),VR1600=025(1),VW1600=010(15)
LIMIT 3420,HR1600=025(1),HW1600=010(15),VR1600=025(1),VW1600=010(15)
```

Because the limiting values and density are the same, these two statements could be combined into a single 34XX LIMIT statement:

```
LIMIT 34XX,HR1600=025(1),HW1600=010(15),VR1600=025(1),VW1600=010(15)
```

- When your 34XX devices are operating at different densities, and you want to use 34XX to designate them rather than individual device type numbers, you cannot fit all four sets of keywords on the single 34XX LIMIT statement allowed.

  In this case, if you specify ***only*** the volume or hardware values for ***both*** densities on a single 34XX LIMIT statement, EREP applies those values to whichever kinds of errors you have not specified. For example:

  ```
  LIMIT 34XX,VR1600=010(1),VW1600=010(1),VR6250=020(1),VW6250=020(1)
  ```

  EREP applies the values specified here for *volume* reads and writes to *hardware* reads and writes for all your 34XX devices. (When EREP checks the LIMIT statement syntax, it fills in any blanks it finds with the corresponding values supplied elsewhere on the same statement. This is why results can be unpredictable when you do not code all the values on a LIMIT statement, or code a value as 0.)

Using the DEVSER Parameter

The DEVSER selection parameter applies only to the Threshold Summary, and, by implication, only to 3410, 3420 and 8809 device types. Furthermore, the DEVSER parameter is valid only with TYPE=O, because only tape OBR records contain device serial numbers.

Bibliography

The following publications provide more detailed information about the 3410 and 3420 Tape Subsystems.

- *IBM 3410 Theory and Maintenance*, SY34-0031, part of the Field Engineering Maintenance Library, contains examples of Detail reports for these devices.
- *IBM 3410/3411 Tape Subsystem*, SV31-0714, includes status and sense byte information.
- *IBM 34XX Subsystem Operator's Guide*, GA32-0066.

3422 Tape Subsystem

Special Considerations for EREP Reports

In addition to the other reports in the System Exception series, the 3422 tape subsystem has a summary of the temporary errors recorded while the device was in forced-logging mode.

The Forced Log Error Summary report for the 3422 is almost identical to the report example for the 3430; Figure 17-1 on page 17-9. Aside from some minor variation in the interpretation of the sense information and some column headings, the reports for the 3422 are the same as the reports for other 34XX tape devices.

Special Considerations for EREP Controls

The 3422 is included as one of the 34XX tape devices for purposes of the System Exception report series and, on the DEV selection parameter, for other reports.

See "LIMIT Control Statement" on page 17-4 for details about the LIMIT statement.

Bibliography

The following publications provide more detailed information about the 3422 Tape Subsystem.

- *IBM 3422 Magnetic Tape Subsystem Reference Manual*, GA32-0089.
- *IBM 3422 Maintenance Information Manual*, SY32-5058.

3430 Tape Device

Special Considerations for EREP Reports

In addition to the other reports in the System Exception series, the 3430 magnetic tape device has a summary of the temporary errors recorded while the device was in forced-logging mode. The report first presents the totals of each type of failure by device address or number (CUA/DEVNO), and then lists each record in the order of its device address or number.

```
3430 TAPE FORCED LOG ERROR SUMMARY                    REPORT DATE 145 83
                                                      PERIOD FROM 240 81
                                                             TO   240 82

DEVICE ADDRESS - 08F3  1

             2      3      4       5       6       7   8       9        10
           LOAD    NON    READ    WRITE                                   LAST TIME
           CHECK   REP    ERROR   ERROR   OTHER   FCS PTR    VOLID    YYDDD HHMMSS

                             6                    15   9     VOL001   82169 113020
                             2                    77   2     VOL001   82169 113020


              0      0       8       0       0

3430 TAPE FORCED LOG ERROR DETAIL  11


            C                                                                          F  P
CHP DEVNO   P                   R/W       --CCW---    SCSW64-95  --------SENSE---------   S  T
-ID /CUA    U DTE   TIME  VOLID  E/U CMD   FLG  CNT    /CSW32-63  0    2    4    6    8    C  R

    08F3    A 169 113020 VOL001      02    32002222      08002222  0800 0000 0000 1590 20   15 9
    08F3    A 169 113020 VOL001      02    32002222      08002222  0800 0000 0000 1590 20   15 9
    08F3    A 169 113020 VOL001      02    32002222      08002222  0800 0000 0000 1590 20   15 9
    08F3    A 169 113020 VOL001      02    32002222      08002222  0800 0000 0000 1590 20   15 9
    08F3    A 169 113020 VOL001      02    32002222      08002222  0800 0000 0000 1590 20   15 9
    08F3    A 169 113020 VOL001      02    32002222      08002222  0800 0000 0000 1590 20   15 9
    08F3    A 169 113020 VOL001      02    32002222      08002222  0800 0000 0000 7720 20   77 2
    08F3    A 169 113020 VOL001      02    32002222      08002222  0800 0000 0000 7720 20   77 2

CPU    MODEL     SERIAL NUMBER
 A     3033          088888
 B     0168          090018
 C     3081XA        220014
```

Figure 17-1. Forced Log Error Summary for 3430

Figure 17-1 on page 17-9 is an example of the Forced Log Error Summary for a 3430 device.

Notes:

1. **DEVICE ADDRESS**. *Failing CUA or device number.*
2. **LOAD CHECK**. *Total number of load checks for this fault symptom code.*
3. **NON REP**. *Total number of non-reportable failures for this fault symptom code.*
4. **READ ERROR**. *Total number of read errors for this fault symptom code.*
5. **WRITE ERROR**. *Total number of write errors for this fault symptom code.*
6. **OTHER**. *Total number of other failures for this fault symptom code.*
7. **FSC**. *Fault symptom code, from the sense bytes.*
8. **PTR**. *Part of the fault symptom code.*
9. **VOLID**. *The volume identifier from the OBR record.*
10. **LAST TIME**. *Time of the most recent failure.*
11. **DETAIL**. *Same as in the 34XX Tape Permanent Error Summary.*

Special Considerations for EREP Controls

- The 3430 is included as one of the 34XX tape devices for purposes of the System Exception report series and, on the DEV selection parameter, for other reports.
- See "LIMIT Control Statement" on page 17-4 for details about the LIMIT statement and the 3430.

Bibliography

The following publications provide more detailed information about the 3430 Tape Subsystem.

- *IBM 3430 Magnetic Tape Subsystem: Description*, GA32-0076, includes status and sense byte data as part of the general problem determination and error recovery information.
- *3430 Magnetic Tape Subsystem Maintenance Information*, SY32-5053, part of the IBM Maintenance Library, includes examples of the EREP reports.

3480 Flexible Media Tape Subsystem

Special Considerations for EREP Reports

System Exception Report

If your installation's configuration includes a 3480 Flexible Media (tape) Subsystem, the device's records will be included automatically when you run the System Exception series.

EREP produces a separate set of System Exception reports for the 3480 Subsystem. Most of the reports do not differ significantly from the examples shown in Part 1 of this book; the titles, and some headings, are specific to the 3480. The figures on the following pages are examples of the System Exception report output for the 3480.

CODING NOTES

- Normally the System Exception report series includes the 34XX tape drives (3410, 3420, and 3430), and 3480 Subsystems. To limit the report to 3480s, run EREP once, requesting no report and the accumulation of selected 3480 records to an output history data set:

  ```
  //STEP1   EXEC  PGM=IFCEREP1,PARM=('PRINT=NO',
                  'DEV=(3480)','TYPE=OT','ACC=Y')
  ```

  Then run EREP again, requesting the System Exception report series and using the history data set created in STEP1 as input:

  ```
  //STEP2   EXEC  PGM=IFCEREP1,PARM=SYSEXN,HIST
  ```

  The result should be a System Exception series using only the records generated by the 3480 Subsystem.

  Note: You must request ***both OBR (type O) and MDR (type T) records in*** *the initial selection process; EREP uses both for the 3480 System Exception report.*

  If you already have a jobstream that generates a System Exception series for your 3420s, change the TYPE parameter from TYPE=O to TYPE=OT to make it work for 3480s.

- EREP uses 3480 MDR records *only* for the System Exception report series; you cannot get Detail Edit reports of those records.

```
SUBSYSTEM EXCEPTION                                            REPORT DATE 207 85
      3480                                                     PERIOD FROM 004 83
                                                                      TO   019 83

                                                    TEMP WRT(CT)  TEMP RD(CT)
          CURRENT LIMITS     HARDWARE                NONE (  )     NONE (  )
            MBYTES/ERR       VOLUME                  NONE (  )     NONE (  )

            VOLUME  DEVNO      EQU     ---MB/ERR  PERM---   ---MB/ERR  TEMP--- BUS OVR  TOTAL-MBYTES  HDR
EXCEPTION   SERIAL  /CUA   CPU CHK     READ(CT) WRITE(CT)   WRITE(CT) READ(CT) OUT RUN   READ WRITE   SER
  HARDWARE
    PERMANENT ERROR

                    01BX    B    1     -- (  0) 3841(  3)    60( 192) -- (  0)  0   0   8206  11523

                    01B2    B    1     -- (  0)   -- (  0)   49(  98) -- (  0)  0   0   4059   4898
                    02B2    B    0     -- (  0)  875(  1)    19(  44) -- (  0)  0   0      0    875
                    01B4    C    0     -- (  0)   -- (  0)   18(  80) -- (  0)  0   0    873   1489

VOLUME OR CREATING DRIVE
    PERMANENT READ OR WRITE ERRORS ON MORE THAN ONE DRIVE

            TAPE01          B    1     -- (  0)   68(  1)     0(  82) -- (  0)  0   0      0     68  0000
            L00100          B    0     -- (  0)   55(  2)    18(   6) -- (  0)  0   0      0    110  0000

                                 1
  TOTAL NUMBER OF DRIVES FAILING LIMITS      3 ( 16%)           TOTAL NUMBER OF VOLUMES USED   =    23
                         PASSING LIMITS     16 ( 84%)           TOTAL NUMBER OF VOLUMES LISTED =     2

CPU    MODEL    SERIAL NUMBER
 A     3081        020344
 B     4341        015760
```

Figure 17-2. 3480 Subsystem Exception Report

3480 Subsystem Exception Report

This report, Figure 17-2, is essentially the same as the other tape Subsystem Exception reports, except that the errors are not grouped by tape density; density is not relevant to the 3480.

Notes:

1. *Drives that "fail" the limits appear on the report; those "passing" the limits do not. The "limits" are LIMIT values.*

```
3480 PERMANENT / RECOVERED ERROR SUMMARY              REPORT DATE 207 85
                                                      PERIOD FROM 004 83
                                                             TO   019 83
                                                                                                    4

  1                                                                                 1    1    1    1    1    2    2    2    2    2 5  3
            C                         R   SENSE BYTES---> 0    2    4    6    8    0    2    4    6    8    0    2    4    6    8    0
 CHP DEVNO  P                         W       CCW  SCSW64-95                        --------CU---------      ---DR----              CU
 -ID /CUA   U DTE    TIME    VOLID    E CMD FLG   /CSW32-63                         ERR1 ERR2 ERRL    HW      ERR1 ERR2              SER#
                                            3          ****** PERMANENT ERRORS ******

**** DRIVE ****

       1B4   B 018 112929 SSAG23 O   00 00    26000000   0041 612E 0000 0020 EF00 7161 7161 7157 D6CC 0001 0264 0264 F604 C780 0000 0000

       2B2   B 006 131152 ASAG65 W   01 64    06004434   0A44 8825 0002 7E20 0000 7401 7401 7401 D002 0000 0000 0000 F084 E780 0000 0000
       2B2   B 007 111537 TAPE01 E   07 20    0E000000   5040 8822 0000 0020 0000 8E05 8E05 8E05 0000 0000 0000 0000 F684 E780 0000 0000
       2B2   B 007 115041        O   02 00    0E400050   0041 892E 0000 0020 EF00 7161 7161 7161 0000 0000 0264 0264 F084 E780 0000 0000
       2B2   B 007 115219        O   02 00    0E400050   0041 892E 0000 0020 EF00 7161 7161 7161 0000 0000 0264 0264 F084 E780 0000 0000
       2B2   B 007 115248        O   02 00    0E400050   0041 892E 0000 0020 EF0.0 7161 7161 7161 0000 0000 0264 0264 F084 E780 0000 0000

**** VOLUME FAILED MORE THAN ONE DRIVE ****

       1B1   B 012 164444 L00100 W   01 64    060020BC   0A44 6025 0017 7320 0024 740D 740D 740D D824 0000 0000 0000 F604 E780 0000 0000

**** OPERATOR OR OPERATIONAL ****

       18B   B 018 105558  /     O   07 20    02000000   4248 783B 0000 0020 0008 8202 0000 0000 0000 0000 00FF 0000 F004 E780 0000 0000
       18B   B 018 110320 SSAG03 O   02 04    02000001   4240 783B 0000 0420 0008 8202 0000 0000 0000 0000 0000 0000 F604 E780 0000 0000

       1B4   B 018 110214        O   07 20    02000050   4248 603B 0000 0020 0004 8202 0000 0000 0000 0000 00FF 0000 F004 E780 0000 0000
**** OTHER ****

       1B4   B 018 113811 SSAG23 O   02 20    06000050   0084 0602 3000 0002 0EF8 0715 7709 3709 3000 0000 1026 4026 4F60 4E78 0000 0000

       2B2   B 012 101901        O   C7 00    02000000   4040 8000 0000 0020 0000 0000 0000 0000 0000 0000 0000 0000 F004 C780 0000 0000
                                                       ****** RECOVERED ERRORS ******

**** CONTROL UNIT ****

       2B2   B 006 131342 ASAG65 O   1F 60    06000050   0044 8848 0002 7C20 0000 3F13 0000 0000 0000 0000 0000 0000 F084 C780 0000 0000
       2B2   B 007 111537 ASAG65 O   9F 60    26000000   4040 8048 0000 0020 0081 6EED 0000 0000 0000 0000 0000 0000 F684 C780 0000 0000

CPU     MODEL     SERIAL NUMBER
 A      3081          020344
 B      4341          015760
```

Figure 17-3. 3480 Permanent Error Summary

3480 Permanent Error Summary

This report shows the sense data in permanent-error OBR records. Sense bytes 10 - 17 and 20 - 24 are presented as error codes for use as indexes into the maintenance manual. The first group are associated with control-unit-detected errors; the second, with drive errors.

Notes:

1. *Channel-path IDs appear for 370-XA mode records.*
2. *R/W/E: Kinds of commands associated with errors — read, write, equipment and operator.*
3. *Command code and flag byte from the CCW.*
4. *Error codes, to be used as indexes into the maintenance manual; associated with the control unit (CU) and device (DR).*
5. *Serial number of the control unit to which the device is attached.*

```
3480 FORCED LOG REPORT                                     REPORT DATE 214 85
                                                           PERIOD FROM 006 83
                                                                  TO   018 83

                                                                                      1    1    1    1    1    2    2    2    2    2    3
         C                          R   SENSE BYTES--->  0    2    4    6    8    0    2    4    6    8    0    2    4    6    8    0
CHP DEVNO P                         W      CCW  SCSW64-95                            ---------CU----------     ---DR----            CU
-ID /CUA  U DTE  TIME   VOLID       E CMD FLG  /CSW32-63                             ERR1 ERR2 ERRL HW         ERR1 ERR2            SER#

    01B1  A 012 164444 L00100 W  01 64   060020BC  0A44 6025 0017 7319 0024 740D 740D 740D D824 0000 0000 0000 F604 E780 0000 0000

    01B2  A 007 111523 TAPE01 O  07 20   02000000  4248 603B 0000 0019 0000 8202 0000 0000 0000 0000 0000 0000 F684 E780 0000 0000
    01B2  A 007 112209 TAPE01 W  01 64   06007FF8  0A44 6825 0002 8E19 0000 7401 7401 7401 D002 0000 0000 0000 F084 E780 0000 0000
    01B2  A 012 164219 L00100 W  01 64   06004434  0A44 6025 0001 F519 0024 740F 740F 740F D82, 0000 0000 0000 F604 E780 0000 0000

    01B4  A 018 112929 SSAG23 O  00 00   26000000  0041 612E 0000 0019 EF00 7161 7161 7157 D6CC 0001 0264 0264 F604 C780 0000 0000

    02B2  A 006 131152 ASAG65 W  01 64   06004434  0A44 8825 0002 7E19 0000 7401 7401 7401 D002 0000 0000 0000 F084 E780 0000 0000
    02B2  A 006 131342 ASAG65 O  1F 60   06000050  0044 8848 0002 7C19 0000 3F13 0000 0000 0000 0000 0000 0000 F084 C780 0000 0000
    02B2  A 007 111537 ASAG65 O  9F 60   26000000  4040 8048 0000 0019 0081 6EED 0000 0000 0000 0000 0000 0000 F684 C780 0000 0000
    02B2  A 007 111537 TAPE01 E  07 20   0E000000  5040 8822 0000 0019 0000 8E05 8E05 8E05 0000 0000 0000 0000 F684 E780 0000 0000
    02B2  A 007 115041        O  02 00   0E400050  0041 892E 0000 0019 EF00 7161 7161 7161 0000 0000 0264 0264 F084 E780 0000 0000
    02B2  A 007 115219        O  02 00   0E400050  0041 892E 0000 0019 EF00 7161 7161 7161 0000 0000 0264 0264 F084 E780 0000 0000
    02B2  A 007 115248        O  02 00   0E400050  0041 892E 0000 0019 EF00 7161 7161 7161 0000 0000 0264 0264 F084 E780 0000 0000
    02B2  A 012 101901        O  C7 00   02000000  4040 8000 0000 0019 0000 0000 0000 0000 0000 0000 0000 0000 F004 C780 0000 0000

.CPU   MODEL    SERIAL NUMBER
 A     4341         015760
```

Figure 17-4. 3480 Forced Log Summary

3480 Forced Log Summary

This report contains data only when the device has been running in forced-logging mode, which produces OBR records for temporary errors. The report summarizes those temporary-error OBR records using the same format and headings as in the Permanent Error Summary.

Note: See the Permanent Error Summary for information about field names, etc.

3480 TEMPORARY ERROR SUMMARY

REPORT DATE 207 85
PERIOD FROM 004 83
TO 019 83

	1			2						3		4						5		
		C		---WRITE----		----READ----		TOTAL-MBYTES		TOTAL BLOCKS --PROCESSED--		----READ-----		----WRITE----		READ RECVY	WRITE ERASE	DRV DET	CU EQU	SPD
DEVNO /CUA	DRIVE ID	P U	MOUNTS	MB/ERR	ERRCT	MB/ERR	ERRCT	READ	WRITE	READ	WRITE	MB/COR	ECC	MB/COR	ECC	ACTS	GAPS	ERR	CHK	VAR
181	01171	A	1	--	0	--	0	0	126	--	3840	--	0	15	8	0	0	0	0	0
182	01172	A	1	--	0	--	0	0	67	--	2048	--	0	22	3	0	0	1	0	0
188	01178	A	11	43	6	--	0	6	263	212992	19456	0	15	0	513	0	6	2	0	0
188	01168	B	2	--	0	--	0	0	8	--	7424	--	0	--	0	0	0	0	0	0
189	01179	A	8	--	0	--	0	199	196	307200	13824	12	16	7	25	0	0	2	0	0
18A	0116A	B	1	--	0	--	0	0	0	16384	--	--	0	--	0	0	0	0	0	0
18B	00000	B	8	--	0	--	0	197	402	118784	16896	65	3	36	11	0	0	1	0	0
1A7	01187	B	0	--	0	--	0	0	0	--	--	--	0	--	0	0	0	0	0	0
1A9	01189	B	0	--	0	--	0	0	0	--	--	--	0	--	0	0	0	0	0	0
1AA	0118A	B	0	--	0	--	0	0	0	--	--	--	0	--	0	0	0	0	0	0
1AB	0118B	B	0	--	0	--	0	0	0	--	--	--	0	--	0	0	0	0	0	0
1AD	0118D	B	0	--	0	--	0	0	0	--	--	--	0	--	0	0	0	0	0	0
1B1	01171	B	49	329	14	--	0	3153	4606	13640K	514304	0	5377	0	12920	0	26	4	0	0
1B2	01162	B	73	49	98	--	0	4059	4898	13984K	451840	4	895	1	4084	0	233	3	0	0
1B3	01173	B	2	--	0	--	0	121	311	1597K	109824	4	28	7	43	0	0	0	0	0
1B4	01174	B	6	18	80	--	0	873	1489	1008K	82688	0	22646	0	18393	0	86	1	0	0
1B8	01168	B	2	--	0	--	0	0	219	--	12288	--	0	3	73	0	0	1	0	0
2B1	01161	B	17	69	17	--	0	0	1187	--	66816	--	0	2	402	0	18	0	0	0
2B2	00000	B	21	19	44	--	0	0	875	--	48896	--	0	3	290	0	73	1	1	0
TOTALS:			202		259		0	8608	14647	30884K	1350K		28980		36765	0	442	16	1	0

AVERAGE MEGABYTES/TEMPORARY READ ERROR	=	*
AVERAGE MEGABYTES/TEMPORARY WRITE ERROR	=	56 7
AVERAGE MEGABYTES/RECOVERED ERROR	=	11627
AVERAGE MEGABYTES/PERMANENT READ ERROR	=	*
AVERAGE MEGABYTES/PERMANENT WRITE ERROR	=	3661
AVERAGE MEGABYTES/PERMANENT ERROR #	=	5813
AVERAGE MEGABYTES/PERMANENT HARDWARE ERROR	=	1937
AVERAGE MEGABYTES/PERMANENT VOLUME ERROR	=	5813
AVERAGE MEGABYTES/PERMANENT OTHER ERROR	=	*
TOTAL MEGABYTES PROCESSED	=	23255

6 (*) = THERE WERE NO ERRORS LOGGED FOR CALCULATION

8 (#) = COMPARABLE TO 3420 AVERAGE MB/PERM. ERROR RATE

CPU	MODEL	SERIAL NUMBER 9
A	3081	020344
B	4341	015760

Figure 17-5. 3480 Temporary Error Summary

3480 Temporary Error Summary

This report totals the temporary errors recorded in MDR records against each device and presents selected statistical data from the records. In addition, EREP calculates the average frequency of the various kinds of temporary errors for this report.

*Note: **All** temporary errors appear on the report, without regard for any limits set via LIMIT statements.*

The 3480 Volume Statistics Summary, not shown here, contains the same kind of statistical data as in the Temporary Error Summary, but for both temporary errors that exceeded the LIMIT values and permanent (OBR) errors.

Notes:

1. *Four-digit control-unit serial number plus the one-digit address position of the drive.*
2. *Totals and frequencies of errors.*
3. *See the 3480 Subsystem Description, GA32-0042, for the definition of blocks.*
4. *Sense data from the MDR record; the correction/recovery actions recorded.*
5. *Data used in calculating the averages at 7.*
6. *Applies to figures presented at 7.*
7. *Translation of block information.*
8. *Megabytes processed per permanent error — can be used to compare the efficiency of two different types of tape devices.*
9. *Processors, identified from filtered data or SHARE statements. XA following the model number indicates that the processor was running in 370-XA mode.*

```
3480 ERROR CODE SUMMARY REPORT                              REPORT DATE 207 85
                                                            PERIOD FROM 004 83
                                                                   TO   019 83

       C   C C C
 DEVNO P   H U U ------------CU-----------  -----DR------               **** DATE/TIME ****
 /CUA  U   R R D ERR1 F    ERR2  ERRL    HW  ERR1 F    ERR2   OCCURRENCES  *** LAST ENTRY ****

  01B1 B   C 0 0 740D 24   740D  740D  D824  0000 00   0000             1  012/83   16:44:44:35

  01B2 B   C 0 1 8E05 00   8E05  8E05  0000  0000 00   0000             1  007/83   11:15:37:57
  01B2 B   C 0 0 A130 12   A130  A130  D58A  0000 00   0000             4  012/83   16:41:15:86
  01B2 B   C 0 0 740F 24   740F  740F  D824  0000 00   0000             1  012/83   16:42:19:90
  01B2 B   C 0 1 7401 00   7401  7401  D002  0000 00   0000             1  007/83   11:22:09:09

  01B4 B   C 0 0 7161 00   7161  7157  D6CC  0264 00   0264             1  018/83   11:29:29:49
  01B4 B   C 0 0 0715 F8   7709  3709  3000  1026 00   4026             1  018/83   11:38:11:11

  02B2 B   D 0 1 7161 00   7161  7161  0000  0264 00   0264             3  007/83   11:52:48:34
  02B2 B   D 0 0 0000 00   0000  0000  0000  0000 00   0000             1  012/83   10:19:01:68
  02B2 B   D 0 1 7401 00   7401  7401  D002  0000 00   0000             1  006/83   13:11:52:81
  02B2 B   D 0 1 8E05 00   8E05  8E05  0000  0000 00   0000             1  007/83   11:15:37:81
  02B2 B   D 0 1 3F13 00   0000  0000  0000  0000 00   0000             1  006/83   13:13:42:27
  02B2 B   D 0 0 6EED 81   0000  0000  0000  0000 00   0000             1  007/83   11:15:37:17

CPU    MODEL    SERIAL NUMBER
 A     3081        020344
 B     4341        015760
```

Figure 17-6. Error Code Summary for 3480

3480 Error Code Summary Report

This report summarizes the error codes recorded by the 3480 subsystem for its various field-replaceable units (FRUs). It lists the errors by device address or number and by the error code, a set of numbers that identifies the particular field-replaceable unit involved. The report notes the number of occurrences for each error code, and the time of occurrence.

Notes:

1. *Channel-adapter interface.*
2. *The path through the control unit to the device.*
3. *Error codes from OBR sense bytes.*
4. *Control-unit-detected errors. "F" is a flag byte that modifies the ERR1 indicator. "ERRL" is the last control unit error code.*
5. *Control-unit-detected hardware errors.*
6. *Drive-detected errors.*
7. *Again, "F" indicates a flag byte used in data analysis.*

3480 DEVNO/CUA Statistics Summary

One of these reports is generated for each device (device number or CUA) that appears as a hardware exception on the Tape Subsystem Exception report. The report presents the DEVNO/CUA's temporary errors that failed the limits set in LIMIT control statements. The errors appear by volume serial number in the order (date and time) in which they occurred.

```
3480 DEVNO/CUA STATISTICS SUMMARY FOR-02B2          REPORT DATE 207 85
                                                    PERIOD FROM 004 83
                                                           TO   019 83

              DEVICES FAILING LIMITS OR PERMANENT ERRORS

              CURRENT LIMITS                  TEMP WRT(CT)  TEMP RD(CT)
                MBYTES/ERR        HARDWARE       NONE (   )    NONE (   )

   |      |      |ER|F |_MB PROC_|        |___DATA  ERR___|_MB/ERROR__|ERSE|READ|DRV_|_CU_|INSTANT|_BLK PROC|  _BLK COR_|_HDR_|C
DTE| TIME |VOLID |PA|M |         |DATA CHK|       RD   RD |TEMPORARY  |GAPS|RTY |DET |EQU |SPD VAR|WRT  READ|  WRT  READ| SER |P
   |      |      |  |T |WRT   RD |WRT  RD | WRT  FWD BKWD|WRITE  READ|    |    |    |CHKS|RD  WRT|         |           |     |U

006 092221 SSAG00 2B 21    1    0    0    0    0    0    0    --    --    0    0    0    0    0    0   12   12    0    0 0000  B
006 092241        2B 21    0    0    0    0    0    0    0    --    --    0    0    0    0    0    0   12   12    0    0 0000  B
006 100627 SSAG00 2B 21    1    0    0    0    0    0    0    --    --    0    0    0    0    0    0   12   12    0    0 0000  B
006 100724 SSAG00 2B 21    1    0    0    0    0    0    0    --    --    0    0    0    0    0    0   12   12    0    0 0000  B
006 100734        2B 21    0    0    0    0    0    0    0    --    --    0    0    0    0    0    0   12   12    0    0 0000  B
006 101800 SSAG00 2B 21    1    0    0    0    0    0    0    --    --    0    0    0    0    0    0   12   12    0    0 0000  B
006 102115 SSAG00 2B 21    1    0    0    0    0    0    0    --    --    0    0    0    0    0    0   12   12    0    0 0000  B
006 102435 SSAG00 2B 21    1    0    0    0    0    0    0    --    --    0    0    0    0    0    0   12   12    0    0 0000  B
006 105756 SSAG00 2B 21    1    0    0    0    0    0    0    --    --    0    0    0    0    0    0   12   12    0    0 0000  B
006 105836 SSAG00 2B 21    1    0    0    0    0    0    0    --    --    0    0    0    0    0    0   12   12    0    0 0000  B
006 105837        2B 21    0    0    0    0    0    0    0    --    --    0    0    0    0    0    0   12   12    0    0 0000  B
006 112858 SSAG00 2B 21    1    0    0    0    0    0    0    --    --    0    0    0    0    0    0   12   12    0    0 0000  B
006 114346 SSAG00 2B 21    1    0    0    0    0    0    0    --    --    0    0    0    1    0    0   12   12    0    0 0000  B
006 114717        2B 21    0    0    0    0    0    0    0    --    --    0    0    0    0    0    0   12   12    0    0 0000  B
006 123258 SSAG00 2B 21    0    0    0    0    0    0    0    --    --    0    0    0    0    0    0   12   12    0    0 0000  B
006 125917        2B 21    0    0    0    0    0    0    0    --    --    0    0    0    0    0    0   12   12    0    0 0000  B
006 131152 ASAG65 25 20                                                                                          0000  B
006 131341 ASAG65 2B 21    0    0    0    0    0    0    0    --    --    8    0    0    0    0    0   12   12    5    0 0000  B
006 131343 ASAG65 2B 21    0    0    0    0    0    0    0    --    --    0    0    0    0    0    0   12   12    2    0 0000  B
006 142557 SSAG21 2B 21  211    0    0    0    0    0    0    --    --    0    0    0    0    0    0 3080   12   19    0 0000  B
006 143021 ASAG65 2B 21    1    0    0    0    0    0    0    --    --    0    0    0    0    0    0  197   12   40    0 0000  B
006 143606 SSAG23 2B 21  211    0    0    0    0    0    0    --    --    0    0    0    0    0    0 3080   12   55    0 0000  B
006 144315 ASAG61 2B 21  211    0    0    0    0    0    0    --    --    0    0    0    0    0    0 3080   12   67    0 0000  B
006 145422 SSAG25 2B 21  213    0    0    0   44    0    0     4    --   65    0    0    0    0    0 3080   12  102    0 0000  B
007 111652 TAPE01 2B 21    0    0    0    0    0    0    0    --    --    0    0    1    0    0    0   12   12    0    0 0000  B
007 111833        2B 21    0    0    0    0    0    0    0    --    --    0    0    0    0    0    0   12   12    0    0 0000  B
007 112018        2B 21    0    0    0    0    0    0    0    --    --    0    0    0    0    0    0   12   12    0    0 0000  B
007 115041        2E 20                                                                                          0000  B
007 115219        2E 20                                                                                          0000  B
007 115248        2E 20                                                                                          0000  B
012 101901        00 20                                                                                          0000  B

CPU    MODEL     SERIAL NUMBER
 A     3081         020344
 B     4341         015760
```

Figure 17-7. 3480 DEVNO/CUA Statistics Summary

Special Considerations for EREP Controls

- 3480 is a valid value for the DEV selection parameter.
- 3480 is also included in 34XX. That is, when you code DEV=34XX, EREP selects records from 3410, 3420, 3430, and 3480 tape drives.[1]

LIMIT Control Statement for 3480

Because tape density is not a factor for the 3480 device, the LIMIT control statement is different for the 3480 than for other supported tape devices. The format of the 3480 LIMIT statement is:

```
LIMIT 3480,xx3480=nnn(ct),xy3480=nnn(ct)
```

xx and xy are pairs of initials indicating the types of temporary errors that are to be limited. The possible values for xx and xy are listed under the valid LIMIT keywords for 3480.

nnn Is a 3-digit decimal value representing the number of megabytes of data processed between errors (MBYTES/ERROR).

ct Is a decimal value from 1 to 99 representing the number of errors encountered before the device or volume appears on the Subsystem Exception report.

Valid LIMIT Keywords for 3480

The LIMIT control statement uses unique keywords for the 3480. The valid keywords are:

To Set Limits For:	*Use Keyword:*
Hardware read	HR3480=nnn(ct)
Hardware write	HW3480=nnn(ct)
Volume read	VR3480=nnn(ct)
Volume write	VW3480=nnn(ct)

EREP uses both the nnn (MBYTES/ERROR) and ct (total errors) values to establish thresholds for temporary errors. If the number of errors recorded against the device or volume is greater than or equal to the count (ct) value, AND the average number of megabytes of data processed between errors is equal to or less than the error frequency (nnn) value, then the device or volume will appear on the 3480 Subsystem Exception report.

---

1 This depends on the report requested. See "34XX Tape Devices" on page 17-4 for details.

Points to Remember:

- If you do not code LIMIT statements for a 3480 device or volume, the Subsystem Exception report includes only the permanent errors recorded against that device or volume. However, all temporary errors appear in the 3480 Temporary Error Summary.
- To force EREP to show all the temporary errors on the Subsystem Exception report, use 999(1) for the nnn(ct) variables on the LIMIT statement.
- You should code all LIMIT keyword values. Results are unpredictable if any values are omitted, or if a value is coded as 0.
- You may not continue a LIMIT statement from one line to the next. Generally, you should use separate LIMIT statements to establish hardware and volume limits for the device.

Other Considerations

- The MDR device code for the 3480 Tape Subsystem is X'41'. This code is one of the bit masks that can appear in Byte 1 of the record-dependent switches in the header portion of the MDR record.
- The OBR code (UCB device class and type: MVS only) for the 3480 is X'8080'. See "Device Type Codes in the OBR Record" on page 12-13 for more information.

Bibliography

The following publications have more information about 3480 error recording, and detailed examples of EREP reports for the device.

- *3480 Subsystem Description*, GA32-0042, contains status and sense byte data, as part of the problem determination and error recovery information.
- *3480 Subsystem Maintenance Information*, in 4 volumes. EREP report examples are in Volume 1, SY32-5048.

9347 Magnetic Tape Drive

Special Considerations for EREP Reports

Useful reports for these devices are:

SYSUM
TRENDS
EVENT
PRINT=PT or PS
SYSEXN

Care should be taken when requesting reports other than these as the results could be misleading.

Subsystem Exception Report

Aside from minor variations in headings the subsystem exception reports are similar to those described for the 3480 (See "3480 Subsystem Exception Report" on page 17-12) with these exceptions:

- Use of the LIMIT control statement is invalid for the 9347. Therefore, the current limits are not reported.
- The count and frequency of permanent and temporary errors is not recorded. Therefore, the MB/ERR counts are not reported.
- There is no Forced Log Summary.

Special Considerations for EREP Controls

Use of the LIMIT control statement is invalid for 9347 devices.

Bibliography

The following publications provide more detailed information about 9347 Magnetic Tape Drive.

- *IBM 9347 Magnetic Tape Drive Setting Up*, SA21-9528.
- *IBM 9347 Magnetic Tape Drive Operating*, SA21-9529.
- *IBM 9347 Magnetic Tape Drive Reference Code Guide*, SA21-9838.
- *IBM 9347 Magnetic Tape Drive Analyzing Problems*, SA21-9839.
- *IBM 9347 Magnetic Tape Drive Service Guide*, SY31-9030.

Chapter 18. OCR/MICR devices

The information in this subsection could be changed or superseded by newer information added under specific device types. If you are interested in a new device or machine number, look under the specific number.

Special Considerations for EREP reports

Useful reports for these devices are:

SYSUM
EVENT
TRENDS
PRINT=PT or PS with DEV=nnnn and TYPE=OTH (OBR, MDR and MIH records)

Care should be taken when requesting reports other than these as the results could be misleading.

Some of the devices listed below may produce different record types. In that case, request that record type when requesting Detail Edit and Summary (PRINT) reports.

Special Considerations for EREP Controls

None.

Devices Supported by EREP

These OCR/MICR devices are valid for DEV=

1255 MICR reader
1270 optical character reader
1275 optical reader/sorter
1285 optical reader
1287 optical reader/sorter
1288 optical page reader
1419 MICR reader/sorter
3886 optical character reader
3890 document processor
3895 document reader/sorter

Chapter 19. Printers

The information in this subsection could be changed or superseded by newer information added under specific device types. If you are interested in a new device or machine number, look under the specific number.

Special Considerations for EREP reports

- Useful reports for these devices are:

  SYSUM
  EVENT
  TRENDS
  PRINT=PT or PS with DEV=nnnn and TYPE=OTH (OBR, MDR and MIH records).

  Care should be taken when requesting reports other than these as the results could be misleading.

  Some of the devices listed below may produce different record types. In that case, request that record type when requesting Detail Edit and Summary (PRINT) reports.

- EREP produces a combined OBR/MDR Summary for the 3800 Printing Subsystem when you request Detail Summaries for that product.

Special Considerations for EREP Controls

None.

Devices Supported by EREP

These printers are valid for DEV=

1053 printer
1403 printer
1443 printer
2245 printer
3203 printer
3211 printer
3262 printer
3284 printer
3286 printer
3287 printer
3288 printer
3289 line printer
32XX
3800 printing subsystem
3820 page printer
38XX
4245 printer.
4248 printer
5080 graphics systems workstation
5203 printer

Note: 32XX and 38XX are general device type designations that include families of IBM printers.

Bibliography

The following publications provide more information about IBM printers:

- *3800 Printing Subsystem Operator's Guide*, GA26-1634.
- *3800-3 Operator's Guide*, GA32-0068.

3820 Printer

The general information about EREP for IBM printers also applies to the 3820 page printer.

Special Considerations for EREP Reports

EREP includes records from the 3820 printer with the other OBR records produced by the 3791 cluster controller. All 3820 records and incidents are identified by "3791."

Special Considerations for EREP Controls

3820 is a valid number for the EREP DEV (device type) selection parameter. However, that number does not appear in EREP reports. Instead, the records are listed under "3791."

Bibliography

The following publication provides more information about the 3820 page printer.

- *3820 Maintenance Library, Volume 2, Section 6-800.*

4248 Printer

The general information about EREP for IBM printers also applies to the 4248 line printer.

Special Considerations for EREP Controls

4248 is a valid number for the EREP DEV (device type) selection parameter.

If the device is running in 3211 mode, code 3211 on the DEV parameter.

Bibliography

The following publication provides more information about the 4248 printer.

- *4248 Printer Model 1 Description*, GA24-3927.

Chapter 20. Processors (CPUs)

The information in this subsection could be changed or superseded by newer information added under specific product types. If you are interested in a new product or machine number, look under that number.

Special Considerations for EREP Reports

- Useful reports for these devices are:

  SYSUM
  TRENDS
  EVENT
  SYSEXN (not for all types)
  PRINT=PT or PS with MOD=nnnn or CPU=nnnnnn.nnnn and
  TYPE=MC (MCH and CCH/SLH records)

  Care should be taken when requesting reports other than these as the results could be misleading.

- The System Exception report applies to only some processors. See the individual machine types for more information.

- The 308X processors are not supported by the System Exception report series as error sources. Instead, these processors have, as part of their error recovery systems, *service action points* that appear on the service support console to indicate which part of the processor complex is experiencing machine-check or channel-check problems.

- EREP lumps SLH and CRW records under a single processor ID, even for a 3084 running in partitioned mode.

- Whether or not the Software Recovery Status bits are valid depends upon the operating system and the processor. The information in these bits applies only to the 3090 processor. See "3090 Processor" on page 20-8 for more information.

Special Considerations for EREP Controls

LIMIT Control Statement

The LIMIT control statement applies only to the System Exception report series. It works differently for each of the product groups it is used for; following is a description of the way you use it for the processor and channel Subsystem Exception reports.

The LIMIT statement has the following format for processors:

```
LIMIT cpu,keyword=nn[,cpu,keyword=nn] . . .
```

cpu is one of the following S/370 processors and its associated channels:

0158
0168
3031
3032
3033

keyword is one of the keywords representing the various types of soft machine checks or channel checks covered by the System Exception report series. See "Valid LIMIT Keywords for Processors/Channels."

nn is a two-digit decimal value ranging from 1-99. It indicates the minimum number of errors that must be recorded during a 60-minute "reference period" for the processor or channel to be included on the Subsystem Exception report. The reference period begins when an error of the type specified in the LIMIT statement is recorded.

Valid LIMIT Keywords for Processors/Channels

The valid LIMIT keywords for processors and channels are:

- **Processor**

To Set Limits For:	*Use Keyword:*
External damage	EXTD=nn
Hardware instruction retry	HIRS=nn
Buffer error	BUFE=nn

- **Channel**

To Set Limits For:	*Use Keyword:*
Channel error	CHAN=nn
Storage error	STOR=nn
Director error	DRCT=nn
Control Unit error	CTRL=nn

Points to Remember:

- If you do not supply a number for "nn," EREP applies a default value of 01, meaning that all soft errors recorded on processors or channels are included in the printed report. In this case, the line in the report showing the CURRENT LIMITS contains 00 for that keyword.

- The LIMIT keywords for processors and channels apply to soft errors only. They represent the types of errors listed under SOFT MACHINE CHECK in the Processor Subsystem Exception report, and as the three SERVICE LEVEL INDICATOR categories in the Channel Subsystem Exception report. See the Subsystem Exception report examples in Chapter 2, "EREP Reports."

- The STOR and DRCT keywords for channel errors are mutually exclusive: STOR applies to the 0158 and 0168 processors, and DRCT applies to the 303X processors.

- You can set limits for processor and channel errors on separate LIMIT statements or on the same statement. For example:

  ```
  LIMIT 3033,EXTD=05,HIRS=05,BUFE=03
  LIMIT 3033,CHAN=01,DRCT=04,CTRL=08
  ```

  or

  ```
  LIMIT 3033,CHAN=01,DRCT=04,CTRL=08,EXTD=05,HIRS=05,BUFE=03
  ```

- You may not continue a LIMIT statement from one line to the next. You may, however, code as many separate LIMIT statements as you need.

Processors Supported by EREP

These processor machine types are valid for CPU= and MOD=

```
0115  0125  0135  0138  0145  0148  0155  0158
0165  0168  3031  3032  3033  3052  3062  3081
3083  3084  3090  4321  4331  4341  4361  4381
9081  9083  9373  9375  9377
```

303X Processors

Special Considerations for EREP Reports

The 303X processors are among those included in the System Exception report series.

Special Considerations for EREP Controls

LIMIT Control Statement

See the discussion of the LIMIT statement under Chapter 20, "Processors (CPUs)" for details about the 303X processors.

The only valid values for the CHAN LIMIT statement keyword for a 303X processor are CHAN=00 and CHAN=01. If you code any other value for CHAN, EREP processes it as if it were CHAN=01.

Other Considerations

Frame Records for Machine and Channel Checks

The machine-check and channel-check records generated by 303X processors must be translated before EREP can interpret them for its reports. The records are translated using frames provided by the processors' Service Record File devices.

Sets of frames are recorded in a reserved area of the ERDS by all three operating systems for EREP's reference.

- VSE systems use the SET RF=CREATE IPL command to initialize SYSREC and write the frame records.
- MVS systems use the IFCDIP00 service aid program to initialize SYS1.LOGREC and write the frame records.
- After system initialization, VM uses the CLEARF operand of the CPEREP command to reinitialize the error-recording area and update the frame records.

The frame records cannot be removed from the ERDS by the ZERO parameter; you must reinitialize the data set in order to erase or update them. See Chapter 24, "System Control Programs (SCPs)."

For detailed information about the 303X frame records, see the maintenance documentation for the 3031, 3032, and 3033 processors, and for the 7443 Service Record File.

Note: When you copy the ERDS to a history data set, you copy every set of frames; if you copy the ERDS repeatedly, you are copying the same frames over and over again.

SET RF = CREATE under VSE

The SET RF=CREATE command creates and initializes or reinitializes the VSE recorder file. If there are frames present, the initialization routines write them on SYSREC and note their presence in the SYSREC header record.

When the frames must be updated, you must re-IPL the system and re-issue the SET RF=CREATE command; the initialization routines recreate the SYSREC header, update the frame records, and set the rest of the file to hexadecimal zeros.

IFCDIP00 under MVS

The IFCDIP00 service aid program sets up LOGREC initially and also reinitializes it. See *SPL: SYS1.LOGREC* for more information.

CLEARF under VM

In order to update 303X frame records in the VM error-recording area, you must use the CLEARF operand with CPEREP. See "Managing Error Data" on page 3-1 for general information about clearing the error-recording area.

There are some usage notes specific to the CLEARF operand:

- You cannot use CLEARF with any other operands. Therefore, you should make sure you have captured pertinent error area information before issuing CPEREP with CLEARF.
- The service support console must be in SRF mode.
- Before using CLEARF, you must access the service record file device for the processor. The method differs depending on the processor model:
  - For the 3031 and 3033:
    1. Enable the I/O interface for the service support console.
    2. Activate the C1 frame on the service support console.
    3. Select the SRF mode - A2.
    4. VARY the SRF online.
    5. Attach the SRF to the console (user class F) running EREP.
  - For the 3032:
    1. Enable the I/O interface for the SRF from which you want to read a frame.
    2. Go to frame 31.
    3. Leave the displayed frame untouched.
    4. Run CPEREP.

- You must use CLEARF to update the frame records in the error-recording area after the installation of engineering changes to the processor(s) and channels.

Bibliography

The following publications contain more detailed information about specific processors.

- *IBM 3031 Maintenance Guide*, SY25-0517.
- *IBM 3032 Maintenance Guide*, SY25-0518.
- *IBM 3033 Maintenance Guide*, SY25-0519.

3090 Processor

Special Considerations for EREP Reports

The 3090 processor is not included in the System Exception report series. Report requests are the same as those listed under "Special Considerations for EREP Reports" on page 20-1.

Special Considerations for EREP Controls

3090 is a valid model number for the CPU and MOD selection parameters.

Other Considerations

- Depending on the mode it is running in, the 3090 can generate the following record types:
  - MCH
  - CCH (370 mode)
  - SLH/CRW (370-XA mode)
- The Software Recovery Status bits may appear in the Detail Edit report for the 3090. The validity of the information depends upon the operating system you are running under.

  Following are excerpts from the Detail Edit reports for MCH, CCH and SLH records showing the Software Recovery Status information. For examples of the entire reports see Chapter 2, "EREP Reports":

CCH	Figure 2-23 on page 2-64
MCH	Figure 2-29 on page 2-69
SLH	Figure 2-42 on page 2-83

```
.
.
.
             UNIT STATUS                                    CHANNEL STATUS
     ATTENTION                          0        PROGRAM CONTROLLED INTERRUPT      0
     STATUS MODIFIER                    0        INCORRECT LENGTH                  0
     CONTROL UNIT END                   0        PROGRAM CHECK                     0
     BUSY                               0        PROTECTION CHECK                  0
     CHANNEL END                        0        CHANNEL DATA CHECK                0
     DEVICE END                         0        CHANNEL CONTROL CHECK             1
     UNIT CHECK                         0        INTERFACE CONTROL CHECK           0
     UNIT EXCEPTION                     0        CHAINING CHECK                    0

         SOFTWARE RECOVERY STATUS
     HARD FAIL                          0   1
     DEGRADE FAIL                       1   2
     SOFT FAIL                          0   3
     PASSED                             0   4
.
.
.
```

Figure 20-1. CCH Detail Edit Report with Software Recovery Status

```
.
.
.
FAILING STORAGE ADDRESS:    NOT APPLICABLE
REGION CODE:                NOT APPLICABLE
EXTERNAL DAMAGE CODE:       NOT APPLICABLE

SOFTWARE RECOVERY STATUS
     HARD FAIL                          0   1
     DEGRADE FAIL                       0   2
     SOFT FAIL                          1   3
     PASSED                             0   4

NOTE: THE PRODUCT FUNCTIONAL CHARACTERISTICS PUBLICATION DESCRIBES THE MACHINE CHECK INTERRUPT CODE SUPPORT.
.
.
.
```

Figure 20-2. MCH Detail Edit Report with Software Recovery Status

```
  .
  .
  .
---UNIT STATUS----   SUB-CHANNEL STATUS   --------------------------SCSW FLAGS-----------------------
                                                 FLAG 0               FLAG 1               FLAG 2
ATTENTION          0  PGM-CTLD IRPT     0  CCW FORMAT       0  RESERVED          0  SUBCHANNEL ACTIV 0
STATUS MODIFIER    0  INCORRECT LENGTH  0  PRE-FETCH CCW    0  SSCH FUNCTION     1  DEVICE ACTIVE    0
CONTROL UNIT END   0  PROGRAM CHECK     0  INIT STATUS      0  HSCH FUNCTION     0  SUSPENDED        0
BUSY               0  PROTECTION CHECK  0  ADDR LIMIT       0  CSCH FUNCTION     0  ALERT STATUS     1
CHANNEL END        0  CHAN DATA CHECK   0  SUPP SUSPEND INT 0  RESUME PENDING    0  INTERMED STATUS  0
DEVICE END         0  CHAN CTL CHECK    1  ZERO COND CODE   0  START PENDING     1  PRIMARY STATUS   1
UNIT CHECK         0  I/F CTL CHECK     0  EXTENDED CONTROL 1  HALT PENDING      0  SECONDARY STATUS 1
UNIT EXCEPTION     0  CHAINING CHECK    0  PATH NOT OPER    0  CLEAR PENDING     0  STATUS PENDING   1

----SOFTWARE RECOVERY STATUS-----

HARD FAIL                     1  1
DEGRADE FAIL                  0  2
SOFT FAIL                     0  3
PASSED                        0  4
  .
  .
  .
```

Figure 20-3. SLH Detail Edit Report with Software Recovery Status

Notes:

1. *Operation not recovered.*

2. *Operation recovered but hardware resource lost.*

3. *Operation recovered and no resource lost.*

4. *VM passed error to guest*

Bibliography

- *IBM 3090 Processor Complex Functional Characteristics* is the general-purpose manual for customers.

- *IBM 3090 Processor Complex Introduction and Locations*, System Volume A01, includes a description of the rest of the 3090 library for IBM Customer Engineers.

3090 Vector Facility

Special Considerations for EREP Reports

The Vector Facility is considered part of a 3090 processor complex rather than a stand-alone product. To see records associated with the Vector Facility, request the various reports listed under "Special Considerations for EREP Reports" on page 20-1. The machine-check (MCH) Detail PRINT reports include information specific to the product.

Special Considerations for EREP Controls

You cannot specify the Vector Facility on any of the EREP selection parameters. If it is installed as part of your 3090 processor complex, it is automatically included in MCH records.

Other Considerations

The MCH records associated with the Vector Facility show two new bit settings in the machine-check interrupt code (MCIC) portion:

- Offset 48 (X'30'), Bit 6, if on, indicates a Vector Facility failure.
- Offset 48 (X'30'), Bit 13, if on, indicates that the Vector Facility is the source of the record.

EREP identifies and formats these bits under ADDITIONAL MCIC FLAGS in the model-dependent part of the MCH Detail Edit report.

Bibliography

- *IBM 3090 Processor Complex Functional Characteristics* is the general-purpose manual for customers.

- *IBM 3090 Processor Complex Introduction and Locations*, System Volume A01, includes a description of the rest of the 3090 library for IBM Customer Engineers.

9373, 9375 and 9377 Processors

Special Considerations for EREP Reports

The 9373, 9375 and 9377 processors are included in the System Exception Report Series. The reports are similar to those described in Part 1 of this book; the titles, and some headings, are specific to these processors.

Special Considerations for EREP Controls

9373, 9375 and 9377 are valid model numbers for the CPU and MOD selection parameters.

Other Considerations

The 9373, 9375 and 9377 processors run in 370 mode only. They can generate the following record types:

- MCH
- CCH

Chapter 21. Punched Tape Devices

The information in this subsection could be changed or superseded by newer information added under specific device types. If you are interested in a new device or machine number, look under the specific number.

Special Considerations for EREP Reports

Useful reports for these devices are:

SYSUM
TRENDS
EVENT
PRINT=PT or PS with DEV=nnnn and TYPE=OH (OBR and MIH records).

Care should be taken when requesting reports other than these as the results could be misleading.

Special Considerations for EREP Controls

None.

Devices Supported by EREP

These punched tape devices are valid for DEV=

1012 paper tape punch
1017 paper tape reader
1018 paper tape punch
2671 paper tape reader

Chapter 22. Teleprocessing (TP) Devices

The information in this subsection could be changed or superseded by newer information added under specific device types. If you are interested in a new device or machine number, look under the specific number.

Special Considerations for EREP Reports

- Useful reports for these devices are:

  SYSUM
  EVENT
  TRENDS
  PRINT=PT or PS with DEV=nnnn and TYPE=OTH (OBR, MDR and MIH records)

  Care should be taken when requesting reports other than these as the results could be misleading.

  Some of the devices listed below may produce different record types. In that case, request that record type when requesting Detail Edit and Summary (PRINT) reports.

- In OBR records, EREP sees the 3725 communications controller as a 3705. Therefore, if you want to isolate an OBR record from a 3725 controller, you must request the Detail report using DEV=3705 and TYPE=O.

- In MDR records, the 3725 has its own device code, so you can select records by coding DEV=(3725) and TYPE=T.

- In the MDR Detail Summary report, the LIB ADDR field contains the line interface base address for 3705s. If the field is all zeros, it means the error is in the device rather than in the line.

Special Considerations for EREP Controls

- The LIA/LIBADR and TERMN parameters are for use with TP devices. LIA/LIBADR is for 3705/3725 communications controllers, and TERMN is for 2700 terminals and 3705 controllers.

- In the case of the TERMN parameter, EREP does not limit the device or record type in response to the parameter alone. You must also code TYPE=OT and DEV=(27XX,3705) to limit a report to TCAM and/or VTAM records from terminals with the specified name(s).

Other Considerations

In 3705 and 3725 communications controllers, the NCP does not recognize and pass on XA-specific MDR record information. Instead, the 3725, for example, always records 370-mode MDR records. This is true even when the device involved is generating XA-mode records.

Devices Supported by EREP

These teleprocessing products are valid for DEV=

1030 data collection system
1050 data communications system
1060 data communication system
1130 computing system
115A Western Union terminal
2701 data adapter
2702 transmission controller
2703 transmission controller
2715 transmission controller
2740 communication terminal
2741 communication terminal
2760 optical image unit
2770 data communication system
2780 data transmission terminal
2790 data communication system
27XX
2947 check collection controller
2970 banking terminal
2972 station control unit
3670 brokerage branch office system
3700 terminal
3704 communications controller
3705 communications controller
3725 communications controller
3735 programmable buffered terminal
3791 cluster controller
37XX
3945 station control unit
3968 terminal controller
83B3 AT&T terminal

Note: 27XX and 37XX are general device type designations that include families of IBM communications devices and controllers.

Bibliography

The following publications provide detailed information about the devices:

- *3705 Communications Controller Theory and Maintenance Manual*, SY27-0107.
- *3705-80 Communications Controllers Theory and Maintenance Manual*, Volume 1, SY27-0208.

Chapter 23. Other Devices

The information in this subsection could be changed or superseded by newer information added under specific device types. If you are interested in a new device or machine number, look under the specific number.

Special Considerations for EREP Reports

The channel-to-channel adapter, CTCA, appears as “CACA” on report output, because the characters must be translated to hexadecimal digits.

Special Considerations for EREP Controls

None.

Devices Supported by EREP

These device and machine type designations are valid for DEV=

2280 high speed microfilm output film recorder
2282 film recorder/scanner
2495 magnetic tape cartridge reader
2930 tape inter-system connection unit
2955 remote service terminal
2956 badge and badge/card reader
3838 array processor
3848 cryptographic unit
7443 service recording facility
7770 audio response unit
7772 audio response unit
BA00 serial OEM interface adapter
CTCA channel-to-channel adapter

In addition, EREP also recognizes the following “unknown” device types:

2101
3703
3967
125D

BA00 Serial OEM Interface Adapter

Special Considerations for EREP Reports

Useful reports for this device are:

SYSUM
EVENT
TRENDS
PRINT=PT or PS with DEV=(BA00) and TYPE=O.

Care should be taken when requesting reports other than these as the results could be misleading.

Special Considerations for EREP Controls

- "BA00" is valid for DEV=

Other Considerations

- The SOEMI adapter generates OBR records. The OBR code for the device is X'1014'.

Bibliography

- *Serial OEM Interface (SOEMI) Description and Programmer's Reference*, GA33-1585, describes the interface that dictates the characteristics of the SOEMI adapter.

Chapter 24. System Control Programs (SCPs)

The information in this subsection could be changed or superseded by newer information added under specific SCPs. If you are interested in a new system release, look under the individual SCP.

This section of the book contains information specific to each of the three families of operating systems that use EREP. The SCP-specific considerations presented here include:

- The creation and processing of software records
- The initialization of the error-recording data set (ERDS)
- Other notes that may be of help

Considerations for EREP Reports:

Useful reports for these devices are:

SYSUM
TRENDS
EVENT
SYSEXN
PRINT=PT or PS with whichever of the following record types the SCP generates:

E Recovery/termination (Formerly EOD)
H Missing interrupt (MIH)
I System initialization (IPL)
S Software (SFT)

Care should be taken when requesting reports other than these as the results could be misleading.

Information about the VSE SCP

Your VSE (formerly DOS) operating system may not include the latest level of EREP. The VSE and VSE/Advanced Functions systems currently are packaged with EREP Version 3 Release 1.

However, if the SCP for your installation is older than VSE/Advanced Functions Release 2.0 or VSE Release 1.3.5, it could include only EREP Version 1 Release 1. If this is the case, you cannot get System Exception reports, or other reports for the newer devices. Even though these things are documented in this book, they do not apply to your installation.

The Creation and Processing of Software records

The VSE systems do not create type X'4X' software (SFT) records in response to abnormal termination. They do, however, record events closely associated with system operation, in system initialization (IPL) and termination (EOD) records. These records are created by the Reliability Data Extractor (RDE), a system component that is standard in VSE/Advanced Functions and optional in earlier VSE systems.

EREP processes these records for the System Summary and Trends reports, grouping them among PROGRAM ERRORs. IPL and EOD records are also included in the System Error Summary report of the System Exception series, under IPL/RESTART/TERMINATION.

The IPL record includes a reason code and a subsystem code, supplied by the operator as part of the interactive IPL process. These two codes help identify the reason for the IPL and the device or program (if any) that failed.

The EOD record is written in response to a ROD (record on demand) command issued before shutting down the system for the day. The ROD command forces the dumping of statistical data counters and buffered logs to SYSREC, thus preserving the latest environmental data about your hardware systems. It also causes RDE to write the end-of-day record to SYSREC.

See Chapter 10, "Error Records for EREP" for the formats of IPL and EOD records, and *VSE/Advanced Functions Operating Procedures* for information about RDE messages associated with IPL and end of day.

Initialization of the Error Recording Data Set (ERDS)

The VSE error-recording data set is called the system recorder file, or SYSREC, and is assigned the logical name IJSYSRC. SYSREC is created on the SYSRES volume during the first IPL following system generation. The SYSRES volume also contains the hard copy file (IJSYSHC) and the system history file (IFSYSHF).

In this book, SYSREC refers only to the system recorder file, or ERDS.

The operator creates and initializes SYSREC by issuing the IPL command SET RF=CREATE. As initialized, SYSREC consists of a header record and, if needed, frame records for 303X processors. See "ERDS Formats" in Chapter 10, "Error Records for EREP" for the contents of the SYSREC header. Unless the file itself is damaged, you should not need to reinitialize it except at system generation.[1]

You cannot reinitialize SYSREC without re-IPLing the system and re-issuing the SET RF=CREATE command. The EREP control parameter ZERO and the special EREP program IFCOFFLD merely zero out the data set; they do not remove the header or the 303X frame records, if present.

See your *System Management Guide* for detailed information about the kinds and amounts of disk space required for SYSREC, and about creating this VSE system file.

Other VSE Notes

- 303X frame records on SYSREC might have to be subdivided because of main storage space limitations. The headers for both machine-check and channel-check frame records contain switches at offsets 3 and 4 that help keep track of the subdivided records. The mapping for the switches follows.

  Byte 0 Sequence number of this frame within a set of frames.

  Byte 1
  Bit 0 Last record of a frame.
  Bit 1 Set for all records of the last frame in a set of frames.

- Do not confuse the EREP history data set with the VSE history file named IJSYSHF.

  The VSE history file contains information about the components of the system and the program fixes applied to those components. It is updated by MSHP (Maintain System History Program) and reflects the change level of your system.

  The EREP history data set contains error records, either copied directly from SYSREC or accumulated after a report is run. It can be a cumulative data set, updated daily or weekly. It can be used as input to the EREP program, either by itself or in combination with SYSREC. Note that the EREP history data set is not created by the system; it is created when you specifically request it during an EREP run.

1 If your system is running on a 303X processor, you must examine (via EREP) and recreate the recorder file whenever an engineering change is installed that affects the frame records.

Bibliography

The following publications have more information about the functions of the VSE systems discussed here:

- *VSE System Data Management Concepts*, GC24-5209.
- *VSE/Advanced Functions System Management Guide*, SC33-6094.
- *VSE/Advanced Functions System Control Statements*, SC33-6095.
- *VSE/Advanced Functions Operating Procedures*, SC33-6097.
- *Device Support Facilities User's Guide and Reference*, SC35-0033.

Information about the MVS SCP

The Creation and Processing of Software (SFT) Records

VS1 VTAM and MVS software records — X'4X' reflect software abends of both application and system programs. IBM system components' recovery routines create SFT records whenever IBM code is known or suspected to be the cause of a failure. The records contain data about SCP failures, operator-initiated restarts, and, for MVS, about program damage caused by machine checks. The software record consists primarily of the system diagnostic work area (SDWA), a control block built during error recovery processing that contains all the pertinent data about the failure the recovery routine is able to find.

See "Software (SFT) Error Record" on page 10-46 for more information about the software record produced by MVS systems.

Initialization of the Error Recording Data Set (ERDS)

SYS1.LOGREC is created and initialized at system generation by the disk initialization program, IFCDIP00. In all MVS systems except MVS/XA, SYS1.LOGREC must reside on the system residence volume.

As initialized, LOGREC consists of a header record and, possibly, frame records for 303X processors,[2] followed by space for error and environmental records. See "ERDS Formats" in Chapter 10 for the contents of the LOGREC header.

If necessary, you can run the IFCDIP00 service aid to reinitialize SYS1.LOGREC. Only IFCDIP00 can update the 303X frame records. You can also use IFCDIP00, together with the IEHPROGM utility, to reallocate the SYS1.LOGREC data set. See *SPL: SYS1.LOGREC* for details.

Other MVS Notes

When running EREP under an MVS system, you can combine history data sets as input to EREP simply by concatenating DD statements for them to the ACCIN DD statement. You will need to make sure the space allocated for the DIRECTWK data set is large enough to hold all the input records, as EREP copies the records to DIRECTWK before starting its selection processing.

2 In MVS's LOGREC, the header is followed by a time-stamp record for use in IPL records.

Running EREP in a Multisystem Environment

In a multisystem environment, where I/O devices are shared between processor systems, special care is required to make sure you get complete and accurate reports about the shared devices. Some suggested procedures, for MVS-MVS installations, are:

- Put all DASDID, LIMIT and SHARE statements into a separate data set.
  - Specify the data set name on the SYSIN DD statement for System Summary, System Exception, Trends, Threshold, and any PRINT reports for shared I/O devices.
  - Do not use PARM='CARD' in these steps; instead, code the EREP parameters on the EXEC statement.
- Reorder the EREP job steps shown in Part 2 to run the processor and SCP Detail reports *before* the system-level reports. For example:
  1. Offload LOGREC to a working data set without requesting any printed report.
  2. Run MCH and CCH Detail PRINT reports.
  3. Run software Detail PRINT reports.
  4. Run Detail PRINT reports for dedicated I/O devices (for example, 2305).
  5. Run Event History against the working data set.
  6. Concatenate the history data sets from each system on the ACCIN DD statement, and run the following reports:
     - System Summary
     - System Exception
     - Detail PRINT reports for shared I/O devices.

     *Note: You may not want to run **all** of these reports every time; select the ones you need, deleting the steps that request reports you do not want.*
  7. Add the records from the concatenated data sets to any existing "permanent" history data set.
- Develop a technique to make sure that each system's LOGREC has been copied before the first step that uses concatenated input runs. For example, include a step that creates a named data set, then test for that data set before requesting the first system-level report.
- Install this procedure on each system in the complex, so reports can be run from any one of them at any time.

Bibliography

The following publications have more information about the functions of the MVS systems discussed here:

- *MVS/XA JCL*, GC28-1148.
- *MVS/XA SPL: SYS1.LOGREC Error Recording*, GC28-1162.
- *MVS/XA Data Management Services*, GC26-4013.
- *MVS/370 SPL: Data Management*, GC26-4056.
- *MVS/370 Data Management Services*, GC26-4058.
- *OS/VS1 Data Management for System Programmers.*
- *OS/VS1 Data Management Services Guide.*
- *OS/VS1 SYS1.LOGREC Error Recording*, GC28-0668.
- *OS/VS1 JCL Reference*, GC24-5099.
- *OS/VS2 MVS SPL: SYS1.LOGREC Error Recording*, GC28-0677.
- *OS/VS2 MVS JCL*, GC28-1300.

Information about the VM SCP

VM Error Recording

The input to EREP through VM can be quite different from the input used by the other systems, because VM creates records differently.

Both the VSE and MVS systems write records to their ERDS via SVC 76. When a VSE or MVS system is running in a virtual machine, VM can tell when the guest system issues an SVC 76, and can divert the record to its own error-recording area. In the process, VM translates the virtual address of the device originating the record to a real address, so the records are meaningful to a user.

However, VM does not divert every record created by a guest system to the error-recording area. Some records it "reflects back" to the virtual machine to be recorded on the guest system's ERDS. The records reflected back could be from devices dedicated to the virtual machine, or could be of certain types, or could contain an error. When VM reflects a record back to the virtual machine, the addresses in the record remain virtual.

This means that sense data logged for I/O error conditions, if reflected back to the virtual machine, is associated with a logical device rather than the actual device. Such sense data is of little use in identifying problem devices.

VM's reflecting back of some records to the virtual machine also means that it is possible for your installation's error records to be divided between the VM error-recording area and your operating system's ERDS.

Figure 10-6 on page 10-12 shows all the records created by VSE or MVS systems, and all the records VM records in its own error-recording area. Here is a list of the records VM does ***not*** record on behalf of the guest system:

- Machine Check Type 13: MCH in multiple storage environment
- Channel Check Type 21: CCH in multiple storage environment
- Unit Check Types 34 and 36: TCAM and VTAM OBRs
- Software (SFT), Type 4X
- System Initialization (IPL), Type 5X
- System Termination (EOD), Type 8X
- Miscellaneous Data/Non-Standard Type 90: MDR from SVC 91

Note: VM creates its own type 2X CCH error record before detecting the guest system's SVC 76 for that record type. Thus, even though it reflects the type 21 record back to the virtual machine, the incident is recorded in the error-recording area.

Capturing All the Data for EREP

When CPEREP runs EREP for you, it uses the records in the VM error-recording area as the only input (unless you use the HIST or MERGE operand). The SERLOG FILEDEF, which implies SYSREC or SYS1.LOGREC input, is only a simulation of that data set, required because of format differences between the error-recording area and the system ERDS. No SERLOG file actually exists, and CPEREP uses neither SYSREC nor SYS1.LOGREC as the source of record input for EREP.

The result of this could be misleading report output, because the VM error-recording area did not contain all the records that would have been on SYSREC or LOGREC. This can be a problem especially with OBR records for your TP devices: if you run EREP only under VM, you will not see these records in any report output, and you might be missing some errors.

One way to make sure you get reports about all the possible errors in your system is to run EREP under VM and then run it again under the guest operating system — VSE or MVS. The second EREP run would include data not recorded by VM.

Another way to make sure you are seeing all your error records in the EREP reports is to combine the data from your system's ERDS and the error-recording area before requesting any reports. Then run EREP under either system using the combined records as history input.

Initialization of the Error Recording Data Set (ERDS)

When the VM ERDS is on a count-key-data device, it consists of at least two adjacent cylinders allocated on the system residence pack. Accordingly, the VM ERDS is sometimes referred to as the "error-recording cylinders."

When the VM ERDS is on a fixed-block-architecture device, it consists of any number of adjacent pages assigned on the system residence volume. Accordingly, the VM error-recording routines see the ERDS as a series of logical pages.

At system initialization, the error-recording area is initialized by CP routines that format each of the recording cylinders and set up the logical pages to receive error records. Each recording cylinder has a header and, possibly, 303X frame records, followed by the space for error records.

To reinitialize the error-recording area, you issue the CPEREP command specifying CLEAR or CLEARF as the only operand. CLEAR creates a new header and re-formats each cylinder; CLEARF does the same, and also replaces the frame records with the latest frames from the 303X Service Recording File. The CLEARF operand is the only way to update the frames in the error recording area.

See "VM Notes" on page 4-51 for information about using the CLEARF operand.

See "ERDS Formats" in Chapter 10 for the contents of the error-recording cylinder header. For other details about VM error recording, see the *OLTSEP and Error Recording Guide* for your VM facility.

Bibliography

The following publications have more information about the VM functions mentioned here:

- *VM/SP: OLTSEP and Error Recording Guide*, SC19-6205.
- *VM/SP HPO: OLTSEP and Error Recording Guide*, SC19-6230.
- *VM/370: OLTSEP and Error Recording Guide*, GC20-1809.

Glossary

The following terms are defined here so that all can agree on what a word is supposed to mean in this book. Some of the definitions are taken from the *IBM Vocabulary for Data Processing, Telecommunications, and Office Systems*; others are from *Webster's Third New International Dictionary*. Still others are derived from the way the words are used in this book.

A

acronym. A combination of letters, usually the first ones, that stand for a multi-word name or expression. For example, "EREP" is the acronym for IBM's Environmental Record Editing and Printing program.

C

CCF. Channel-check frame. The record on the ERDS that EREP uses to format channel-check records from the 303X group of processors.

CCH. Channel-check handler. A S/370 hardware feature that, when a channel error occurs, records information about the error and issues a message to the operator. In VSE, machine check analysis and recording (MCAR) performs a similar function. The records created in both cases are called CCH records.

CE. IBM Customer Engineer.

Channel. The physical connector between a processor and an input/output device, usually via a control unit of some kind. In the case of the extended architecture (System 370/XA), the hardware channels are replaced by subchannels, which are capable of dynamic variation controlled by microcode in the processor complex.

In this book, we refer to "subchannels" when talking about fields in 370-XA report output, for example, but use "channel" in the general sense to mean the connection between controller and device.

Code. The programming-language instructions that make up a computer program. As a verb, "to code" is the same as "to write code."

Controller. A single unit which provides an interface between one or more SCUs and a group of devices. It usually resides within the same unit as the lowest drive addresses.

CRW. Channel-report word. In S/370-XA, the channel-report word is part of the channel-subchannel recovery mechanism. It contains information about channel incidents reported through machine checks, specifying the error environment and the severity of the error. MVS/XA builds a CRW record that, in combination with the subchannel logout handler (SLH) record, replaces the CCH record.

D

DDR. Dynamic device reconfiguration. A facility that allows a demountable volume to be moved, and repositioned if necessary, without abnormally terminating the job or repeating the IPL procedure. The MVS operating systems create DDR records to provide information about operator-assisted recovery involving the relocation of tape and movable DASD volumes.

DOS. Disk Operating System. An obsolete name, replaced by VSE, Virtual Storage Extended. In this book, "VSE" includes and implies all releases of this operating system, from DOS to VSE/Advanced Functions.

E

ERDS. The error-recording data set; input to the IFCEREP1 program. In MVS systems, the ERDS is SYS1.LOGREC; in VSE systems, it is SYSREC; in VM, it is the error-recording area or cylinders.

ERP. Error-recovery program/processing; the system routines that detect and process errors, writing records to the ERDS.

H

Hard machine check or error. A hardware error that disables the processor or other unit.

I

Installation. A data processing system location: a computer center housing processors, I/O devices, other hardware devices, the software that controls the machines, and the people that control the computer center.

IPL. Initial program load. The process by which an operating system is initialized at the beginning of the day or session. At IPL, the system operator enters the installation-specific information the operating system must have in order to manage the installation's computing system and handle the installation's application programs. This information includes system parameters, system data set definitions, and other information needed so the operating system can begin operating.

M

MCF. Machine-check frame. The record, on the ERDS, that EREP uses to format machine-check records from the 303X group of processors.

MCH. Machine-check handler. A S/370 hardware feature that analyzes errors and attempts recovery by retrying the failing instruction. If unsuccessful, it causes an interrupt that triggers the creation of an error record. In VSE systems, machine check analysis and recording (MCAR) performs similar functions. The records created in either case are called MCH records.

MDR. Miscellaneous data record. A record type that records error and usage information from buffered control units or communications controllers, and device failures on TP devices connected to 3705/3725 communications controllers. The record is created when there is an overflow of statistical counters; its purpose is to provide more information about the accompanying failure.

MIH. Missing-interrupt handler. An MVS and MVS/XA facility that keeps track of I/O interrupts, informing the operator and creating a record whenever an expected interrupt fails to occur in a pre-set time interval.

MVS and MVS/XA. Multiple Virtual Storage and Multiple Virtual Storage/Extended Architecture, two versions of the System/370 operating system that are extensions of OS/VS2.

In this manual the term "MVS" is used to refer to a family of operating systems that control IBM System/360 and System/370 computing systems. "MVS" includes OS/VS1, OS/VS2, MVS/370, and MVS/XA.

O

OBR. Outboard recorder. In VSE systems, the outboard recorder is a feature that records pertinent data about an unrecoverable I/O error. MVS systems create a similar record from information recorded when an I/O device is in *unit-check* status. The resulting record in both cases is called an OBR record.

OS/VS. Operating System/Virtual Storage. A family of operating systems that control IBM System/360 and System/370 computing systems. OS/VS includes VS1, VS2, MVS/370, and MVS/XA.

In this book, these operating systems are referred to by the general term "MVS."

OS/VS1. Virtual Storage 1. One of the MVS operating systems.

OS/VS2. Virtual Storage 2 (MVS, Version 1). MVS/370; one of the MVS operating systems.

P

PCT. Product control table. The internal table that contains the data EREP needs in order to identify and process records from a particular IBM device or product.

PSR. IBM Programming Service Representative. The person responsible for helping you maintain the IBM software in your installation.

R

RCT. Record control table.

S

S/370 and S/370-XA. System/370. The computing systems built around large IBM processors. XA stands for Extended Architecture, the architecture basis for the 3081 and later processors, characterized by 31-bit addresses. S/370 implies not only the processor but also the many other data processing devices that can be connected to it to make a 370 (or 370-XA) data processing *system*.

SCP. System control program. The minimum software package that will make your operating system work.

SCU. Storage control unit. A functional unit which resides between channel(s) and controller(s).

SLH. Subchannel-logout handler. A S/370-XA feature that provides detailed model-independent information relating to a subchannel; the subchannel logout describes equipment errors detected by the channel subsystem. MVS/XA builds an SLH record that, in combination with the CRW record, replaces the CCH record.

Soft machine check or error. A hardware error that is not disabling.

Subchannel. The extended architecture version of "channel"; see the explanation of "channel" in this Glossary, and the *Principles of Operation* for System/370 Extended Architecture.

Subsystem. In hardware terms, a group of devices that function together to perform I/O operations. An I/O subsystem can consist of a control unit (controller) and its associated drives — either disk or tape; or it can consist of *all* the DASD or tape storage — including drives and controllers — in an installation. In the case of newer DASD, the I/O subsystem also includes storage control units (SCUs) and storage directors (SDs), within the controller.

Syntax. The relationships among the elements and characters in a parameter or language statement. For our purposes, the way you have to code something in order for the program to understand and accept it.

SYSGEN. System Generation. The process of selecting optional parts of an operating system and of creating a particular operating system tailored to the requirements of a data processing installation. Can also include I/OGEN, which is the time when the system programmer defines the installation's computing system configuration to the operating system.

V

VSE. Virtual Storage Extended. A family of disk operating systems that controls IBM System/360 and System/370 computing systems and includes VSE and VSE/Advanced Functions.

VS1. Virtual Storage 1. One of the OS/VS operating systems.

VS2. Virtual Storage 2 (MVS, Version 1). MVS/370; one of the OS/VS operating systems.

VM. Virtual Machine. A time-sharing system control program that manages the resources of an IBM System/370 computing system so that multiple remote terminal users have a functional simulation of the computing system (a virtual machine) at their disposal. In this book, "VM" means all versions of the Virtual Machine system control program, including VM/370, VM/System Product, VM/SP/High Performance Option, and VM/XA Migration Aid.

Abbreviations Used in EREP Output

BPI	Bits per inch
BUFE	Buffer error
BYTES RD/SRCHD	Megabytes read/searched
CCH	Channel check handler
CCH-CRH	CCH-channel reconfiguration hardware
CCH-INC	CCH incomplete record
CDDA	Command data
CHK	Check
CHNL	Channel
CHP	Channel path ID
CHPID	Channel path ID
CK	Check
CLNACT	Cleaner action
CMD	Command
CMND	Command
CNT	Count
CNTRL	Control
CNTRLR	Controller
COMP	Component
CONS + UR	Console plus unit record
CORR	Correctable
COR	Corrected
CP	Central processing unit; interchangeable with "processor"
CPU	Central processing unit
CRW	Channel report word
CSECTID	Control section (CSECT) identification
CSID	Channel set ID
CSW	Channel status word
CT	Controller; count
CTLID	Controller ID
CTLR	Controller
CU	Control unit
CUA	Channel-control unit-device address
DATAXFR	Data transfer
DATA CKS CORR/RTRY	Data checks correctable/retry
DDR	Dynamic device reconfiguration
DDR-OPR	DDR-Operator requested
DDR-SYS	DDR-System requested
DEV	Device number
DEVNO	Device number
DEVNUM	Device number
DEVT	Device type

DNO	Device number
DRCT	Storage director
DTE	Date
EOD	End of day
EQUCHK	Equipment check
EQUIP	Equipment
ERDS	Error-recording data set: SYS1.LOGREC (MVS), SYSREC (VSE), error recording area (VM)
ERROPS	Error operations
ERSGAP	Erase gap
ESW	Extended status word
EXTD	External damage
FCG	Floating channel group
FLG	Flag
FMT	Format
HDR SER	Header (tape)/serial number of drive that created tape
HIRS	Hardware instruction retry (successful)
ID	Identification
INV	Invalid
INVK	Invoked
IPL	Initial program load
IRB	Interrupt response block
LEN	Length
MB	Megabyte
MBYTE	Megabyte
MCH	Machine check handler
MCH-TRM	MCH-System terminated
MCK	Machine check
MDR	Miscellaneous data record
MDR-DAS	DASD MDR record
MIH	Missing interrupt handler
MIH-CE	MIH-channel end pending
MIH-DE	MIH-device end pending
MOD	Module
OBR	Outboard record
OBR-DMT	OBR-demount record
OBR-DPA	OBR-dynamic pathing availability
OBR-EOD	OBR-End-of-day
OBR-PRM	OBR-Permanent error record
OBR-SHT	OBR-Short record
OBR-TMP	OBR-Temporary error
OVERRN	Overrun
OVERRUN CDDA	Overrun command data
OVRN	Overrun
PCUA	Primary channel-control unit-device address
PERM	Permanent
PFU	Probable failing unit
PRGM INT	Program-initiated
PRI	Primary
PRM	Permanent
PROG-EC	Program-extended control mode
PSW	Program status word
RCVRYXIT	Recovery exit module
RD(S)	Read error(s)
REC-TYP	Record type

RTN	Routine
RTRY	Retry
SCSW	Subchannel status word
SCP	System control program
SCU	Storage control unit
SCUA	Secondary channel-control unit-device address
SCUID	Storage control unit ID
SD	Storage director
SEC	Secondary
SEEKS CNTR/HH	Seek errors cylinder track/head
SFT	Software (record)
SLH	Subchannel logout handler
SFT-ABN	SFT-ABEND record
SFT-MCH	SFT-machine error, recoverable
SFT-PI	SFT-program interrupt
SFT-RST	SFT-restart
SIO	Start I/O
SKS	Seeks; data access errors
SNID	Sense path group ID (DPA)
SPID	Set path group ID (DPA)
SRCHD	Searched
SSYS ID	Sub-system identifier
STOR	Storage error
TEMP	Temporary
TERM	Terminal
TMP	Temporary
UCB	Unit control block
VOLID	Volume serial number
WRT(S)	Write error(s)

Associated Publications

The following pages list the IBM manuals either referred to in this book or associated with this book and the use of EREP.

For DOS/VS and DOS/VSE Users:

- *DOS/VSE System Control Statements*, GC33-5376. Contains detailed information about the system controls required when running EREP under DOS/VSE.
- *DOS/VSE Messages*, GC33-5379. Lists and interprets the messages that IBM's DOS/VSE issues to the operator and the programmer.
- *DOS/VSE Serviceability Aids and Debugging Procedures*, GC33-5380. Aids System/370 operators and programmers in determining and isolating the cause of a DOS/VSE system malfunction.
- *DOS/VSE Error Recovery and Recording Transients Logic*, SY33-8552. Provides general information and detailed flowcharts of the Recovery and Recording Transient Programs of DOS/VSE.

For VSE/Advanced Functions (VSE/AF) Users:

- *VSE/Advanced Functions, System Management Guide*, SC33-6191. A guide for users of the VSE/AF operating system. It includes information about using EREP under VSE and VSE/AF.
- *VSE/Advanced Functions, System Control Statements*, SC33-6198. The VSE/AF version of SC33-5376.
- *VSE/Advanced Functions, Messages and Codes*, SC33-6098. The VSE/AF version of SC33-5379.
- *VSE/Advanced Functions, Diagnosis: Service Aids*, SC33-6195. The VSE/AF version of SC33-5380.
- *VSE/Advanced Functions, Diagnosis Reference: Error Recovery and Recording Transients Logic*, LY33-9108. Provides general information and detailed flowcharts of the Recovery and Recording Transient Programs of VSE/AF.

For VSE/System Package (VSE/SP) Users:

- *VSE/System Package, Messages and Codes*, SC33-6181. Interprets the messages that VSE/SP issues to the operator and the programmer.

For OS/VS1 Users:

- *OS/VS1 JCL Reference*, GC24-5099. Describes how to code job control language statements to override default parameters, use cataloged procedures, and allocate space for data sets.
- *OS/VS1 Utilities*, GC26-3901. Describes how to use utility programs to print certain types of service aid output and to allocate data sets with the IEHPROGM utility.
- *OS/VS1 SYS1.LOGREC Error Recording*, GC28-0668. Describes how to use the IFCDIP00 service aid program and how error records are built and recorded on SYS1.LOGREC.
- *OS/VS Message Library: VS1 System Messages*, GC38-1001. Describes the messages issued by the IFCDIP00 service aid program.
- *OS/VS1 SYS1.LOGREC Error Recording Logic*, SY28-0669. Describes the internal logic of

IFCDIP00 and the system recording routines: DDR recorder, the deferred incident recorder, MCH emergency recorder, MCH error recorder, miscellaneous data recorder, outboard recorder, SVC 76, SVC 91 and VTAM SYS1.LOGREC recorder.

- *OS/VS1 Recovery Management Support Logic*, SY24-5170. Describes the function and logic of the machine check handler, the channel check handler, and dynamic device reconfiguration.

For OS/VS2 (MVS/SP Version 1 or MVS/370) Users:

- *OS/VS2 MVS Utilities*, GC26-3902. Describes how to use utility programs to print certain types of service aid output, and to allocate data sets with the IEHPROGM utility.
- *OS/VS2 MVS SPL: SYS1.LOGREC Error Recording*, GC28-0677. Describes how to use the IFCDIP00 service aid program and how error records are built and recorded on SYS1.LOGREC.
- *OS/VS2 SPL: Debugging Handbook*, GC28-1047. Contains the control blocks used in MVS, including the SDWA (System Diagnostic Work Area).
- *MVS JCL*, GC28-1300. Describes how to use job control language statements; how to override default parameters, use cataloged procedures, and allocate space for data sets; and how to use JES2 and JES3 control statements with other JCL statements.
- *OS/VS Message Library: VS2 System Messages*, GC38-1002. Includes the messages issued by the IFCDIP00 service aid program.
- *OS/VS2 SYS1.LOGREC Error Recording Logic*, SY28-0678. Describes the internal logic of IFCDIP00 and the system recording routines: asynchronous recording facility, DDR/MIH recorder, MCH emergency recorder, OBR/MDR recorder, SVC 76, and SVC 91.
- *OS/VS2 System Logic Library, Volume 7*, LY28-1083. Describes the function and logic of the channel check handler, dynamic device reconfiguration, the machine check handler, and the missing interrupt handler.

For OS/VS2 (MVS/SP Version 2 or MVS/XA) Users:

- *MVS/Extended Architecture JCL*, GC28-1148. Describes how to use job control statements to override default parameters, use cataloged procedures, and allocate space for data sets; and how to use JES2 and JES3 control statements with other JCL statements.
- *MVS/Extended Architecture Message Library: System Messages*, GC28-1156. Includes the messages issued by the IFCDIP00 service aid program.
- *MVS/Extended Architecture SYS1.LOGREC Error Recording*, GC28-1162.
- *MVS/Extended Architecture Debugging Handbook, Volume 5*, GC28-1168. Documents the structure and contents of MVS/XA system control blocks, including the SDWA (System Diagnostic Work Area). Describes how to use the IFCDIP00 service aid program and how error records are built and recorded on SYS1.LOGREC.
- *MVS/Extended Architecture Data Administration: Utilities*, GC28-4018. Describes how to use utility programs to print certain types of service aid output, and how to allocate data sets with the IEHPROGM utility.
- *MVS/Extended Architecture SYS1.LOGREC Error Recording Logic*, LY28-1187. Describes the internal logic of IFCDIP00 and the system recording routines: asynchronous recording facility, OBR/MDR recorder, SVC 76 and SVC 91.
- *MVS/Extended Architecture System Logic Library, Volume 8*, LY28-1234 and LY28-1235. Describes the function and logic of the missing interrupt handler and the subchannel logout handler.
- *MVS/Extended Architecture System Logic Library, Volume 9*, LY28-1238. Describes the function and logic of the dynamic device reconfiguration recorder and the machine check handler.

For VM/370, VM/SP, and VM/SP HPO Users:

- *VM/SP: System Messages and Codes*, SC19-6204. The VM/System Product version of GC20-1808.
- *VM/SP OLTSEP and Error Recording Guide*, SC19-6205. The VM/System Product version of GC20-1809.
- *VM/SP HPO OLTSEP and Error Recording Guide*, SC19-6230. The VM/SP/High Performance Option version of SC19-6205.
- *VM/370 System Messages*, GC20-1808. Includes the system messages that relate to the VM/370 modules supporting EREP (DMSIFC AND DMSREA).
- *VM/370 OLTSEP and Error Recording Guide*, GC20-1809. Describes the way VM records error records originating in a virtual machine running under its control, as well as the way VM records errors occurring in its own environment.

For All Users:

- *IBM System/370 Principles of Operation*, GA22-7000. Explains in detail the machine functions of the System/370 processors.
- *IBM System/370 Extended Architecture Principles of Operation*, SA22-7085. The MVS/XA version of GA22-7000.

Index

D

F

N

O

P

R

S

Environmental Record
Editing and Printing Program
(EREP)

READER'S COMMENT FORM

GC28-1378-2

This manual is part of a library that serves as a reference source for systems analysts, programmers, and operators of IBM systems. You may use this form to communicate your comments about this publication, its organization, or subject matter, with the understanding that IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you.

Note: ***Copies of IBM publications are not stocked at the location to which this form is addressed. Please direct any requests for copies of publications, or for assistance in using your IBM system, to your IBM representative or to the IBM branch office serving your locality.***

Possible topics for comment are:

Clarity   Accuracy   Completeness   Organization   Coding   Retrieval   Legibility

If you wish a reply, give your name, company, mailing address, and date:

____

____

____

____

What is your occupation? ________________________________

How do you use this publication? ________________________________

Number of latest Newsletter associated with this publication: ________________

Thank you for your cooperation. No postage stamp necessary if mailed in the U.S.A. (Elsewhere, an IBM office or representative will be happy to forward your comments or you may mail directly to the address in the Edition Notice on the back of the title page.)

Note: Staples can cause problems with automated mail sorting equipment.
Please use pressure sensitive or other gummed tape to seal this form.

Cut or Fold Along Line

Environmental Record Editing and Printing Program (EREP)

GC28-1378-2 S370-37

Reader's Comment Form

Cut or Fold Along Line

Fold and tape Please Do Not Staple Fold and tape

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 40 ARMONK, N.Y.

POSTAGE WILL BE PAID BY ADDRESSEE

International Business Machines Corporation
Department A08, Building 035-3
740 New Circle Road NE
Lexington, Kentucky 40511

Fold and tape Please Do Not Staple Fold and tape

IBM®

Printed in U.S.A.

Environmental Record
Editing and Printing Program
(EREP)

READER'S
COMMENT
FORM

GC28-1378-2

This manual is part of a library that serves as a reference source for systems analysts, programmers, and operators of IBM systems. You may use this form to communicate your comments about this publication, its organization, or subject matter, with the understanding that IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you.

Note: ***Copies of IBM publications are not stocked at the location to which this form is addressed. Please direct any requests for copies of publications, or for assistance in using your IBM system, to your IBM representative or to the IBM branch office serving your locality.***

Possible topics for comment are:

Clarity Accuracy Completeness Organization Coding Retrieval Legibility

If you wish a reply, give your name, company, mailing address, and date:

____
____
____
____

What is your occupation? ______________________________

How do you use this publication? ______________________________

Number of latest Newsletter associated with this publication: ____________________

Thank you for your cooperation. No postage stamp necessary if mailed in the U.S.A. (Elsewhere, an IBM office or representative will be happy to forward your comments or you may mail directly to the address in the Edition Notice on the back of the title page.)

Note: Staples can cause problems with automated mail sorting equipment.
Please use pressure sensitive or other gummed tape to seal this form.

Cut or Fold Along Line

Environmental Record Editing and Printing Program (EREP)

GC28-1378-2 S370-37

Reader's Comment Form

Cut or Fold Along Line

Fold and tape Please Do Not Staple Fold and tape

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 40 ARMONK, N.Y.

POSTAGE WILL BE PAID BY ADDRESSEE

International Business Machines Corporation
Department A08, Building 035-3
740 New Circle Road NE
Lexington, Kentucky 40511

Fold and tape Please Do Not Staple Fold and tape

IBM®

Printed in U.S.A.

Environmental Record
Editing and Printing Program
(EREP)

READER'S
COMMENT
FORM

GC28-1378-2

This manual is part of a library that serves as a reference source for systems analysts, programmers, and operators of IBM systems. You may use this form to communicate your comments about this publication, its organization, or subject matter, with the understanding that IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you.

Note: ***Copies of IBM publications are not stocked at the location to which this form is addressed. Please direct any requests for copies of publications, or for assistance in using your IBM system, to your IBM representative or to the IBM branch office serving your locality.***

Possible topics for comment are:

Clarity Accuracy Completeness Organization Coding Retrieval Legibility

If you wish a reply, give your name, company, mailing address, and date:

What is your occupation? ______

How do you use this publication? ______

Number of latest Newsletter associated with this publication: ______

Thank you for your cooperation. No postage stamp necessary if mailed in the U.S.A. (Elsewhere, an IBM office or representative will be happy to forward your comments or you may mail directly to the address in the Edition Notice on the back of the title page.)

Note: Staples can cause problems with automated mail sorting equipment.
Please use pressure sensitive or other gummed tape to seal this form.

Cut or Fold Along Line

Environmental Record Editing and Printing Program (EREP)

GC28-1378-2 S370-37

Reader's Comment Form

Cut or Fold Along Line

Fold and tape Please Do Not Staple Fold and tape

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 40 ARMONK, N.Y.

POSTAGE WILL BE PAID BY ADDRESSEE

International Business Machines Corporation
Department A08, Building 035-3
740 New Circle Road NE
Lexington, Kentucky 40511

Fold and tape Please Do Not Staple Fold and tape

IBM®

Printed in U.S.A.

Publication Number
GC28-1378-2

File Number
S370-37

Program Number
5654-260
5656-260
5658-260

Printed in
USA

IBM ®

GC28-1378-02