File No. S370-37
Order No. GC20-1809-7

Systems

IBM Virtual Machine Facility/370: OLTSEP and Error Recording Guide

Release 6 PLC 1

This publication is intended for the IBM customer engineer (CE). The information in this book aids the CE in performing hardware I/O maintenance from a virtual machine. It includes data on error handling, the error recording process, describes how to run the online test system (OLTS) under OLTSEP and how to use the CMS CPEREP command that interfaces with OS/VS EREP (IFCEREP1) and error recorded data.

PREREQUISITE PUBLICATION

IBM Virtual Machine Facility/370: Introduction,
Order No. GC20-1800

COREQUISITE PUBLICATIONS

OS/VS, DOS/VSE, VM/370 Environmental Recording, Editing and Printing (EREP) Program,
Order No. GC28-0772
OS/VS, DOS/VSE, VM/370 Environmental Recording Editing and Printing (EREP) Program Logic,
Order No. SY28-0773
OS/VS, DOS/VSE, VM/370 EREP Messages,
Order No. GC38-1045
IBM Virtual Machine Facility/370:

CP Command Reference for General Users,
Order No. GC20-1820

System Messages, Order No. GC20-1808

Terminal User's Guide, Order No. GC20-1810

IBM®

Eighth Edition (March 1979)

This is a major revision of, and obsoletes, GC20-1809-6 and Technical Newsletters GN25-0417, and GN25-0476. This edition applies to *Release 6 PLC 1* (Program Level Change) of IBM Virtual Machine Facility/370 and all subsequent releases until otherwise indicated in new editions or Technical Newsletters.

Technical changes and additions to text and illustrations are indicated by a vertical bar to the left of the change.

Changes are periodically made to the informaticn herein; before using this publication in connection with the operation of IBM systems, consult the latest *IBM System/370 Bibliography*, Order No. GC20-0001, for the editions that are applicable and current.

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

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, VM/370 Publications, Dept. D58, Bldg. 706-2, P.O. Box 390, Poughkeepsie, New York 12602. IBM may use cr 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.

Preface

This publication is intended for the IBM customer engineer (CE), and assumes that the CE is familiar with OLTS testing procedures. This publication also assumes that the CE is knowledgeable about VM/370 and virtual machine concepts as outlined in the VM/370 Introduction. The CE must also be familiar with the VM/370 logon process as described in the VM/370 Terminal User's Guide.

This publication is divided into four sections.

Section 1 compares the environments available to the CE for testing and repairing I/O devices. The advantages of using the virtual machine as a tool for fault analysis is also described. A comparison of OLTS (online test system) results from both the real and the virtual system/370 is also discussed.

Section 2 discusses the requirements for testing I/O devices from a virtual machine environment which includes the following:

- The CE virtual machine
- How to log onto a virtual machine
- How to run the online tests
- Samples of test runs

This section provides information to permit the CE to run diagnostic tests in a virtual machine environment from a virtual machine console (terminal).

Section 3 describes the VM/370 system error recovery, error recording, and system console error messages, and the control blocks used in the error recovery/ recording process.

Section 4 describes VM/370 facilities that allows more detailed information to be obtained for problem analysis and repair. These include:

- CPEREP and OS/VS EREP
- Intensive Recording Mode
- Trace Option
- VMFDUMP

PREREQUISITE PUBLICATIONS

IBM Virtual Machine Facility/370:

Introduction, Order No. GC20-1800

Terminal User's Guide, Order No. GC20-1810

If the IBM 3767 Terminal is used, IBM 3767 Terminal Operator's Guide, Order No. GA18-2000, is also prerequisite.

If the system being serviced makes use of the IPCS (Interactive Problem Control System) component of VM/370, then the VM/370 Interactive Problem Control System (IPCS) User's Guide, Order No. GC20-1823 is also a prerequisite.

COREQUISITE PUBLICATIONS

IBM Virtual Machine Facility/370:

CP Command Reference for General Users, Order No. GC20-1820

System Messages, Order No. GC08-1808

OS/VS, DOS/VSE, VM/370 Envionmental Recording, Editing and Printing (EREP) Program, Order No. GC28-0772

OS/VS, DOS/VSE, VM/370 Environmental Recording, Editing and Printing (EREP) Program Logic, Order No. SY28-0773

OS/VS, DOS/VSE, VM/370 EREP Messages, Order No. GC38-1045

Figure 1, which follows the Preface, shows the relationship of VM/370 publications to one another within the VM/370 Library.

RELATED PUBLICATIONS

The following texts, although not required, will broaden the CE's knowledge of VM/370 and virtual machines.

IBM Virtual Machine Facility/370:

Planning and System Generation Guide, Order No. GC20-1801

CMS User's Guide, Order No. GC20-1819

Operator's Guide, Order No. GC20-1806

Remote Spooling Communications Subsystem User's Guide, Order No. GC20-1816.

IBM 3704 and 3705 Communications Controllers Network Control Program/VS, Program Logic Manual, Order No. SY30-3007.

In this publication, the term "3330 series" is ued in reference to the IBM 3330 Disk Storage, Models 1, 2, and 11 and the IBM 3333 Disk Storage and Control, Model 1 and 11.

In this publication, the term "2741" is applicable and equivalent to the IBM 3767 Terminal unless otherwise specified.

The term "3270" is used in this publication to refer to a series of display devices, namely, the IBM 3275, 3276, 3277 and 3278 display stations. A specific device type is used only when a distinction is required between the device types.

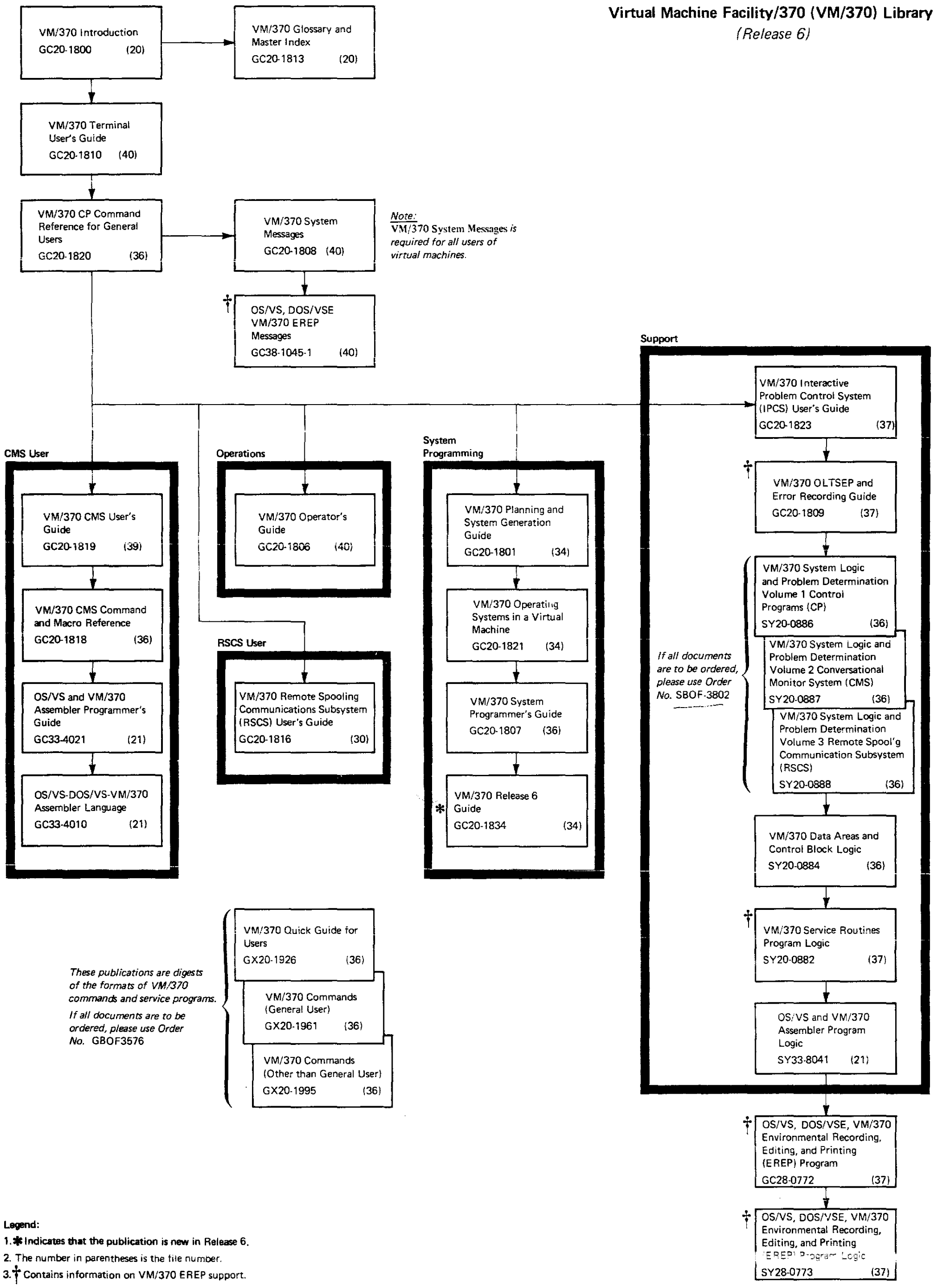
Information about display terminal usage also applies to the IBM 3138, 3148 and the 3158 display consoles, when used in display mode unless otherwise noted.

Any information pertaining to the IBM 3284 or 3286 printer also pertains to the IBM 3287, 3288 and 3289 printers unless otherwise noted.

Notes:

1. External interrupt reflection may cause OLTSEP Release 4.0, 4.1, or 5.0 execution problems; refer to the topic: "Invoking OLTS" for circumvention.

2. VM/370 provides limited 3704/3705 RAS support. Although VM/370 has enough function to effectively utilize the 3704 and 3705, provisions are not available with Release 3 to use the OLTTEP/OLLT/OLTT diagnostic package. If these test facilities are to be invoked then they must be used with VS with TCAM in a standalone System/370.


Virtual Machine Facility/370 (VM/370) Library
(Release 6)
VM/370 Introduction GC20-1800 (20)
VM/370 Glossary and Master Index GC20-1813 (20)
VM/370 Terminal User's Guide GC20-1810 (40)
VM/370 CP Command Reference for General Users GC20-1820 (36)
VM/370 System Messages GC20-1808 (40)
Note: VM/370 System Messages is required for all users of virtual machines.
† OS/VS, DOS/VSE VM/370 EREP Messages GC38-1045-1 (40)
Support
VM/370 Interactive Problem Control System (IPCS) User's Guide GC20-1823 (37)
† VM/370 OLTSEP and Error Recording Guide GC20-1809 (37)
VM/370 System Logic and Problem Determination Volume 1 Control Programs (CP) SY20-0886 (36)
VM/370 System Logic and Problem Determination Volume 2 Conversational Monitor System (CMS) SY20-0887 (36)
VM/370 System Logic and Problem Determination Volume 3 Remote Spool'g Communication Subsystem (RSCS) SY20-0888 (36)
If all documents are to be ordered, please use Order No. SBOF-3802
VM/370 Data Areas and Control Block Logic SY20-0884 (36)
† VM/370 Service Routines Program Logic SY20-0882 (37)
OS/VS and VM/370 Assembler Program Logic SY33-8041 (21)
† OS/VS, DOS/VSE, VM/370 Environmental Recording, Editing, and Printing (EREP) Program GC28-0772 (37)
† OS/VS, DOS/VSE, VM/370 Environmental Recording, Editing, and Printing (EREP) Program Logic SY28-0773 (37)
CMS User
VM/370 CMS User's Guide GC20-1819 (39)
VM/370 CMS Command and Macro Reference GC20-1818 (36)
OS/VS and VM/370 Assembler Programmer's Guide GC33-4021 (21)
OS/VS-DOS/VS-VM/370 Assembler Language GC33-4010 (21)
Operations
VM/370 Operator's Guide GC20-1806 (40)
RSCS User
VM/370 Remote Spooling Communications Subsystem (RSCS) User's Guide GC20-1816 (30)
System Programming
VM/370 Planning and System Generation Guide GC20-1801 (34)
VM/370 Operating Systems in a Virtual Machine GC20-1821 (34)
VM/370 System Programmer's Guide GC20-1807 (36)
✱ VM/370 Release 6 Guide GC20-1834 (34)
VM/370 Quick Guide for Users GX20-1926 (36)
VM/370 Commands (General User) GX20-1961 (36)
VM/370 Commands (Other than General User) GX20-1995 (36)
These publications are digests of the formats of VM/370 commands and service programs.
If all documents are to be ordered, please use Order No. GBOF3576
Legend:
1. ✱ Indicates that the publication is new in Release 6.
2. The number in parentheses is the file number.
3. † Contains information on VM/370 EREP support.

| Figure 1. Virtual Machine Facility/370 (VM/370) Library Release 6

Contents

SECTION 1. HARDWARE MAINTENANCE--REAL MACHINE SYSTEM/370 VS VIRTUAL MACHINE SYSTEM/370. 1
The Ideal Repair Environment--Total Resources of a System/370 and Time for Problem Analysis. 1
Queued Diagnostic System Task--Another Method for Fault Analysis 2
The Virtual Machine--An Alternative Method for System I/O Fault Analysis. . . .2
Online Diagnostics From A Virtual Environment--Test Results 4
Points for the CE to Consider about Virtual Machine Use 5

SECTION 2. VM/370 MAINTENANCE ESSENTIALS.9

TESTING FROM A VIRTUAL MACHINE 11

SYSTEM OPERATOR/CE RELATIONSHIP. 13

THE CE's VIRTUAL MACHINE 14

COMMAND PRIVILEGE CLASS FOR THE CE . . . 16

CONSOLE TERMINAL COMMUNICATIONS CONSIDERATIONS. 18

CONDITIONS FOR INVOKING TESTS. 21
Processor Reliability. 21
Hookup to the Test and Diagnostic Residence Device. 21
Successful Logon 21
Line and Terminal Facility Check . . . 22
Using the LOGON Command. 23

INVOKING OLTS. 24
Invoking OLTS to Virtual Machine Console Terminals 26

OLTS-FRIEND. 28

OLTSEP-RETAIN. 30

BASIC TERMINAL CHECK VIA THE MESSAGE COMMAND 32

BASIC TERMINAL CHECK VIA ECHO COMMAND. . 34

SECTION 3. ERROR HANDLING. 35

OVERVIEW OF I/O ERROR HANDLING 37
I/O Error Recording and SVC 76 38
Error Recording--VM/370 versus an Operating System in a Dedicated Environment 39
SVC 76 40

RECORD MODIFICATION FOR VM/370 ERROR RECORDING 42

I/O Error Recovery--Detailed Description 44

I/O ERROR RECORDING AND ERROR RECORDING AREA. 48
I/O Statistical Data Recording (SDR) . . 49
Permanent I/O Error Recording. 50
Environmental Data Recording 50
Intensive Mode Recordings. 52
VM/370 I/O Error Recordings. 53
Error Recording Record Layout. 53
VM/370 Recovery Features--Introduction . 60
Machine Check Handler. 60
Channel Check Handler (CCH). 60
Handling of Hard Machine Checks. 61
Handling of Soft Machine Checks. 62
Error Recovery Procedures. 63
Recording Facilities 64
VM/370 Repair Facilities 64
VM/370 Restart Facilities. 64

HARDWARE ERRORS AND RECOVERY MANAGEMENT SUPPORT 66
Machine Check Handler--An Overview . . . 66
Levels of Error Recovery 67
Machine Check Handler--Summary 68
Recovery Modes 70
Channel Check Handler. 71
Channel Check Handler--Initialization. 72
Channel Check Handler--Summary 73
Fixed Storage Assignment and Logout Areas 74

SECTION 4. ADDITIONAL CE AIDS. 77

VM/370 CPEREP AND OS/VS EREP 79
CPEREP and OS/VS EREP--An Overview . . . 80
Differences/Exceptions in VM/370 Support of OS/VS EREP 82
Using CPEREP and the Facilities of OS/VS EREP. 82
Entering CPEREP and EREP Operands. . . . 91
Prompting Method 91
File Entry Method. 92
Mixed Method 92
Logon to CPEREP Execution--An Example. 92
CPEREP FILEDEFs. 94
CPEREP Applications. 96

SET RECORD AND SET MODE FACILITY 98
SET RECORD Facility. 98
SET MODE Facility. 99

TRACE FACILITIES 101
CP's TRACE Command 101
NETWORK TRACE. 103
RSCS Logging 103

VMFDUMP. 106

INDEX. 115

FIGURES

Figure 1. Virtual Machine Facility/370 (VM/370) Library............. v

Figure 2. VM/370's Terminal 1051 or 1052 Determination Procedure....................19

Figure 3. Determining the Line Transmission Code for the 2741........................19

Figure 4. Code Comparison Using the LOGON Command.................20

Figure 5. 2741 Printout--A Typical CE Terminal Session Using OLTSEP-OLTS...................25

Figure 6. 2741 Printout--A CE Terminal Session Invoking OLTSEP-FRIEND........................28

Figure 7. 2741 Printout--A Terminal Session Showing Use of RETAIN/370...................31

Figure 8. 2741 Printout--A CE Terminal Session Invoking the ECHO Command.............34

Figure 9. I/O Operation Control Block Linkage......................44

Figure 10. I/O Error Recovery Control Block Structure for Sense-Byte Analysis...........45

Figure 11. Control Block Linkage for Retry.........................46

Figure 12. Control Block Linkage SDR Counter Update...........47

Figure 13. Control Block Linkage--Fatal Error Condition........50

Figure 14. 2305 Control Block Structure --Environmental Data Recording.....................51

Figure 15. 3330/3340/3350 Control Block Structure--Environmental Data Recording...............52

Figure 16. Unit Check Record Layout (Long).......................55

Figure 17. Nonstandard Record Layout....55

Figure 18. Unit Check Record Layout (Short)......................55

Figure 19. Machine Check Record Layout..56

Figure 20. Channel Check Record Layout..56

Figure 21. Record Breakdown Table (Except Header)..............57

Figure 22. Layout of the Header Portion of the Error Records (24 Bytes)....................58

Figure 23. Header Record Table...........59

Figure 24. Summary of RMS Functions......66

Figure 25. Condition/Action Table for Uncorrectable Errors.........69

Figure 26. Handling of Channel Data Check, Channel Control Check, and Interface Control Check..72

Figure 27. Handling of Interface Inoperative..................72

Figure 28. Channel Check Action Table...74

Figure 29. VM/370 Fixed Storage and Logout Areas..................75

Figure 30. VM/370 vs OS/VS1 and OS/VS2 Error Record Types Recorded......................81

Figure 31. CPEREP Command Format........83

Figure 32. Operands Used with CPEREP....85

Figure 33. Types of Reports, Showing Operands Allowed with Each...90

Figure 34. Segment of a Trace Printout of a Program's I/O Operation..................103

Figure 35. Read and Write Log Records for SML.....................105

Figure 36. Formatted VMFDUMP...........108

4331, 4341 PROCESSORS WITH 3278 MODEL 2A CONSOLE AND 3031 AP SUPPORT

New: Processor Support

VM/370 now supports the 4331 and 4341 processors as well as the 3031 attached processor. The 3278 Model 2A console can be used as a virtual machine console terminal supported by VM/370 for customer engineer use in conducting diagnostic tests. When running channel check handler, limited channel logout is still available for the 4331, 4341 and 3031 AP processors. However, there is no fixed or I/O extended logout area for these new processors. Errors corrected by error checking and recording (ECC) are not recorded by the new processors. Only errors corrected through processor retry are recorded.

3203 MODEL 5 PRINTER SUPPORT

New: Hardware Support

VM/370 now supports the 3203 Model 5 printer for use in hardcopying errors encountered during diagnostic testing of the system.

MISCELLANEOUS

Changed: Documentation only

- Figure 30 has been amended to include further documentation of error record types recorded by DOS, DOS/VS, OS/VS1, OS/VS2, and VM/370.
- Correction of the default for the ACC= operand of CPEREP command in Figure 31 from NO to YES has been made.
- An expanded description of the function of the CLEARF operand of CPEREP command has been added to Figure 32.
- The term "error recording cylinder(s)" has been changed to "error recording area(s)" where applicable in the text.
- Mincr editorial changes have been made.

Summary of Amendments
for GC20-1809-6
as updated by GN25-0476
VM/370 Release 5 PLC 12

ENSURING VM/370 CONTROL PROGRAM HAS ACCESS TO SRF DEVICE

Changed: Documentation only

Added to the discussion of the SRF device as it relates to VM/370 is the means of ensuring that the VM/370 control program has access to the SRF. Also documented are the steps necessary to activate the SRF device.

VM/370 SUPPORTS THE 3031, 3032 AND 3033 PROCESSORS

New: Processor Support

VM/370 supports the 3031, 3032 and 3033 processors as processors capable of running VM/370.

Support for these processors includes support for the new integrated channels. These channels store a limited channel logout, as well as an I/O extended logout of 640 bytes. The channels conform to standard System/370 channel interface protocol for normal operation, but the interface has been expanded to allow for signaling I/O interface inoperative and group inoperative conditions detected by the hardware. Such conditions, when detected, result in the termination of VM/370.

CPEREP has been expanded to handle error records produced by these processors in conjunction with the SRF (Service Record File) 7443 device. The SRF device contains SRF frames that are used to format the records on the error recording area. This necessitates the following changes:

- A new CPEREP operand, CLEARF, clears error records and SRF frame records from the error recording area, then writes new SRF frame records to the area. The CLEAR operand clears only error data, not frame data.

- The CLEARMC, CLEARIO, and CLEARALL operands are deleted. There is no longer one error area for I/O and another for machine checks and channel errors.

From two to nine error recording cylinders may be defined at system generation. Errors are recorded in order of occurrence until allotted cylinder space is exhausted.

This new support is described in the following sections:

Section 2. VM/370 Maintenance Essentials
Section 3. Error Handling
Section 4. Additional CE Aids

VM/370 SUPPORTS CHANNEL CHECK REFLECTION

Changed: Program and Documentation

VM/370 now reflects channel control checks, channel data checks, and interface control checks on user-initiated I/O events (excluding diagnose-initiated events, where the recovery process is handled by CP) to the user so that the user can attempt recovery. For diagnose-initiated events, the results are reflected to the user's virtual machine after the retry. If the user's SCP fails in its recovery attempt, the user may terminate operations or the task affected by the channel check.

This new support is described in the following sections:

Section 1. Hardware Maintenance -- Real Machine System/370 vs. Virtual Machine System/370
Section 3. Error Handling

VM/370 SUPPORTS ONLY IPCS VMFDUMP

Changed: Program and Documentation

VM/370 supports only the IPCS version of VMFDUMP. The DMKEDM module is no longer contained in the system.

"Section 4. Additional CE Aids" has been updated to reflect this change.

MISCELLANEOUS

Changed: Programming and Documentation

Minor technical and editorial changes have been made to clarify the text.

Section 1. Hardware Maintenance-Real Machine System/370 VS Virtual Machine System/370

Most system hardware failures are caused by storage and I/O device errors. Most of the errors, once sense data and other information is analyzed, can be repaired offline (physically and/or logically disconnected from the rest of the system). However, there are instances where offline test equipment is not adequate to simulate the fault condition as it occurred on the system; therefcre, the system must be used to effect the repair. Similarly, the system must be used in a final diagnostic checkout of a device after it has been repaired offline and prior to returning the device to the customer fcr operating system usage. Another consideration for system use is to check the reliability of a device following an EC (engineering change). The customer may also use basic diagnostics that utilize the system as an aid in the initial analysis of whether a system fault is hardware or software in origin.

The previously described uses of a System/370 as a tool for the repair and checkout of I/O devices is addressed in more detail by the following discussion. Cost factors are not a consideration in the analysis.

The Ideal Repair Environment -- Total Resources of a System/370 and Time for Problem Analysis

Testing and troubleshooting device/storage malfunctions cr suspected device malfunctions, when established local and offline troubleshooting techniques fail, is best achieved from an environment that is totally and exclusively under the absolute control of the CE. Total control cf a system and its resources excludes the use cf the system by other users, their data sets, their programs, and their hardware system requirements. This exclusive test mode allows the CE to use the total resources and power of the system in conjunction with diagnostic aids to track and isolate the system fault.

There are two reasons why such an environment is ideal for the isolation of device faults. First, there is no contention by other programs for the data paths and control paths to and from the device. Second, any approach to troubleshooting, no matter how bizarre cr radical, can be undertaken because there is nc risk of destroying the customer's programs and data sets. However, this ideal method cf problem analysis is not without its shortcomings. Field engineering personnel CEs are only granted the total resources of a system when:

- The malfunction to the system or to a system resource cannot be analyzed or repaired by offline equipment
- The malfunction is so catastrophic that the entire system can be classified as unavailable to the customer
- System preventive maintenance is to be done. Unfortunately, on large systems, this endeavor is usually scheduled cn weekends or cn other than prime shift hours.

Outside of preventive maintenance work, loss of the system to the customer for productive work is traumatic. The CE is placed under great personal stress to diagnose and repair the system and get it back in operation as soon as possible.

Queued Diagnostic System Task -- Another Method for Fault Analysis

As a compromise to the totally dedicated use of a System/370 repairing or checking out the hardware after a repair is made, it is possible with some online systems to place the CE diagnostic program on a task queue. At times, the diagnostic task is at the top of the queue and ready to be used to exercise and test selected hardware.

The disadvantages and difficulties associated with this method of device repair or checkout are as follows:

- Possible contention on data and control signal paths to or from the device

- Complexity of problem analysis imposed by the programming levels and the queued task diagnostics

- Constraints of time and priorities imposed by the system operator

- Limited flexibility in the diagnostic approach to a given problem

Expanding on the possibility of data path and control path contention, suppose the CE is monitoring control signals or a teleprocessing line. Is the data represented on the scope or monitor device related to the diagnostic test or exercise or is it related to an "automatic" polling sequence or another control program task? If the data being monitored is related to another system function as well as the CE diagnostic activity, the problem of fault isolation becomes more complex and time consuming. In addition, the diagnostic test sections are controlled by a "driver" program (for example: CLTEP) which, in turn, is controlled by the operating system. This tier of programming overhead imposes an added level of understanding on the part of the CE who must repair the malfunctioning unit.

This method of repair also requires the help of the system operator who must allocate the time and the resources needed to make the repair. Quite frequently, the CE's request for system time to run diagnostic tests is given a relatively low run priority in relation to customer tasks; this is particularly true if the device to be serviced by diagnostic programs fulfills no immediate need for the customer. In such situations, the CE has no alternative but to wait for the system to be relinquished to him.

Another problem with this method of problem analysis is that (with the available diagnostic test sections and options) there are limits to the test patterns and loop conditions that can be used to exercise the failing unit. Generally, means are not provided for dynamically changing storage or register values to build more stringent and exhaustive tests tailored to the CE's own test criteria.

The Virtual Machine -- An Alternative Method for System I/O Fault Analysis

The virtual machine is a counterpart of a real System/370. It is generally available for use for the CE whenever the CE has a need to use the system. The CE can immediately use the system and diagnostic test sections to check out or locate I/O faults on an I/O device after he has completed the virtual machine logon process (as described in the VM/370 Terminal User's Guide) and solicited a minimum amount of assistance of the system operator in attaching the failing device to the CE's virtual machine.

The CE's virtual machine, or virtual system is part of the real system but only a time slice utilization of it. Low storage as well as system registers and processor functions of the virtual machine are simulated by the control program (CP) component of VM/370. Protective features of the VM/370 System Control Program isolate and protect the action, programs and data sets of one virtual machine from interfering with the action, programs and data sets of other virtual machines. Thus, operations of the CEs virtual machine have negligible effect on other System/370 operations. As an alternative to having the power of a real System/370 at a CE disposal, the virtual machine can provide similar functions with some sacrifice to performance. However, with the use of the virtual machine, there are certain timing dependencies, device applications and processes that are not supported by VM/370; these are detailed in the VM/370 System Messages under the title, "VM/370 Restrictions."

The facilities provided by VM/370 and the virtual machines it supports are briefly described in the VM/370 Introduction. The virtual machine has almost the full range and capabilities of a real System/370. That is, it has registers and storage comparable to a real System/370. It has unit record devices (virtual unit record devices) called spooling devices that programs or data sets can utilize for punched or printed output. A virtual card reader is available to read data or programs into the virtual machine for processing. In addition, a virtual machine can be expanded or contracted by the use of commands that attach or detach devices/resources for the exclusive use of the virtual machine operator. The means of controlling the virtual machine and these devices is through a terminal that serves as a system console for the virtual machine. By keying in what are termed "console function" commands on the terminal, simulation of many of the functions that are performed by buttons and switches on a real System/370 control panel can be accomplished.

Some of the functions that can be simulated for the virtual machine by use of commands follow:

ATTN,REQUEST	Attention interrupt from a system console
ADSTOP	Address stop facility
DISPLAY	Display storage and display register capabilities of a system console
EXTERNAL	External interrupt key on the system console
IPL	Console LOAD key
NOTREADY	Loss of READY to a virtual device
READY	READY state of a virtual device
REWIND	Function of the Tape Drive Rewind Key
STORE	Function provided by the store key on the system console

In addition to the commands that have a direct relationship to function provided by the System/370 control panel and console, there are other commands available to the user or the system operator that can benefit the CE in his role as troubleshooter; these commands and a brief explanation of their uses are shown in an appendix in the *VM/370 CP Command Reference for General Users*.

Commands that are available to the general user (and likewise, the CE) are described in the *VM/370 CP Command Reference Guide for General Users*. The format and use of commands that pertain to all other users of virtual machines including the privilege class F user (that is, commands designed for the CE engaged in hardware maintenance) are contained in the *VM/370 Operator's Guide*. Section 4 of this book contains more detailed information about the privilege class F commands.

Online Diagnostics From a Virtual Environment -- Test Results

The CE must have confidence in the virtual machine as a tool for device checkout and hardware debugging. But how does the virtual machine environment compare with a real System/370 environment when both use the same OLTS sections as the diagnostics for testing identical devices? The answer: very favorably.

Tabulated results were compiled from OLTSEP OLTS test runs. Tests were initiated from a dedicated System/370 Model 145 (standalone) environment and also from VM/370's multiuser virtual machine environment. Concurrent testing was accomplished by the CE using OLTSEP and OLTS via the assigned CE's virtual machine.

The tabulated results of OLTS indicates that only 7.5 percent of the 140 sections tested resulted in errors that were unique to virtual machine operations. These errors were a reflection of those OLT sections that violated VM/370 architecture (see the publication *VM/370 System Messages* under the topic "VM/370 Restrictions"), such as, dynamically-modified channel programs and time dependent routines.

The tabulated OLTSEP/OLTS results also indicate failures that were generated in the standalone (dedicated System/370 environment) as well as in the virtual machine environment. Those errors that are common to both the real and the virtual system were caused by one of the following:

- OLTS section fault (program)

- Hardware malfunction

- Hardware and OLTS were not at a compatible EC (engineering change) level

- Incorrect program options selected for the devices involved

- Incorrect hardware strapping, plugging, or switch selection

No attempt was made to diagnose the specific reason for all of the indicated failures. What is significant is that *all* the failures that occurred on the standalone system also occurred in the virtual system. No error detected by the dedicated system operation escaped detection during a subsequent run of the same OLT sections from a virtual machine.

The tabulated results were also indicative cf the fluid nature cf computing systems; neither the hardware nor the programs remain in a dormant state for any length of time. Either the system configuration changed, program test sections were updated, or the system hardware had been modified by EC and RPQ changes. For the CE, maintaining up-to-date diagnostics that reflect the current system configuration is not without its problems. To help circumvent these problems, it would be wise to create and maintain a history file for OLTS printouts that reflects both virtual machine and standalone operations. This file would receive copies of OLTS results run in both a virtual machine and standalone system after ensuring that all

- System and/or device installation site modifications have been made.
- Sales or engineering changes to system hardware have been installed.
- Modifications and updates to the OLTS sections are complete.

If properly maintained, an OLTS history file can prevent unnecessary and time-consuming problem analysis for conditions that only reflect an incompatibility of program and hardware.

The test results were obtained from a System/370 Model 145 and the following typical hardware mix:

Machine	Model	
1403	N1	HS Printer
2305	2	
2318	1	
2319	A0	
2400	5	
2540	1	
2703	1	
2821	1	CTRL Unit for Card/Printer Equipment
2835	2	TCS Tape Drive Control Unit
2803	2	
3145	1	
3215	1	
3330	1	
3803	1	
3830	1	Disk Control Unit

Bear in mind that the test results did not shcw every OLT section run nor did they indicate every device supported by VM/370; rather, the test results indicated that with a good hardware mix, there was a typical error fallout. Conceivably, tests run on other System/370 VM/370 systems would reflect similar but different inconsistencies between OLTS and the hardware and options involved.

Note: OLTS tabulations as a result of RETAIN and the 2955 interface are identical tc the results obtained by the site CE invoking the tests (see "OLTSEP--RETAIN").

None of the tests executed in the VM/370 virtual machine environment resulted in a hang, reset, or loop condition of the virtual machine, ncr was there any perceivable effect on the operations and security cf VM/370 and cther associated virtual machines.

Points for the CE to Consider about Virtual Machine Use

As stated previously, the *VM/370 Introduction* will acquaint the CE with the power and versatility of the virtual machine. A more in-depth study of virtual machine use (with other operating systems operating in the virtual machine environment) is found in the *VM/370 Operating Systems in a Virtual Machine*.

With the CE's use of the virtual machine, the following considerations should be made:

- To provide all of the functions and tests described in this publication, the CE needs a directory entry with a privilege class F and G.

- The list of VM/370 restrictions as documented in VM/370 System Messages should be consulted to see whether or not the malfunction or suspected malfunction is a violation of VM/370 architecture. Certain OLTS diagnostics violate VM/370 rules; particularly those tests that have time dependencies or dynamically modify channel programs.

- Loaded diagnostics programs and related test sections reside at their virtual address. The virtual address is not the same as the real storage address unless the V=R special performance option is invoked.

- Parts or all of the CE's diagnostic programs may be paged out from processor storage to auxiliary storage because of concurrent use by the other virtual machines. The system operator can, if the situation warrants, lock virtual machine page(s) in processor storage.

- An I/O device address may be a virtual address. The virtual address may be represented by its full size real counterpart, such as a tape drive or because it can be a logically subdivided portion of a disk drive (such as a minidisk).

- All system errors and I/O errors are not written out to the VM/370 error recording area; consult "Section 2. Error Handling" for details. If SVC 76 is used by virtual machines to effect error recording, then the virtual machines must meet specific parameter passing criteria. Also, VM/370 itself does not generate EOD and IPL records. No error recordings of these types are accepted for the VM/370 system as well as other virtual systems. Certain other error types are also not processed.

- Most CCWs and CCW chains are subject to VM/370 control program modifications in order for VM/370 to maintain its overall paging environment correctly.

- Because of the time slice technique used in dispatching virtual machines by the control program, the run time for diagnostic test sections is longer. It may be considerably longer if there is heavy concurrent System/370 use by other virtual machine users.

- The system operator has control of certain special virtual machine options and other VM/370 options that can, if the situation warrants, be invoked to aid the CE and his virtual machine in problem analysis. Brief descriptions of these options are contained in the VM/370 Introduction.

- The facilities of the CMS EDIT command can be used to modify or create short diagnostic loops or tests for problem analysis. For details on this command, consult the VM/370 CMS Command and Macro Reference, and VM/370 CMS User's Guide.

- Analysis of system and I/O problems can be accomplished by the CE from a remote isolated (virtual machine) terminal provided the area of the CE's terminal is serviced by an RSCS (Remote Spooling Communications Subsystem) workstation. By using the workstation for the spooled output of the results of the diagnostic tests invoked from the terminal, the CE can make a preliminary but through analysis of a machine's malfunction.

- In attempts to service components of a 3850 Mass Storage System, the CE should be aware that the virtual machine is interfacing with virtual 3330 volumes (3330V) and not with a real 3330 device; thus, the misapplication of diagnostics could lead to erroneous interpretations.

- In testing components of the 3850 Mass Storage System (MSS), most functions provided by the online test system (OLTS) require that MSS activity be quiesced. To insure a quiesced mass storage system, it is recommended that the CE's virtual machine be run in a standalone environment.

- The CPUID found in the error recording records is the CPUID associated with the real machine and not the one associated with a virtual processor.

- If the facilities of an IBM 3850 Mass Storage System (MSS) are used with VM/370 virtual machine operations and MSS errors are reflected to VM/370's error recording area, CPEREP must be invoked so that MSS-related errors recorded in the error area can be directed to an accumulation (ACC=Y) tape for further processing by the VS1/VS2 Subsystem Data Analyzer (SDA) Program. Because MSS logged out data is voluminous and the interrelationship of MSS components is complex, it is imperative that this service program be used to effectively diagnose and isolate mass storage problems.

- The virtual machine used by the CE normally does not have a dedicated high speed printer. Therefore, long listings (such as console spooling records, dumps, error recording records, and diagnostic output tabulations) are queued to a common spool output device along with the files generated by users of other virtual machines. These files are queued by class as well as by the time at which the files are closed. If the queue for output is long or contains files that are sequentially ahead of the CE's output records, the wait for output could be quite lengthy. However, the system operator can alter the order (sequence) of output files, if the need is urgent.

- The I/O configuration of the virtual machine should be such that each virtual channel maps to real channels of a single type and model. This requirement is explained in detail in "Appendix E. VM/370 Restrictions" in _VM/370 System Messages_. If this requirement is not met, the STIDC instruction may return inconsistent results, and any data from a channel extended logout may be misinterpreted since it depends on the channel model. Also note that there is a restriction against using control register 14 to mask out channel extended logouts; if this is done in a virtual machine, the logout does not remain pending and instead is lost.

Section 2. VM/370 Maintenance Essentials

- Testing from a Virtual Machine
- System Operator/CE Relationship
- The CE's Virtual Machine
- Command Privilege Class for the CE
- Console Terminal Communication Considerations
- Conditions for Invoking Tests
- Input Line Editing
- The Terminal Session
- Invoking OLTS
- OLTS-FRIEND
- OLTSEP-RETAIN
- Basic Terminal Check via the MESSAGE Command
- Basic Terminal Check via the ECHO Command

VM/370 is a system control program (SCP) that can be used on IBM System/370 computing systems equipped with the dynamic address translation (DAT) and the system timing features (STF).

The Online Test Standalone Executive Program (OLTSEP) and associated online test system (OLTS) are not part of the VM/370 system. OLTSEP and OLTS are ordered for the particular computer site and its related equipment by the customer engineer (CE) for use in diagnostic servicing.

Maintaining and upgrading OLTSEP and OLTS test sections and the transfer of this data to different storage media are not a responsibility of the VM/370 system. Existing documentation associated with OLTSEP and OLTS describes these procedures.

Testing From A Virtual Machine

The following conditions must be satisfied to permit testing from a virtual machine environment.

1. The integrity of the complete computer system cannot be degraded to the point where the VM/370 program cannot be run.
2. I/O and channel logic communication paths are operative as applied to OLTSEP and OLTS.
3. The virtual machine assigned to the CE is available and functicning.
4. The communication path from and to the CE's terminal is functioning and in an enabled state.
5. The CE user virtual machine identification (userid) and password are kncwn to the CE.
6. The device(s) to be tested must be available for the CE's exclusive use.

Note: If any of the above conditions is not satisfied, the System/370 operations personnel must correct the situation by command entries or a system reconfiguration process if concurrent maintenance is desired. These processes are described in the *VM/37C Planning and System Generation Guide* or the *VM/370 Operator's Guide*.

The hardware maintenance enccmpasses the following major areas of a system complex: the main processing unit (and the attached processor, if applicable) and input/output (I/O devices). Each is maintained in a different way.

- The processor (or attached processor) is maintained in a dedicated environment. There is no method available that allows the concurrent maintenance of the processor including its main storage and channels, while running user jobs under VM/370.
- The I/O equipment, however, can be maintained by using online tests system (OLTS) under OLTSEP in its own virtual machine. It is this relationship that this book addresses.

Data Security

Tapes and files created by CMS and CP do nct ccnform to the OS or DCS labeling techniques, nor do their tape and disk files use the security protection byte found in other control systems. Files and tapes generated by an OS or DOS controlled virtual machine under VM/370 supervision could contain these protection features. Therefore, the CE must proceed cautiously, since tape and disk files encountered on a VM/370 system, as OLTSEP, *may not* restrict the CE from inadvertently destroying customer or system data.

Note: This consideration arises when a disk pack is mounted on the specific device dedicated to the CE's virtual machine via the CP ATTACH command.

System Operator/CE Relationship

Working from a virtual machine, the CE should be aware of the time slicing and device sharing environment of VM/370. The management cf these facilities belongs in part to the system operator. The CE's virtual machine is also part cf the system cperator's domain. The system operator can (if the system is large enough to sustain such action) dedicate devices, control units, and even channels to the CE's virtual system.

The CE should be aware cf the system operatcr's responsibility to other users' virtual machines. The system operator, because of schedules and system workload, may not grant the CE's every request.

The shared system responsibility of the operator and the CE manifests itself in situations where a CE testing from a remote location performs system maintenance. Through mutual cooperation and the MESSAGE and ATTACH commands, a complete I/O diagnostic check can be accomplished.

The CE should also be aware that maintenance operations affect the throughput time of other users' virtual machine operations, and conversely, that other virtual machines' cperaticns affect the throughput time of the CE's diagnostic operations.

The relaticnship between the system operatcr and the CE may enhance I/O maintenance. This can be done by having the system operatcr exercise system options within his control. Suppose, for example, a problem exists and the tag lines are suspect. Oscillcscope trace interpretation can be difficult if many virtual machines shared the same bus and tag paths. To alleviate this problem, the system operator (if it is within his control) can dedicate a complete channel and all its related hardware to the CE's use. Thus, while looping on an OLTS routine the I/O data and control paths would be free cf other user I/O activity. Note also, if the system operator has access to the problem report file system of the Interactive Problem Control System virtual machine, an initial and instant analysis for the current problem by comparison to a base of previously reported custcmer VM/370 problems can help determine whether the malfunction occurred in the hardware or software.

The CE's Virtual Machine

Hardware I/O maintenance can be accomplished by having the CE operate his own virtual machine from a terminal device while permitting other VM/370 users to continue operating their own virtual machines. The CE's virtual machine is unique in that his CP command privilege class F allows him to run CPEREP to set intensive mode recording and invoke the NETWORK TRACE function (both these options are described later).

The virtual machine described below, accessed through the remote terminal device, provides the CE with almost all of the facilities of a dedicated System/370. The CE can store, display, PSW restart, IPL, start, and stop the program of his choice without affecting other users.

In most instances, the CE needs no dedicated computer time for most of the preventive maintenance tests. There is usually little or no problem in being granted additional time for additional test runs if they are needed. The CE can be granted time to create his own subroutines if he so desires. This can be done by using some of the CP console function commands that are fully described in the _VM/370 CP Command Reference for General Users_.

The typical CE virtual machine configuration specified in the user directory may consist of the following:

```
USER CEMAINT PASSWORD 512K 1M GF
CONSOLE      009      1052[1]
SPOOL        00C      2540     READER
SPOOL        00D      2540     PUNCH B
SPOOL        00E      1403     PRINTER A
MDISK        190      2314     000   050  CMS19C R
MDISK        191      2314     010  005  CMS001 W
```

The above configuration is interpreted as follows:

- The first line contains, in left-to-right order, the user identification for the virtual machine, the security password, the normally assigned storage size, the maximum storage size the user can specify, and the assigned CP command privilege classes G and F. The G class is necessary for the CE to examine or change any values in his virtual machine, such as to examine sense bytes or change PSW values. The F class allows the CE to specify intensive recording mode, to invoke a trace facility to a 3704/3705 resource, or to invoke CPEREP to edit and clear error records. There are no other facilities offered with this privilege class.

- The second line identifies the virtual console address and type. This entry does not need to be related to the type of terminal device the CE logs in on.

[1] The terminal used by a CE can be any of those listed under "Console Terminal Communications Considerations." Some of these consoles are display consoles and the input, output, and attention handling techniques differ from document-producing terminals. If usage difficulty is experienced with any terminal, consult the _VM/370 Terminal User's Guide_.

- The third, fourth, and fifth lines represent the virtual unit record spool devices and addresses that are mapped by the system to equivalent real devices. The letter B located on line four and the letter A located on line five represent the assigned spool class for that device.

- The sixth line is an entry describing a 2314 minidisk with an address of 190 that contains the CMS system residence files on cylinders 000 to 050. This disk is labeled CMS190 and this user has read-only (R/O) access to this disk, since it is the CMS system disk.

- The seventh line is interpreted the same as the sixth line with the following exceptions: only five cylinders are allocated to the user of the 2314 volume labeled CMS001. However, the write (W) access privilege allows the CE to write routines or modify existing routines and store them permanently on this disk.

For further details on VM/370 user directory entries, see the *VM/370 Planning and System Generation Guide*.

Command Privilege Class for the CE

The CE's virtual machine is similar to other virtual machines running under VM/370. The CE's virtual machine reacts to the System/370 machine instruction set in much the same manner as on a dedicated System/370. Control of the virtual machine is through a terminal and CP commands. These commands are grouped into eight privilege classes. Each class relates to specific system functions. The privilege class or classes of commands assigned to a particular virtual machine are stored in the VM/370 directory along with the user's virtual machine identification code and password.

As a user of a virtual machine, it is assumed that the CE has the class G and F commands and CMS allocated for his use. CMS is discussed briefly in the VM/370 Introduction. CMS is important to the CE because this environment must be entered to execute the CPEREP command. CPEREP, when invoked, calls EREP modules that format and print error recording data; optionally, CPEREP may be used to create an accumulation tape (ACC=option) or edit an existing accumulation tape (HIST option); even SYS1.LOGREC data sets on tape or disks compiled from other systems may be used. If the CE in a remote location has access to any of the remote terminals supported by the RSCS component of VM/370, he may utilize the facilities of RSCS to transfer bulk data, such as trace output and error recording printouts, to a remote printer. Remote spooling procedures are described in the VM/370 RSCS User's Guide.

The use of CPEREP is also important in relation to its use with the 3850 Mass Storage System. Errors accumulated on the VM/370 error recording area must be placed in the CPEREP accumulator output tape for additional processing and analysis by the VS1/VS2 subsystem data analyzer (SDA) program. For details on how this is accomplished, refer to CPEREP and OS/VS EREP in Section 4 (for a description on how to create an output tape) and then refer to either OS/VS1 SYS1. LOGREC Error Recording, Order No. GC28-0668 or OS/VS2 System ProgrammingLibrary SYS1. LOGREC Error Recording, Order No. GC28-0677 for details.

The class F commands include the SET RECORD and SET MODE commands. With these commands, the CE can set requirements for intensive or soft error recording. Refer to "Section 3. Additional CE Aids." Class F allows the CE to void error recording that occurs as a result of the CE's virtual machine activity except for the device and condition specifically named in the SET RECORD command.

Class F also allows the CE to generate trace data for a specified 3704 or 3705 BTU (basic transmission unit) or resource by means of the NETWORK TRACE command. The produced trace data is then spooled to the CE's virtual printer. Class G commands comprise a complete set of commands for virtual machine use.

In addition to the Class F and G commands, there are commands that are not confined to any assigned command category. These commands, referred to as the class "Any" commands, can be invoked regardless of logon status. Examples of these commands are MESSAGE and LOGON.

This book illustrates the use of only those VM/370 commands necessary for CE applications. However, if additional help is necessary, the CE can solicit help from the system operator via MESSAGE OPERATOR command, or use the VM/370 CP Command Reference for General Users, the VM/370 Operator's Guide, and VM/370 CMS Command and Macro Reference.

Also be aware that, although many commands are discussed in this book, not all operands pertaining to these commands are discussed. Full descriptions of all CP and CMS commands and their operands are contained in the above publications.

Included in the grouping of CP commands are those commands that might be used in applying a diagnostic program against a generated device condition. These commands may be a beneficial troubleshooting aid in a comparison study between virtual device reaction and real device reaction.

CAUTION: Although not specifically discussed in this text, CMS commands exist that can destroy existing CMS files by erasure or by overlaying. Refer to the *VM/370 CMS User's Guide* for a discussion of the CMS file management system.

Console Terminal Communications Considerations

A console terminal is used as a communications device between the user and the processor. Those devices supported by VM/370 can also be used as virtual machine console terminals. Some of those devices, however, need specific hardware features to facilitate VM/370 usage. For a complete list of devices supported by VM/370 and used as console terminals, refer to "Part 1. Planning for System Generation" in _VM/370 Planning and System Generation Guide_.

VM/370 also supports the following IBM transmission control units (TCU), communications controllers, and display control units to process the data to and from the terminal devices.

Transmission Control Units and _Communications Controllers_	_Display Control Unit_
2701	3271 Model 2
2702	3272 Model 2
2703	3274 Model 1B
3704	3274 Model 1C
3705	3276 Models 2, 3 and 4

The _VM/370 Planning and System Generation Guide_ contains a list of the features necessary for each device to operate in the VM/370 environment.

VM/370 supports virtual machine operation through the user's terminal linkage to the system. Each terminal type uses its own communication language, data transmission speed, and communication sequence technique. Therefore, for intelligent and meaningful data transfer between each user and his virtual machine, use of the correct translation tables and command sequences must be established.

EBCDIC is the code used by the hardware logic of all VM/370 supported devices listed in "Part 1. Planning for System Generation", _VM/370 Planning and System Generation Guide_. One exception is the 2741 which uses either PTTC/EBCD or Correspondence code.

The supported terminal devices can be categorized as belonging to either IBM Terminal Control Type 1 or IBM Telegraph Terminal Control Type 2.

For a list of the features and RPQs necessary for VM/370 usage of these terminals and consoles, consult the _VM/370 Planning and System Generation Guide_.

VM/370 system generation defines to the operating system the physical hardware components on that system. This entails matching the hexadecimal hardware address of that device to a device type designation (for example, 2314). This is necessary so that data communication between the processor program and the devices is decipherable and meaningful. This is accomplished by using the correct translation tables for terminals and consoles. In VM/370, this merging of address to device type is done for all devices except 1052s, 2741s, and 3767s. The 1052s and 2741s can reside on any remaining available telecommunication lines. The matching of the device transmission code to a designated line address is a function of the enabling sequence to

Determination of whether the device on the enabled line is a 1052 cr a 2741 is handled by the initial communication sequence between VM/370 and the terminal. This is illustrated in Figure 2. VM/370 handles the 3767 terminal and the 2741 terminal identically.
that device and the deciphering of the LOGON (or DIAL) command.

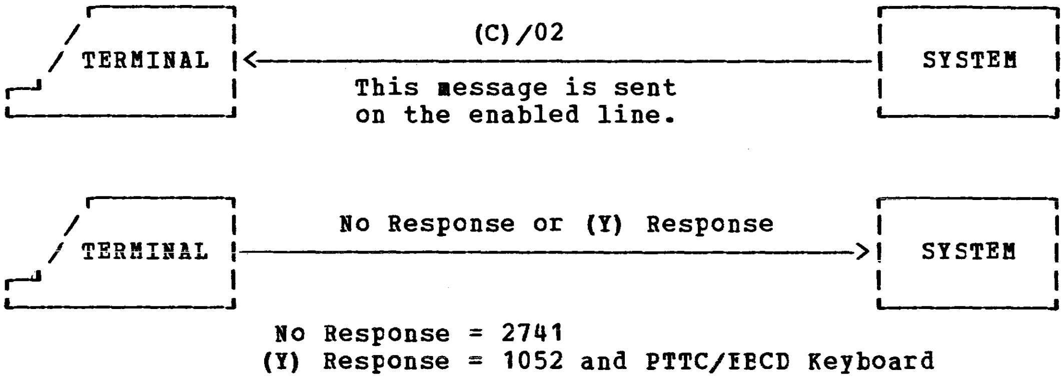


Figure 2. VM/370's Terminal 1051 or 2741 Determination Procedure

The code that the 2741 Terminal uses—PTTC/EBCD or Correspondence—is determined by deciphering a privilege class Any command. For a complete list of these commands, see the *VM/370 CP Command Reference for General Users*. One of these CP commands is the LOGON command shown in Figure 3.

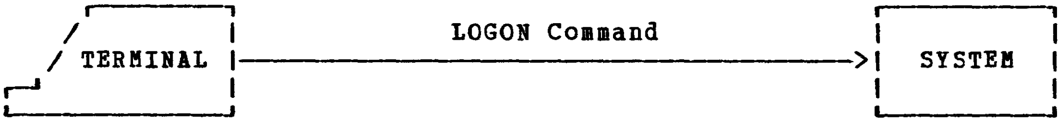


Figure 3. Determining the Line Transmission Code for the 2741

Code determination is done by the program examination of the LOGCN word initiated at the beginning of the terminal session. Deciphering LOGON or any valid contraction of LOGON followed by a blank character to one of the two codes, establishes the code to the applicable terminal.

Figure 4 indicates the differences of the two codes involved with LOGON.

	Start	1	2	3	4	5	6	7	Stop
L =	X	X	X	X	–	–	–	–	X
O =	X	X	–	X	–	–	–	–	X
G =	X	X	X	–	–	–	X	–	X
O =	X	X	–	X	–	–	–	–	X
N =	X	–	X	–	–	X	–	–	X

Correspondence Code

	Start	1	2	3	4	5	6	7	Stop
L =	X	X	–	–	–	X	X	–	X
O =	X	X	–	–	X	X	–	–	X
G =	X	X	X	–	X	X	X	–	X
O =	X	X	–	–	X	X	–	–	X
N =	X	X	–	–	X	–	X	–	X

PTTC/EBCD

Figure 4. Code Comparison Using the LOGON Command

The merge of device type, transmission code, and line address is indicated in the RDEVBLOK (Real Device Block) applicable to that virtual machine.

The RDEVBLOK is defined as storage area that contains a specific number of doublewords that describe the characteristics, and other data applicable to a designated device. (These blocks of data are shown in a formatted dump output. See VMFDUMP in Section 4). Nested in this block of data is the device type and its communication code (PTTC/EBCD, correspondence, APL, and so forth).

RDEVBLOK information is available to the program support representative, who has privilege class E command access to the CP storage areas that contain the control blocks associated with CP and virtual machines. The command classes assigned to the CE (classes G and F) do not permit display of VM/370 control program areas. In VM/370, the 1052 and 3215 are architecturally and functionally equivalent. Therefore, the 1052 and 3215, along with 3210 and 2150, all equate to the same hexidecimal equivalent. This causes the output of the OER summary records to reflect the device type as a 1052 rather than as a real device type.

Conditions for Invoking Tests

Processor Reliability

VM/370 and the system it runs on must achieve a basic reliability for CE diagnostic use. This is achieved by hardware configurations that meet VM/370 criteria for system generation. Refer to "Part 1. Planning for System Generation", in _VM/370 Planning and System Generation Guide_ for a complete list of devices, model numbers, and features supported by VM/370. _VM/370 Planning and System Generation Guide_ also gives sample hardware configurations that comprise the minimum system requirements for running VM/370 after system startup.

Service time might be arranged for CE diagnostic or maintenance usage of spooling devices and tape drives after system generation and initialization procedures are complete. This is possible since VM/370 may be able to continue operating for a short time without the availability of these devices. The availability of the spooling devices and tape drives for CE diagnostic testing, however, depends on priorities established by system initialization personnel. When availability has been established with the system operator, these devices can be placed offline for service.

Note: In attached processor operations, a serious malfunction on the attached processor could cause uniprocessor mode on the main processor.

Basic VM/370 performance adequate to run diagnostic tests can be assumed if the CP MESSAGE command correspondence between the CE and the system operator can be established and the system performs and responds to other requests and queries of system personnel.

Hookup to the Test and Diagnostic Residence Device

System diagnostic OLTSEP and OLTS normally reside on a tape or a disk pack. Therefore, besides establishing a path to the device to be tested, a data path must be established to the device that contains the test. This is done by having the system operator mount the pack cr load the tape containing the diagnostics (assuming that the CE is at a remote location) onto a suitable device. The operator must then prepare the test device that is to be used (insert cards, tape, make the device "ready", and so on) by the diagnostic tests. The operator then, by the use of the ATTACH command, attaches these devices to the CE's virtual machine.

Note: Only the system operator with the proper privilege class can invoke the ATTACH command. After the ATTACH command has been invoked, a confirming message to that effect appears at the CE's console. To achieve this, the CE must have previously lcgged cnto his virtual machine.

Successful Logon

Successful logon indicates terminal and communication path reliability, CE virtual machine accessibility, and compliance with VM/370 logon requirements.

Logon is successful if the terminal responses progress as far as the LOGMSG.

Example

```
vm/370 cn line         ijh359  qsyosu

logon ce
ENTER PASSWORD

LOGMSG-08:04:35 03/30/74
.
.
.
```

If LOGON is not accomplished, one of the follcwing has occurred:

- A 2741 terminal is connected to VM/370 via a 3704/3705 line in NCP (network control program) mode. The Return key must be pressed before the "vm/370 online" message appears at the terminal.
- The user's terminal is connected to a 3704/3705 line that is in NCP mode and has the Multiple Terminal Access (MTA) feature. The MTA feature requires special sign-on procedures (see VM/370 Terminal User's Guide for details).
- The data path between the control unit and the terminal is incomplete or the terminal itself is not operational.
- The virtual machine does not exist or is already in use by another user.
- LOGON procedures or VM/370 terminal entry rules have been violated.
- VM/370 program not operational.
- Successful logon would exceed current system operating parameters; therefore, it is not allowed.

LINE AND TERMINAL FACILITY CHECK

Data path failure or VM/370 not operational may result in failure to receive the "vm/370 online" message. This problem can be resolved by communicating to the computer site to determine whether or not VM/370 is indeed operational and if the terminal in question is online and enabled to the system. If the system operator response is affirmative, then local testing and communication line checks should be initiated.

Violation of VM/370 terminal logon procedures may prompt the terminal message "restart." In this situation, use the VM/370 Terminal User's Guide to recheck the logon procedures. If satisfied that the procedures invoked are correct, check the device for correct local operation, then initiate tests to check the data path to the control unit.

If the CE receives "virtual machine already in use," or "...exceeds system parameters" in addition to "vm/370 online" and possibly RESTART, use the MESSAGE command and contact the system operator for assistance.

To invoke the MESSAGE command for communication to the system operator any of the following forms may be used:

```
message cperator message-text
msg op message-text
m op message-text
```

If a message response from the system operator is nct forthcoming cr the message cannct be entered via terminal equipment, then other media must be used to establish communication with the system operator.

If an ackncwledgement of the message is received by the CE, then line and terminal communication have been successfully established.

USING THE LOGON COMMAND

If line and terminal performance is satisfied, failure to log on can be the result of improper use of the LOGON command and its associated operands. The correct procedure involves knowing the correct password and CE userid.

Assume that the LOGON was invoked correctly tut the response was a facsimile of cne of the following:

```
MAXIMUM USERS EXCEEDED
INVALID USERID
USERID NOT IN CP DIRECTORY
PASSWORD INCORRECT
ALREADY LOGGED ON LINE raddr
```

CE action should be to relay this data via the MFSSAGE command tack to the system operator. The system operator can then defer maintenance to a later time period, or can arrange an environment so that CE LOGCN is successful. Once logon is successful, the CE can use OLTSEP and the online test sections (OLTS). Samples of invoking OLTSEP are shown later in the text.

To assist in the process of entering the tests or other data, the CE can use VM/370's four input line edit functions; they are described in the *VM/370 Terminal User's Guide*. Briefly, an @ symtol when entered deletes the previously entered character on the logical input line. The ¢ symbol deletes the previously entered input line. The # symbol is used to signal the end of a logical input line so that multiple logical input lines can be entered on the same terminal input line. The " symbol is issued as an escape character, that is, it cancels the line edit function of a following @, #, ¢, or " character and allows that line edit functicn character to be accepted as data.

After successful logon, the CE must enter the environment needed to perform the function he desires. To store or display storage or registers in the virtual machine, the environment tc use is CP; to invoke CPEREP to edit error recording, the CE must first perform an IPL and enter the CMS environment. To use the online test sections, the OLTSEP program must be loaded into the virtual machine. Details cn logon, the initial program load (IPL) operation, and the virtual logoff process (ending the terminal session) are described in *VM/370 Terminal User's Guide*.

Invoking OLTS

To load the OLTSEP and OLTS programs from a tape or a disk, the CE must have the operator attach the IPL device containing the tests to his virtual machine. This can be accomplished by asking the operator, or, if the CE is at a remote location, the CE can communicate by sending him messages on a terminal such as the following:

```
msg operator mount my diagnostic pack on 181 - ce

msg operator put scratch tape on 583 - test device
```

The operator will then mount and make ready the devices desired fcr testing by the CE. The operator then issues the ATTACH command; the CE's terminal then indicates:

```
DEV 181 ATTACHED
DEV 583 ATTACHED
```

In the case of system-owned volumes (DASD devices) that cannot be directly attached to the CE's virtual machine, testing is facilitated by defining the device as a full extent minidisk with a relocation factcr of 0 (that is, the DASD device is described in the system with its minimum and maximum cylinder values). The CE can then use the LINK command to link to the device (via password identification) in write mode to execute the prescribed tests.

Under these conditions, the diagnostic used _must_ confine its write operation to the CE cylinders only. Use of system owned disks by the CE can be achieved by directory entry in the CE's virtual machine or by the use of the LINK ccmmand.

The CE is now ready to load his virtual machine with OLTSEP. This is done by issuing the IPL command to the addressed device. Upcn completion of the operation, OLTSEP responds tc the CE's terminal as though he were using the real system console (3215, etc.). Figure 5 shows a sample of the complete lcgon operation, OLTS testing, and logoff operation as initiated from a 2741 console. The 2741 sample sessicn shown in Figure 5 would suffice for diagnostics run from a display terminal. The major difference is that the exclamation point is nct indicated cn the screen's output area; instead, a change in screen status information is indicative of attention signaling.

Notes:

1. While the execution of OLTS in a virtual machine is usually identical to execution on a real machine, differences exist fcr specific types of test devices. Terminal control devices (2701, 2702, 2703, 3704, and 3705) often appears tc respond differently to tests executed in a virtual machine. A control run should be executed against a device that is known to be operating correctly, and the error shown should be considered the normal results when the OLTS are run in virtual machine.

2. If the OLT section selection (DEV/TEST/CPT) defines the same terminal that is serving as the virtual system console, refer tc the topic, "Invoking OLTS to Virtual Machine Console Terminals."

```
vm/370 online     xdhxjr qsyosu


logon ce
ENTER PASSWORD:

LOGMSG - 08:04:35  03/30/76
* CP/CMS
* COLD START 16:30
LOGON AT 08:32:14 EST THURSDAY 03/30/76
msg operator attach oltsep tape on 382 as 382 <--- (Note 1)
TAPE 382 ATTACHED
msg operator attach dasd 333 as 333 <---(Note 1)
DASD 333 ATTACHED
CP
define 009 as 01f <---(Note 2)
i 382 <---(Note 3) (see Note 4 if OLTSEP Release 4.0, 4.1, or 5.0
                    is being used)
04 SEP188D ENTER DATE IN FOLLOWING FORMAT 'MM/DD/YY'
r 04,'03/17/72'
04 SEP330D ENTER TIME IN THE FORMAT 'HH.MM.SS'
r 04,'08.30.00'
SEP392I OLT LOAD ADDRESS IS 020000 HEX.
log
LOGOFF AT 08:41:13 ON 03/30/76
```

__Notes__:

1. Messages to the operator to attach devices is necessary only if the CE invokes tests from a remote site. In most cases, the CE is on-site and simply asks the operator to fulfill his requests.
2. Normally, system consoles have an address of 01F. Therefore, assembled diagnostic tape reflects this address. Virtual consoles are configured as 009 in the system directory. To resolve the conflict of different addresses, the DEFINE command is used as shown. If the CE wishes to run diagnositc excerises on his own virtual console, he should see the topic "Invoking OLTS to Virtual Machine Console Terminals."
3. Initialize and load the device that contains the OLTSEP and OLTS program.

4. Loading OLTSEP Release 4.0, 4.1, or 5.0 into the VM/370 environment may cause the program to enter a loop condition because of the manner in which external interrupts are processed. To circumvent this problem, the CE can, before issuing the IPL command, either:

   a. Turn off the virtual machine's interval timer by issuing:

   ```
   set timer off
   ```

   b. Initially set the virtual machine's interval timer to a maximum value via the STORE command, thus:

   ```
   store 50 ffffff00
   ```

Figure 5. 2741 Printout--A Typical CE Terminal Session Using OLTSEP-OLTS (Part 1 of 2)

```
SEP102I OLTS RUNNING
SEP107I OPTIONS ARE NTL,NEL,NPP, FE,NMI, EP, CP, PR, SI,NRE
01 SEP105D ENTER DEV/TEST/OPT/ <------(Note 5)
r 01,'333/3830a-z/nfe,pp(3)/'
SEP158I  S T3830A   UNIT 0333
SEP210I ROUTINE 0003 BYPASSED, MANUAL INTV REQUIRED.
SEP158I  T T3830A   UNIT 0333
! <------(Note 6)
SEP158I   S T3830B   UNIT 0333
CP
log
LOGOFF AT 08:41:13 ON 03/30/76
```

Notes:

5. Description of OLTSEP test options are disclosed in the CE document IBM Maintenance Program: OLTSEP Operator's Guide, Order No. D99SEPD.
6. Observe that in this example, a long string of OLT sections were requested to run on unit 0333. The exclamation point (!) indicated is produced by the CE pressing the attention key twice quickly on the console. This allows the CE to enter the CP environment to perform some virtual machine function; and, at the same time, temporarily suspends the previously engaged operation. In this instance, the CE chose to log off the system. This action relinquishes the user's allotted storage and temporary disk space, which then can be allocated to other users. If, however, the program OLTS sections were not interrupted, the program would have concluded normally by reissuing the following line at the conclusion of the current set of test requests.

   ```
   01 DEP105D ENTER DEV/TEST/OPT
   ```

   This response indicates that new values are to be entered for subsequent test runs.

Figure 5. 2741 Printout--A Typical CE Terminal Session Using OLTSEP-OLTS (Part 2 of 2)

INVOKING OLTS TO VIRTUAL MACHINE CONSOLE TERMINALS

Situations can occur where the CE may wish to initiate OLTSEP and run OLT sections on the same device. In such cases, spurious results can occur. The reason for this is that the data and control path to the device are being used by two independent programs and, as a consequence, format and control switches set within the control unit or the device by one program may be incompatible with the operation of the other program. As a case in point, assume a CE wishes to run diagnostic tests on his virtual console, a 3277. The CE logs onto the VM/370 system, loads OLTSEP and directs the OLT sections to be run on the same terminal. OLTs expects a nonformatted screen. The display screen has previously been formatted by VM/370 to be compatible for its own use. Thus, displayed results are dissimilar to expected OLT test patterns.

To circumvent this, the CE must logon to another terminal and then have the system operator attach the 3277 to be tested to the CE's virtual machine (using the real device address). By exercising the device in this manner, any conflict in the use of control and data paths is avoided.

It is permissible, in some cases, to designate the virtual console as the test device without great conflict. The reason for this is that OLTS and VM/370 service the device in a similar manner. The 2741 serving as the virtual console and as the test device falls into this category. Use Figure 5 and substitute values.

OLTS-FRIEND

A sample of an OLTS-FRIEND operation invoked from a virtual machine environment is shown in Figure 6. To make the example more meaningful consult _IBM Maintenance Program--Online FRIEND OS/DOS_ (D99-0200A). Observe that invoking OLTS-FRIEND employs the same mechanics as invoking other OLTS sections from a System/370 environment.

```
logon ce
ENTER PASSWORD:

LOGMSG - 9:35:28 03/30/76
*PLANNED SHUTDOWN AT 1700 TODAY FOR HDWR ADDN & PROGRAM CHANGE
*QUERY LOG FOR ADDITIONAL DATA
LOGON AT 11:24:45 EST WEDNESDAY 03/30/76

TAPE 381 ATTACHED
DASD 131 ATTACHED
CP
ipl 381
DISABLED WAIT STATE. CP ENTERED; REQUEST, PLEASE<--(Note 1)
CP
query lines
CONS 009 ON DEV 04B<------(Note 2)
st b48 00000009
STORE COMPLETE
ext
04 SEP188D ENTER DATE IN FOLLOWING FORMAT 'MM/DD/YY'
r 04,'03/30/76'
04 SEP330D ENTER TIME IN THE FORMAT 'HH.MM.SS'
r 04,'03/30/76'
```

Notes:

1. OLTSEP expects a console address of 01F tc be used as the input device for inserting OLTS and device values. The virtual console address was assigned at system generation time and resides in the user directory. When CLTSEP attempted to send a message to the console address specified ty storage locaticn B48, CP recognized that no such virtual device existed; therefore, the virtual machine's OLTSEP operation was suspended and the virtual system entered the wait state in the CP environment. To resolve the differences between the console addresses, the CE can either change the virtual console address or redesignate the console address called for in the OLTSEP program. In this example, the CE chose the latter technique by employing the CP QUERY command to find the virtual address of his console. Then, using the CP STORE command, placed the address in the proper OLTSEP program location. Resumption of OLTSEP operation is invoked ty using the CP EXTERNAL command (the virtual machine's external interrupt). interrupt).
2. 009 indicates the virtual console address and 04B represents the true line address to which the terminal is connected.

Figure 6. 2741 Printout--A CE Terminal Sessicn Invoking OLTS-FRIEND (Part 1 of 2)

```
SEP302I OLT LOAD ADDRESS IS 020000 HEX.
SEP102I OLTS RUNNING
SEP107I OPTIONS ARE NTL,NEL,NPP, FE,NMI, EP, CP, PR, SI,NTR
01 SEP105D ENTER DEV/TEST/OPT/
r 01,'131/t0200a//'
SEP125I UNREADABLE LABEL ON 0131
SEP137I CSW 000104600E000005
SEP137I SNS 000800400700000000000000000000000000000000000000
04 SEP139D REPLY B TO BYPASS, R TO RETRY, P TO PROCEED
r 04,'p'
SEP158I  S T0200A  UNIT 0131
SEP100I FRIEND RUNNING - V/L=10
SEP100I DATA AREA IN BYTES = 122864
04 SEP120D CAN VOL DATA ON 0131 BE DESTROYED. REPLY YES OR NO.
r 04,'yes'
SEP100I ALL OF DEVICE 131 ALLOCATED
04 SEP101D ENTER FRIEND COMMAND
r 04,'seek/cyl=50/hd=00'
04 SEP101D ?
r 04,'rh into $a'
04 SEP101D ?
r 04,'go'
04 SEP101D ?
r 04,'dump $a'
SEP100I 02200E  00003200 00000000 00000000 00000000
04 SEP101D ?
r 04,'end'
SEP FRIEND ENDING
SEP158I  T T0200A  UNIT 0131
SEP107I OPTIONS ARE NTL,NEL,NPP, FE,NMI, EP, CP, PR, SI,NTR
01 SEP105D ENTER DEV/TEST/OPT/
!
CP
logoff
LOGOFF AT 11:52:02 ON 03/30/76
```

Figure 6. 2741 Printout--A CE Terminal Session Invoking OLTS-FRIEND (Part 2 of 2)

OLTSEP-RETAIN

To invoke the facilities of RETAIN/370 through the media of OLTSEP in a virtual machine, the following must be invoked in the order listed.

1. Establish line communication to RETAIN center.
2. Using the CE meter key, turn the "degate interface" lamp off on the 2955.
3. Enable the 2955 via the enable/disable switch.
4. The CE logs onto the system from a terminal.
5. The system operator, per the CE's request, will vary the 2955, test device(s) and the OLTSEP device online.
6. The system operator, using the ATTACH command, connects the device(s)/line(s) to the CE's virtual machine.
7. The CE issues an IPL command to the device that contains OLTSEP.
8. The CE provides the date and time in response to the date and time prompt message and then to the following message:

   ```
   01 SEP105D ENTER DEV/TEST/OPT/
   ```

   The CE responds with:

   ```
   r01,'rei   <----(Retain input request)
   ```

   The system, if it honors the request, will respond with

   ```
   SEP163I * RETAIN/370 READY
   01 SEP105D ENTER DEV/TEST/OPT/
   ```

From this point on, the on-site CE and the operator at the RETAIN remote location can communicate via terminal action by using the Response 3 format as shown:

```
R 03, 'message'
```

Device testing can be invoked by either the RETAIN site personnel or the on-site CE after the initial test on the specified device was initiated by the on-site CE and RE is specified in the option field.

The terminal data that appears on one terminal will be a replica of the data that appears on the other terminal after hookup conditions are satisfied.

__Note:__ Be aware that the RETAIN operation utilizing the OLTS tests from a virtual environment is subject to the same restrictions as are other programs run in other VM/370 virtual machines. See __VM/370 System Messages__ for the list of VM/370 restrictions.

A result of an OLTSEP/RETAIN operation is shown in Figure 7.

```
vm/370 online    ljh359 qsyosu

l ce
ENTER PASSWORD:
service
DMKLOG092E DEV 009 NOT DEFINED; CONS 009 ALREADY DEFINED
DMKLNK108E CE 230 NOT LINKED; VOLID PIDSK1 NOT MOUNTED
LOGMSG - 08:57:37 EDT FRIDAY 03/30/76
* RUNNING SYS056 IPL3
LOGON AT 09:35:54 EDT FRIDAY 03/30/76
DMKDSP450W CP ENTERED; DISABLED WAIT PSW
CP
msg op attach 380 to ce as 380
m op attach line 080 to ce as 080
m op attach oltsep to 137 as 137

TAPE 380 ATTACHED
define 009 as 01f
DASD 137 ATTACHED
CONS 01F DEFINED

LINE 080 ATTACHED

i 137
04 SEP188D ENTER DATE (AND TIME)-'MM/DD/YY,HH/MM/SS'
r 04'03/30/76,11/00/00'
SEP392I OLT LOAD ADDRESS IS 020000 HEX.
SEP102I OLTS RUNNING
SEP107I OPTIONS ARE NTL,NEL, FE,NMI, EP, CP, PR, SI, NTR
01 SEP105D ENTER DEV/TEST/OPT/
r 01,'rei'  <-------(initial RETAIN request)
SEP163I * RETAIN/370 READY
01 SEP105D ENTER DEV/TEST/OPT/
r 01,'380/2400a/nfe,re/'   <-------(Initiated by site CE.
                                    Note: re=RETAIN option)
SEP119I NON-STANDARD TAPE LABEL ON 0380
04 SEP139D REPLY B TO BYPASS, R TO RETRY, P TO PROCEED, MAY DESTROY
  DATA
r 04,'p'
SEP158I  S T2400A $ UNIT 0380
SEP158I  T T2400A $ UNIT 0380
SEP107I OPTIONS ARE NTL,NEL,NPP,NFE,NMI, EP, CP, PR, SI,NTR, RE
01 SEP105D ENTER DEV/TEST/OPT/
R01,'/2400A-D//'   <-----(initiated by RETAIN site)
SEP158I  S T2400A $ UNIT 0380
SEP158I  T T2400A $ UNIT 0380
SEP158I  S T2400B $ UNIT 0380
SEP158I  T T2400B $ UNIT 0380
SEP158I  S T2400C $ UNIT 0380
SEP158I  T T2400C $ UNIT 0380
SEP158I  S T2400D $ UNIT 0380
SEP158I  T T2400D $ UNIT 0380
SEP107I OPTIONS ARE NTL,NEL,NPP,NFE,NMI, EP, CP, PR, SI,NTR, RE
01 SEP105D ENTER DEV/TEST/OPT/
r 03,'is this test sufficient?'  <-----(Message from site CE)
01 SEP105D ENTER DEV/TEST/OPT/
R 03,'YES THANKS TERMINATE OPERATIONS'  <----(Response from RETAIN)
01 SEP105D ENTER DEV/TEST/OPTION/
!!   <--(Attention key hit twice to enter CP environment for logoff)
CP
log
LOGOFF AT 11:41:05 ON 03/30/76
```

Figure 7.   2741 Printout--Terminal Session Showing Use of RETAIN/370

Basic Terminal Check Via the MESSAGE Command

By the use of the MESSAGE command, basic terminal checkout can be made at any time VM/370 is operational and the related interface to the terminal is enabled.[1] The MESSAGE command, a CP command, can be used by any user on any terminal prior to and after the LOGON operation. With the MESSAGE command, the CE can:

- Send a message to any logged on user
- Solicit a response from the System Operator
- Send a message to himself

The requirements for the basic check of a VM/370 terminal and line condition are:

-- The VM/370 program must be operational
-- The teleprocessing line must be enabled or the related 3704/3705 loaded, ready, and its resources enabled
-- The MESSAGE command format must be familiar to the CE
-- The keyboard must be unlocked

The format of the MESSAGE command is described in the *VM/370 CP Command Reference for General Users*. Essentially, you enter the command MESSAGE, MSG, or a valid contraction of MESSAGE. Then, the user identification of the virtual machine that is to receive the message is entered, followed by the message text. However, if you are sending a message to yourself use an asterisk (*) in place of the userid.

When the asterisk (*) operand is used prior to a successful logon operation, the system creates a VMBLOK and then unites the LOGON keyboard with the line address (XXX). This is the three-digit hexadecimal address of the 270x communications line that connects to a terminal device. This is indicated in the response.

Note: If the asterisk (*) operand is used after logon, then the valid userid is inserted in response messages.

The following is an example of a basic terminal and line checkout without involving logon procedures using a 2741 terminal. Assume that terminal hookup has been established per instructions outlined in the *VM/370 Terminal User's Guide* and that the terminal is placed in COMMUNICATE mode.

[1] If the terminal is a 2741 connected to VM/370 via a 3704/3705 line in NCP mode, the Return key must be pressed before the "vm/370 online" message will appear at the terminal. If the terminal device is connected to a 3704 or 3705 line that is in NCP mode and has the MTA feature, an additional terminal sign-on procedure is necessary in order to use the MESSSAGE command prior to the LOGON operation (see the *VM/370 Terminal User's Guide* for details).

Example

```
     vm/370 on line <---------(ATTN key [or its equivalent] pressed)

     msg * abcdefghijklmnopqrstuvwxyz0123456789
     (text of message sent to self)

     MSG FROM LOGON058:  ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789
     (response of message to self prior to logon)
```

Note: VM/370 normally translates lowercase alphabetic data to uppercase in responding to terminal MESSAGE commands.

Basic Terminal Check Via ECHO Command

Assume that the CE can successfully logon tc his assigned virtual machine. Assume also that his terminal is failing because of a local cr line condition. In such a situation, instead of invoking OLTS, the CE can invoke the CP ECHO command to exercise the terminal. ECHO may serve as an adequate test for printing and keyboard problems.

The ECHO command differs from the MESSAGE command in that there is no translation tc uppercase letters in processing the command text. That is, the command will be returned to the terminal in the same form in which it was transmitted. The ECHO command is exclusive to the G privilege class.

Information on the format and use of the ECHO command is detailed in the VM/370 CP Command Reference for General Users. In summary, to use the ECHO command, you must be lcgged onto your virtual machine and ycu must be in the CP environment. With these conditions satisfied, ycu enter the ECHO command and specify the number of times you want the message that you will enter returned to you. After this is done, the system prompts you for the message text. If the ECHO command is entered without the return message value, ECHO will default to one response fcr each line entered.

Figure 8 is a terminal session using the CP ECHO command.

```
 vm/370 online    Xdhxjr qsyosu
 logon ce
 ENTER PASSWORD:

 LOGMSG - :35:28 03/30/74
 *RUNNING SYS009 - DIRECTORY CHANGE SCHEDULED AT 1800
 *QUERY LOG FOR ADDITIONAL INFORMATION
 LOGON AT 11:24:45 WEDNESDAY 03/30/74

 CP  <---virtual machine is in the CP environment
 echo 3  <---echo environment entered; three responses elected
 ECHO ENTERED:  TO TERMINATE TEST TYPE END
 ENTER LINE
 NOW is THE time  <---(text plus return key depression)
 NOW is THE time  <---(three system responses)
 NOW is THE time
 NOW is THE time
 end  <--- (end entered by CE to exit from ECHO environment)
 CP   <--- (system now back in CP environment; echo test complete)
```

Note: It is imperative to type "end" at the end of the test inasmuch as a subsequent command entered would be treated as ECHO text.

Figure 8. 2741 Printout--A CE Terminal Session Invoking ECHO Command

Section 3. Error Handling

- Overview of I/O Error Handling
- Environmental Data Recording
- Error Recording Area
- Recovery Management Support
- Machine Check Handler
- Channel Check Handler

Overview of I/O Error Handling

In attached processor systems, only the main processor has real I/O processing capabilities. Therefore, when the attached processor encounters channel operations, the channel program is reflected to the main processor for execution. I/O operations and I/O error recording for attached processor systems are no different from the techniques employed on uniprocessor systems.

I/O events initiated by CP fall into one of two general categories:

CP I/O Requests:

- Paging
- Spooling
- CMS I/O (diagnose interface)

Virtual User Requests:

- Any I/O request issued by an operating system running in a virtual machine

When an I/O event results in an I/O error, the action taken by CP depends on the type of request: CP-related request, or virtual user request. Since each virtual machine is a functional equivalent of an IBM System/370 and its associated I/O devices, CP reflects virtual machine I/O errors to the virtual machine that initiated the I/O event; this is done so that the errors appear the same as if the user were running standalone on a real machine. Device error recovery, error recording, and messages issued to the operator of the virtual machine depend upon the virtual machine's operating system, release levels, and other data, and are not part of CP.

I/O error recovery is attempted for CP-initiated I/O operations to CP-supported devices, and for virtual user I/O operations initiated through VM/370's Diagnose Interface, which is mainly an interface for CMS.

When channel status word indicators show that an error occurred during a CP-initiated I/O event, a device-dependent error recovery procedure is invoked and a cycle of restarts begins that continues either until the error is corrected or until it is determined to be fatal (uncorrectable). The controlling routine of the cycle is the I/O error recovery routine which, upon exit, may indicate that:

- The activity is to be retried
- The error is fatal
- The error has been corrected

If the I/O error recovery routine indicates that the I/O event is to be retried, the I/O supervisor queues the I/O request for processing. The restart takes place when the channel restart routine, during its normal processing, initializes the I/O operation. After the I/O activity is completed, the I/O supervisor tests the recovery-in-progress bit and causes control to return to the I/O error routine, even if no errors occurred during the retry.

I/O error routines count the number of retries and indicate to the I/O supervisor that the error is fatal when the maximum allowable number has been reached. When an I/O error routine indicates that an error condition is fatal, the I/O supervisor places a "permanent error" return code in the user's IOBLOK and returns control to the caller.

An error is considered to be corrected when no errors occur during a retry of the I/O activity. For corrected errors, the I/O supervisor places a "completed without error" code in the user's IOBLOK, updates Statistical Data Recording (SDR) counters in the SDRBLOK, then continues processing as it would have if there had been no error the first time the activity was attempted.

I/O Error Recording and SVC 76

Because the IBM operating systems that are commonly run in virtual machines have adopted the convention of using SVC 76 to do their error recording, CP can centralize nearly all error recording. The following types of errors are recorded in the error recording area of the VM/370 system residence pack.

- VM/370 spooling errors

- VM/370 paging errors

- I/O errors resulting from user's CMS or RSCS operation

- I/O error events resulting from a user-initiated diagnostic interface

- I/O errors or error-related data compiled by an operating system running in a virtual machine and interpreted by CP when the operating system issued SVC 76 in an attempt to do its own error recording.

When CP intercepts an SVC 76 issued by an operating system running in a virtual machine, it records the error on the VM/370 error recording cylinders and passes control back to the virtual machine at the instruction following the SVC 76. CP handles SVC 76 in this way only if all of the following conditions are met:

1. All pertinent passed parameters concerning the error record are valid for CP's implementation of error recording.

2. There can be a resolution of virtual address to real device address.

3. The record type matches a CP-supported type.

If any of these conditions is not met, VM/370 does not record the error on its error recording area; the SVC 76 interruption is reflected back to the virtual machine for the virtual machine operating system to do the error recording. Note that the management and processing of SVC 76 is unaffected by the virtual machine assist and the Extended Control Program Support for VM/370 (VM/370 ECPS) on systems supported by VM/370.

ERROR RECORDING--VM/370 VERSUS AN OPERATING SYSTEM IN A DEDICATED ENVIRONMENT

An operating system in a dedicated environment (for example: DOS operating standalone in a System/370 Model 145) exercises complete control over the entire hardware configuration. In the utilization of this hardware, there is usually a direct relationship between the residence of data and the address used to access this data (device address as well as the access location within the device). Error recording, therefore, can be accomplished easily because any data- and address-handling schemes used by that operating system can be used to create a factual error record.

With VM/370, these operating systems operate under the control of the Control Program (CP) component of VM/370. A system resource under DOS or OS constitutes real hardware with its real hexadecimal hardware address and data records residing at precise locations on that device. In most cases, under VM/370's control, the following are virtual, not real: (1) the device, (2) the data address, and (3) the device type parameter used by other control programs operating in the virtual machine environment. For example, what DOS considers a 2311 device residing at address 214 with certain data at track location 10 could, in reality, be a 2314 device with a device address of 310 and a track location of 65.

A virtual 3330, Model 1 mapped onto a real 3330, Model 11 would be another example. Other devices, whether or not supported by VM/370, can be dedicated to an operating system, in which case VM/370 does not translate data addresses or device types. Device address mapping, however, may still be done.

In 3850 Mass Storage Systems, the 3330V (3330 virtual volume) associated with a given CPUID in a MSS application is specified as input to the *OS/VS Mass Storage Control Table Create Program* (GC35-0013). The Mass Table Create creates IODEVICE cards that are used as input to the VS system generation procedure. CP's DMKRIO configuration must agree with the input to Mass Table Create and the OS system generation configuration. This addressing agreement is necessary because CP provides the real I/O interface from VS1/VS2 operating systems to MSS devices. Operating protocol dictates that in using the ATTACH or DEFINE commands, the virtual address must match the real address (VM/370 generated addresses) as all errors are reflected to the virtual machine for error recovery and the logging process.

Note: Devices dedicated to a virtual machine's operating system may have no address or device translation. These devices may or may not be supported by VM/370's recovery management support (RMS) and error recording package.

As stated previously, the operating system in the virtual machine not only executes its own I/O error recovery, but can generate its own LOGREC data. Keep in mind that these records usually reflect the virtual values as VM/370 CP initiates all I/O privileged instructions with translated values applicable to the real hardware. As a consequence, sense data reflected to the virtual machine because of I/O error conditions is associated with a logical device. This virtual machine LOGREC data is then of very limited use to the CE since he may not know the real device address corresponding to the virtual address from which the error was recorded. The SVC 76 interface capability of VM/370 takes care of this problem.

SVC 76 is the supervisor call used by the IBM operating systems to record either statistical data or a permanent I/O error incident. VM/370 traps a valid SVC 76 event issued by an IBM operating system running in a virtual machine environment and captures permanent I/O error incidents as well as other specific recording types as explained in the following paragraphs.

The minimum release level of program systems that support SVC 76 is as follows:

VM/370 (running in a virtual machine environment) (Release 2)
OS/360 (Release 21)
VS1
VS2 Release 1 (with single address space)
VS2 Release 2 (with multiple address spaces)
DOS Release 27 (with PTF as required)
DOS/VS

SVC 76 Handling of I/O Device Errors

When a valid SVC 76 is issued by an operating system running in a virtual machine, VM/370 traps it. VM/370 checks the error recording data parameters and the type of error record passed with the SVC 76. If invalid, the SVC 76 is reflected to the virtual machine's operating system. If valid, VM/370 will:

1. Translate virtual device addresses found in the record to real device addresses.

2. Record the error in VM/370's error recording area.

3. Inform the VM/370 system operator of the I/O error via a console message.

4. Return control to the operating system at the instruction address following the SVC 76 instruction, thereby causing the SVC 76 to act as a no-op instruction as far as the virtual machine is concerned.

Processing the SVC 76 interrupt in this manner bypasses the error recording mechanism of the virtual machine and allows the virtual machine's job processing to continue after VM/370 gathers the data for the error recording record.

Any of the above mentioned operating systems is run standalone, then when the SVC 76 is issued in the process of I/O error recovery, SVC 76 generates an interrupt that signals the operating system supervisor to record the error on the operating system's LOGREC data set.

In either case, as far as job processing is concerned, SVC 76 and I/O error recording is not apparent to the user.

SVC 76 Handling of Channel Errors

Channel errors are handled differently from device errors. CP records a channel check in the VM/370 error recording area immediately and informs the VM/370 system operator of the channel check via a console message (but for a channel data check, no message is issued). Then CP reflects the channel check to the virtual machine. After seeing the error, the operating system in the virtual machine issues SVC 76. Since CP has already recorded the error, CP ignores the SVC 76 and reflects it to the virtual machine (without translating virtual channel and device addresses in the error record to real addresses). The reflected SVC 76 then causes the operating system in the virtual machine to record the channel error in its own LOGREC data set.

SVC 76--Parameter Passing

VM/370 examines the contents of general registers 0 and 1 to determine if valid conditions exist for handling the error recording data.

If the system is OS (Release 21 or above), VS/1, VS/2, or VM/370 (in a virtual environment) then:

```
General register 0 = two's complement of the error record length
General register 1 = address of the record
```

If the system is DOS (Release 27) or DOS/VS then:

```
General register 0 = address of the error record minus 8
General register 1 = Byte 0, Bit 0 must be a 1,
                     Bytes 1, 2, and 3 contain the CCB address
                     (DOS control block for I/O)
```

VM/370 then locates the formatted error record and examines the record header for a valid operating system identity (ID). The record type then examined to determine if it is one of the supported recording types.

Record Modification for VM/370 Error Recording

The error record is modified, changing virtual information to real. The fields modified vary with the type of record.

- Type 30, OBR (Outboard Recorder)

Common Fields:

Primary and Alternate CUA are replaced with the real device address corresponding to the virtual device address.

CPUID (CPU model number) is replaced with the real machine model number.

JOBID is replaced with the virtual user ID.

Device Dependent Fields:

For dedicated DASD devices no modification is required. For nondedicated DASD devices, the following modifications are required:

Seek Address, the relocation factor, found in the VDEVBLOK, adjusts the seek address field of the record in order to reflect the true real seek address.

Home Address Read, the relccation factor, found in the VDEVBLOK, adjusts the home address read field in order to reflect the true real hcme address.

Volume ID, the volume label in the RDEVBLOK, replaces the volume ID in the record.

3330, 3340, 3350, and 2305, the relocation factor in the VDEVBLOK, adjusts the cylinder address portion of the sense data (sense bytes 5 and 6).

Virtual 2311 on 2314, the device type is changed to 2314 and sense byte 3 is altered to reflect 2314 information. For this situation, the 2314 module ID usually found in the sense byte is nct available.

Note: The failing CCW and CSW fields are not altered. This results in the CCW address in the CSW and data address in the CCW being virtual, not real.

- Type 40, 41, 42, 44, 48, and 4F programming abend records:

Common Fields:

CPUID (CPU model number) is replaced with the real machine model number.

JOBID is replaced with the virtual user's ID.

- Type 60, DDR (Dynamic Device Reallocation)

  Common Fields:

  CPUID (CPU model number) is replaced with the real machine model number.

  JOBID is replaced with the virtual user's ID.

  Primary CUA of "from" Device is replaced with the real CUA corresponding to the virtual device.

  Primary CUA of "to" Device is replaced with the real CUA corresponding to the virtual device.

- Type 70, MIH (Missing Interrupt Handler)

  Common Fields:

  CPUID (CPU model number) is replaced with the real machine model number.

  JOBID is replaced with the virtual user's ID.

  CUA is replaced with the real CUA corresponding to the virtual device.

  Primary CUA is replaced with the real CUA corresponding to the virtual device.

  Device Dependent Fields:

  DASD: For dedicated DASD devices, no modification is required. For nondedicated DASD devices, the following modification is required:

  Volume Serial Number is replaced with the volume label from the RDEVBLOK.

- Type 91, MDR (Miscellaneous Data Records)

  Common Fields:

  CPUID (CPU model number) is replaced with the real machine model number.

  JOBID is replaced with the virtual user's ID.

  Primary CUA is replaced with the real CUA corresponding to the virtual device.

Recording of the Error Record

The recording of the error record is accomplished by using existing routines in DMKIOC, DMKIOE, and DMKIOF.

I/O Error Messages

In most cases, CP provides the I/O interface to real devices for the initiated I/O activities of virtual machines. Therefore, encountered I/O unit check conditions (OBR 30 error recording condition) are recorded in the VM/370 error recording area. In addition, a message is sent to the VM/370 primary system operator informing him of the real unit address of the device and the userid that is performing the I/C. The same action occurs when a unit check is detected on a dedicated device where SVC 76 is invoked. This message also appears when VM/370 error routines are invoked for recording counter and buffer overflow statistics for various devices, for recording demounts, and for recording general statistical data in VM/370s error recording area.

I/O Error Recovery -- Detailed Description

I/O error recovery is attempted for CP-initiated I/O operations tc CP-supported devices, and for user-initiated operations to CP-supported devices through use of the diagnose interface. The primary control blocks used for error recovery are the RDEVBLOK, the IOBLOK, the SDRBLOK, and the IOERBLOK. In addition, auxiliary storage may be obtained to generate recovery channel programs. The initial error is first detected by the I/O interrupt handler. An IOERBLOK is constructed and a sense command is performed to place the sense data into the IOERBLOK. The I/O supervisor then examines the IOBLOK to determine if the event was initiated by CP or by a virtual machine. For the case cf a virtual machine event, the I/O interrupt is reflected to the virtual machine. Fcr CP-related I/O errors, device-dependent error recovery procedures are invoked. Unit record errors are handled by the CP spooling routines; terminal errors are handled by the console handling routines; and DASD and tape errors are handled by other CP routines.

In attached processor applications, I/O processing and I/O error recovery procedures are essentially the same as uniprocessor methods. Virtual I/O can occur on either processor, however, the end processing of the virtual-to-real CCW string can only be executed on the main processor. Only the main processor has real I/O capabilities.

During an I/O operation, the control block linkage shown in Figure 9 is in effect.

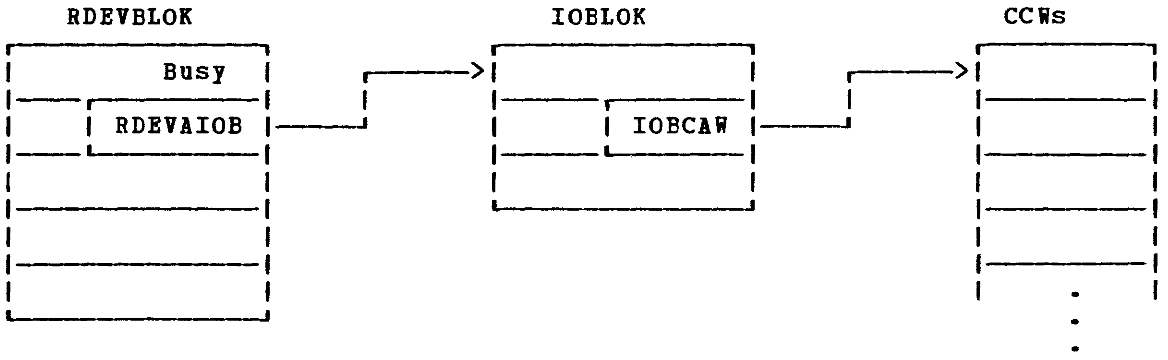


Figure 9. I/O Operation Control Block Linkage

When channel status word indicators show that an error occurred during I/O activity, the I/O interrupt handler constructs an IOERBLOK. The I/O supervisor performs a sense command to place the sense data in the IOERBLOK, and the error CSW is also placed in the IOERBLOK. When the sense operation is complete, the I/O supervisor invokes the I/O error recovery routines for sense data analysis with the control block structure shown in Figure 10.

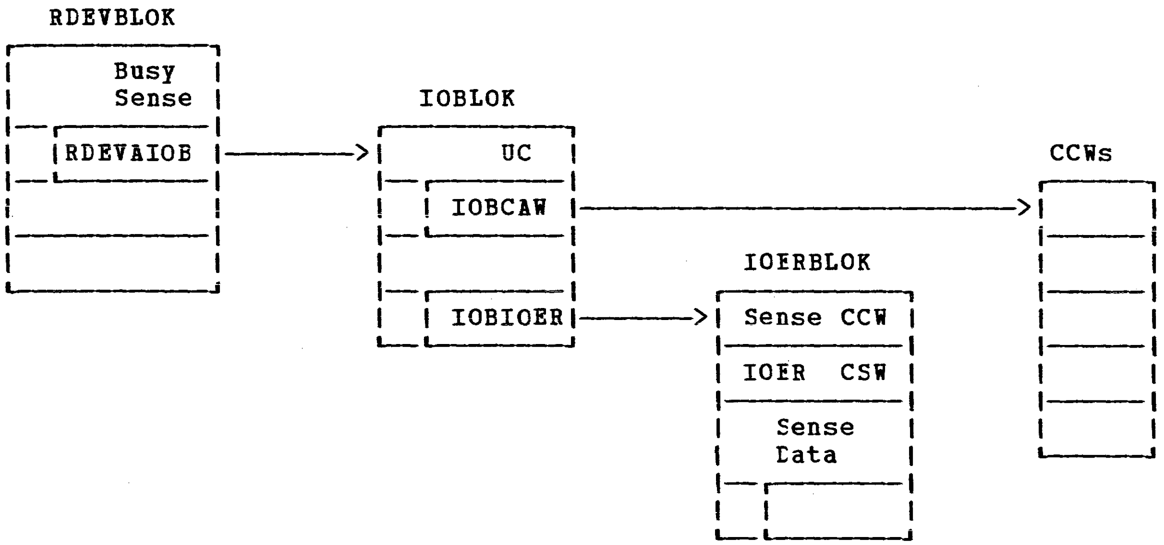


Figure 10. I/O Error Recovery Control Block Structure for Sense-Byte Analysis

The error recovery procedure analyzes the error and, if recovery is possible, builds a recovery CCW string to be executed to attempt recovery. In order to preserve the original IOERBLOK, the error recovery procedure places the pointer to the IOERBLOK in the RDEVBLOK. The error recovery procedure keeps track of the number of retries in the IOBRCNT field of the IOBLOK. This count is used to determine if a retry limit has been exceeded for a particular error. On initial entry from the I/O supervisor, the count is zero; and for each retry attempt, the count is increased by one. The error recovery procedure communicates to the I/O supervisor by way of the IOBSTAT and IOBFLAG fields of the IOBLOK. When retry is to be attempted, the error recovery procedure turns on the restart bit in the IOBLFLAG field of the IOBLOK. In addition, the ERP bit of the IOBSTAT field in the IOBLOK is turned on to indicate to IOS that the error recovery procedure is to receive control when the I/O event has completed. This enables the error recovery procedure to receive control even if the retry was successful so that SDR counters can be updated and any storage that was obtained for the recovery process can be relinquished. When recovery is attempted, the IOBRCAW in the IOBLOK is set to point to the recovery CCW string and control is returned to the I/O supervisor with the control block linkage as shown in Figure 11.

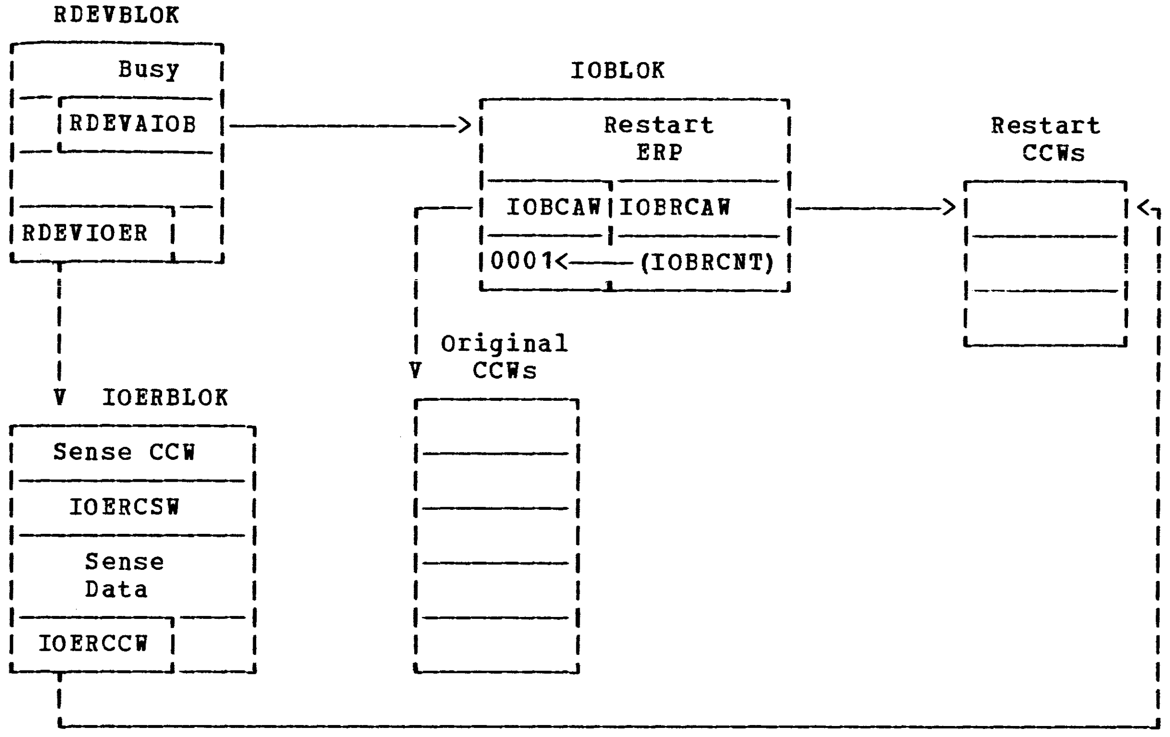


Figure 11. Control Block Linkage for Retry

If the retry attempt is successful, control is still returned to the error recovery procedure. The ERP flag bit in the IOBLOK determines this.

If another unit check occurs on the retry attempt, the I/O supervisor will follow the same procedure as the initial error sequence by building an IOERBLOK and performing a sense command. When the I/O error recording routine returns control to the I/O supervisor, the ERP bit of the IOBSTAT flag in the IOBLOK being set causes control to be returned to the ERP.

The error recovery procedure notes that this is a retried operation (ERP flag and IOBRCNT field nonzero). If the recovery procedure retries the operation, the restart procedure is again followed with the IOBRCNT value increased by one. The IOERBLOK and recovery CCWs associated with the unsuccessful recovery attempt are purged by returning the storage to the system. (Remember that the original IOERBLOK is being saved by placing a pointer to it in the RDEVBLOK.) It can be seen that the error recovery procedure, not the I/O supervisor, is the routine controlling recovery attempts and determining when an error is a permanent one. The SDR counters are updated using the sense information from the original IOERBLOK. Figure 12 shows the control block relationship while updating the SDR counters. The repetitive correction cycle is followed until recovery is accomplished or the error recovery procedure determines (from the retry count, IOBRCNT) that the error is permanent. If the specified number of retries fails to correct the error, the fatal flag (permanent error) in the IOBLOK is turned on (IOBSTAT=IOBFATAL) and control is returned to the I/O supervisor. The I/O supervisor will call the I/O error recording routine. The I/O error recording routine analyzes the sense data to determine if a recording condition exists, if so, an I/O error formatted record is constructed and the record is queued to be written out in the I/O error recording area of the VM/370 system residence device. If the user of the virtual machine has privilege class F, the I/O error recording routine tests flags in the

RDEVBLOK to determine if intensive recording mode is in effect for this device. If the conditions are met, an I/O error record is created. This record is constructed and recorded as described previously. Control is returned to the I/O supervisor, which reflects the error to the user of the I/O operation.

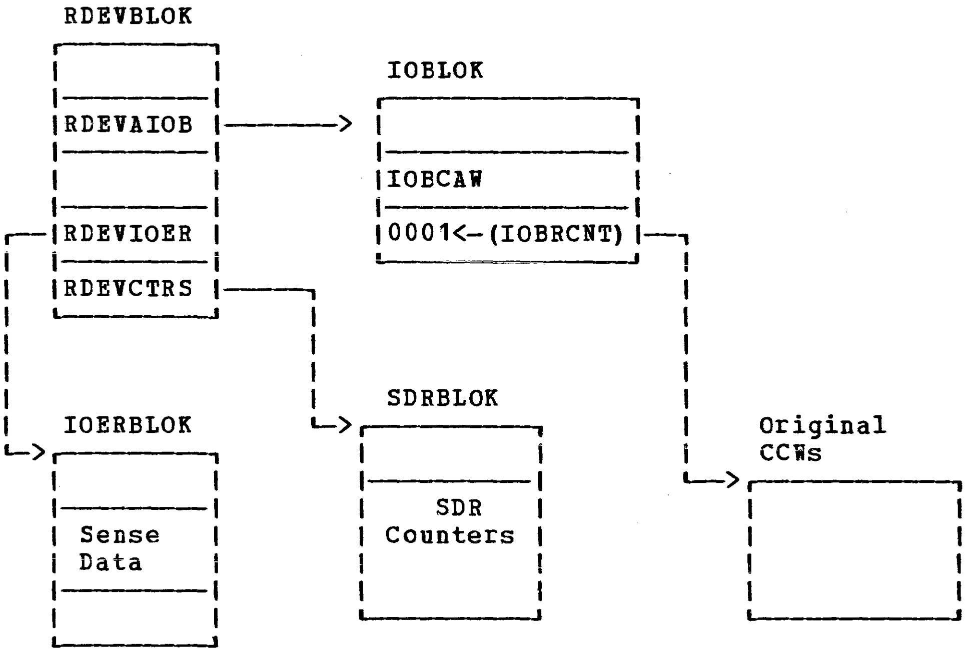


Figure 12. Control Block Relationship for SDR Counter Update

I/O Error Recording and Error Recording Area

The error recording facilities of VM/370 format and record outboard error records, and record formatted machine check and channel check records created by the RMS routines of VM/370.

The error recording routines of VM/370 do not actually perform I/O operations. Instead, the I/O error routines treat the error recording area allocated on the VM/370 system residence pack as a logical extension of VM/370 storage. These extensions of VM/370 storage are in the form of logical pages that can be read and written out of by the paging supervisor of VM/370. The error recording routines place multiple error records within a page; when an error record is assembled within a page, a pointer is updated to indicate the beginning of any unused area. The next error record is checked to see if it can be contained in the remainder of this page. If it can, the error record is read into the page and the pointer is updated to again reflect any residual storage available for the next error record. This process continues until an error record is encountered that cannot be contained within the page. When this happens, the page is scheduled to be read out to the next available slot in the error recording area and a new page in storage is assigned to accept and retain the error record. The process continues in like manner.

The error recording area is from two to nine adjacent cylinders assigned on the system residence volume. The starting cylinder number and number of cylinders are specified in VM/37C generation procedures. When the error recording area is 90% full, and again when 100% full, the I/O error routines instruct the VM/370 system operator to invoke the CPEREP command to print (or create a tape of) the error data and erase the recording area. Errors are recorded in the order of occurrence until the allotted space is exhausted.

With the support provided for the 3031, 3032, and 3033 processors, CPEREP need not be aware of the content or the EC level of the processor logouts in order to format machine check and channel check records. Format and content information is provided via the SRF (Service Record File) device. Frames (records containing text and scan buffer codes) are maintained on the SRF device by customer engineering, and software makes use of these frames to interpret and format inboard errors. Whenever the VM/370 error recording cylinders are formatted on a 3031, 3032 or 3033 processor, the SRF is accessed and the frames are retrieved, formatted as frame records, and recorded at the beginning of the VM/370 error recording area by the process described above. When CPEREP is invoked, these frame records are used to format MCH and CCH records for the printed report.

The SRF device is accessed by VM/370 to read frame data (a) during VM/370 system initialization if the error recording cylinders have not been previously formatted; and (b) as a result of running CPEREP with the CLEARF operand. To ensure that the VM/370 control program has access to the SRF device, the following steps should be followed to activate the SRF:

1. Check that the I/O interface for the service support console is enabled.

2. Obtain the configuration frame on the service support console.

3. The SRF appears disabled until accessed on the 3032. Activate the SRF on the 3031 and 3033 by selecting SRF mode A2.

4. VARY ON cuu (SRF address) on the operator's console.

5. ATTACH cuu * cuu to attach the SRF device to the operator's console; or ATTACH cuu userid cuu to attach the SRF device to the console of the class F user who runs CPEREP.

In a 303x environment, access to the SRF device by an SCP in a virtual machine must be considered when planning to run EREP to print the error log belonging to that virtual machine. The SRF device must be accessible to the operating system in a virtual machine when it initializes its error log in order that frame data may be read from the SRF. The VM/370 system operator should attach the SRF device to the virtual machine before that SCP initializes its error log (for example, in the case of OS/VS2, before running IFCDIP00); the virtual machine operator should then vary the SRF online.

The error recording facilities of VM/370 are of the following types:

OUTBOARD RECORDING:

- Statistical data recording
- Permanent I/O errors
- Environmental data records
- Intensive mode recordings
- Specific DASD recording requirements
- Specific tape recording requirements
- Software abend records

INBOARD RECORDING:

- Machine checks
- Channel checks

I/O Statistical Data Recording (SDR)

Statistical data recording is the accumulation and the recording of I/O error statistics that relate to specific devices. VM/370 supports SDR recording for CP-initiated I/O events by building and maintaining device statistics tables (counters) in the SDRBLOK associated with the I/O device. These counters are updated when a device-dependent error recovery procedure (ERP) determines that the error has either been corrected successfully or is a permanent error. SDR counters are updated based on the sense information in the original IOERBLOK. The updating of the counters is done asynchronously. If the update function causes a counter overflow, a short OBR record is built. The OBR record is then placed on the asynchronous output queue. This causes the OBR record to be written on the error recording area asynchronously.

When the SHUTDOWN command or NETWORK SHUTDOWN command is issued by the system operator, any devices that have SDR counters associated with them cause control to be passed to the I/O error recording routine to format a short OBR record. (A long OBR is formatted for 3400 tapes.)

The VARY OFFLINE command or NETWORK VARY OFFLINE command of a device that has associated SDR counters also causes control to be passed to the I/O error recording routine to format a short OBR record (a long OBR is formatted for 3400 tapes).

The VARY OFFLINE, SHUTDOWN, NETWORK VARY OFFLINE, and NETWORK SHUTDOWN commands result in an OBR record being written to the error recording area synchronously.

Permanent I/O Error Recording

Permanent I/O errors related to VM/370-initiated I/O events are recorded by the I/O error recording routines of VM/370. When a device-dependent error recovery procedure determines that an I/O event cannot be successfully recovered, the fatal flag is turned on in the IOBLOK and control is returned to the I/O supervisor. The I/O supervisor invokes the I/O error recording routines with the control block structure as shown in Figure 13. The I/O error recording routines format the error and record it on the error recording area.

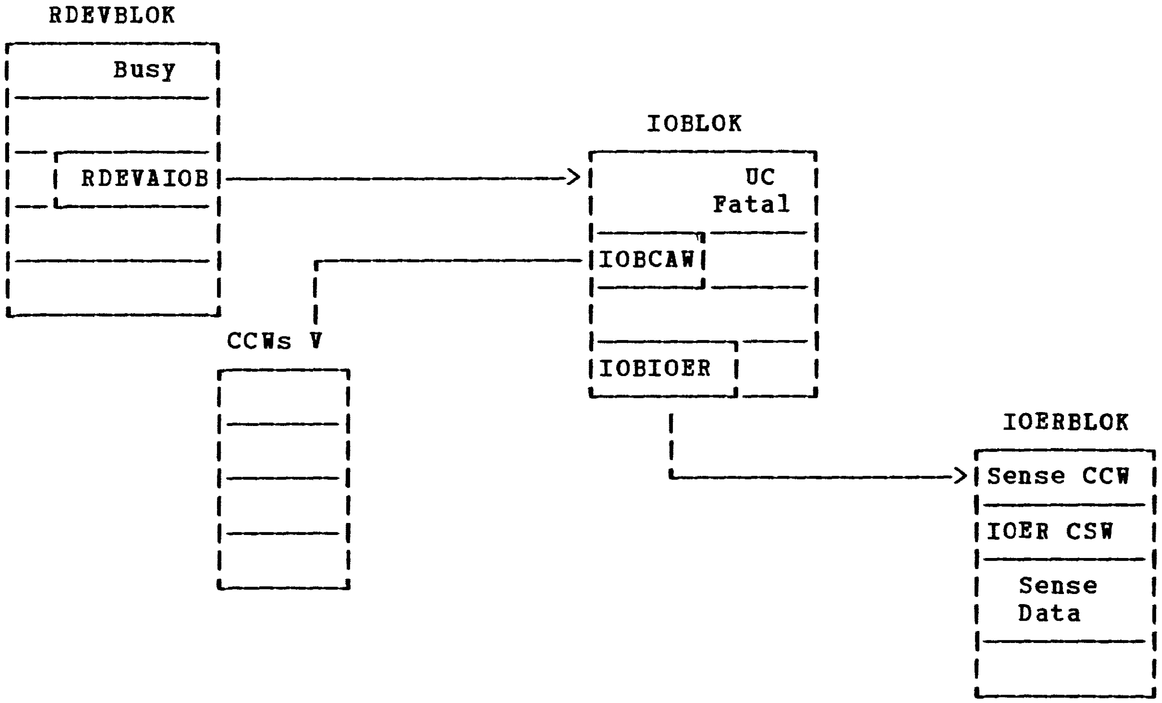


Figure 13. Control Block Linkage--Fatal Error Condition

Environmental Data Recording

When the I/O supervisor receives a unit check interruption from a 3330, 3340, 3350, or 2305, the error recovery procedure is invoked. If the sense information indicates that an environmental data recording is required, the error recovery procedure builds the necessary channel program to retrieve the error log data from the file control unit.

The sense data that indicates this condition is as follows:

Machine	Sense Byte	Bit	Condition
2305	2	0	Buffer Log Full
3330,3340,3350	2	3	Environmental Data

The manner in which the error recovery procedure passes the data to the I/O error recording routine differs between the 2305 and the 3330/3340/3350 as shown in Figures 14 and 15, respectively.

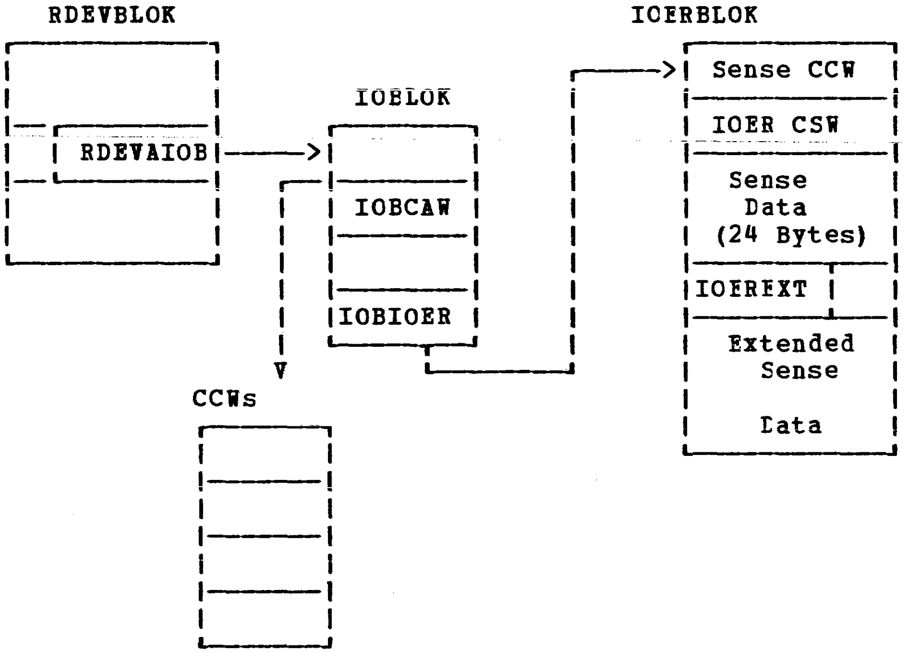


Figure 14. 2305 Control Block Structure--Environmental Data Recording

The IOEREXT field in the IOERBLOK contains the length in doutlewords of the extended data area. The I/O error recording routine builds an environmental data record in the proper format, queues the request fcr recording, and returns to the I/O supervisor. The CASC error recovery procedure retries the operation and normal processing ccntinues.

A different control block linkage exists on 3330/3340/3350 environmental data recordings due to the amount of data. The DASD errcr recovery prccedure builds multiple IOERBLOKS and chains them together tc pass the data to the I/O error recording routines.

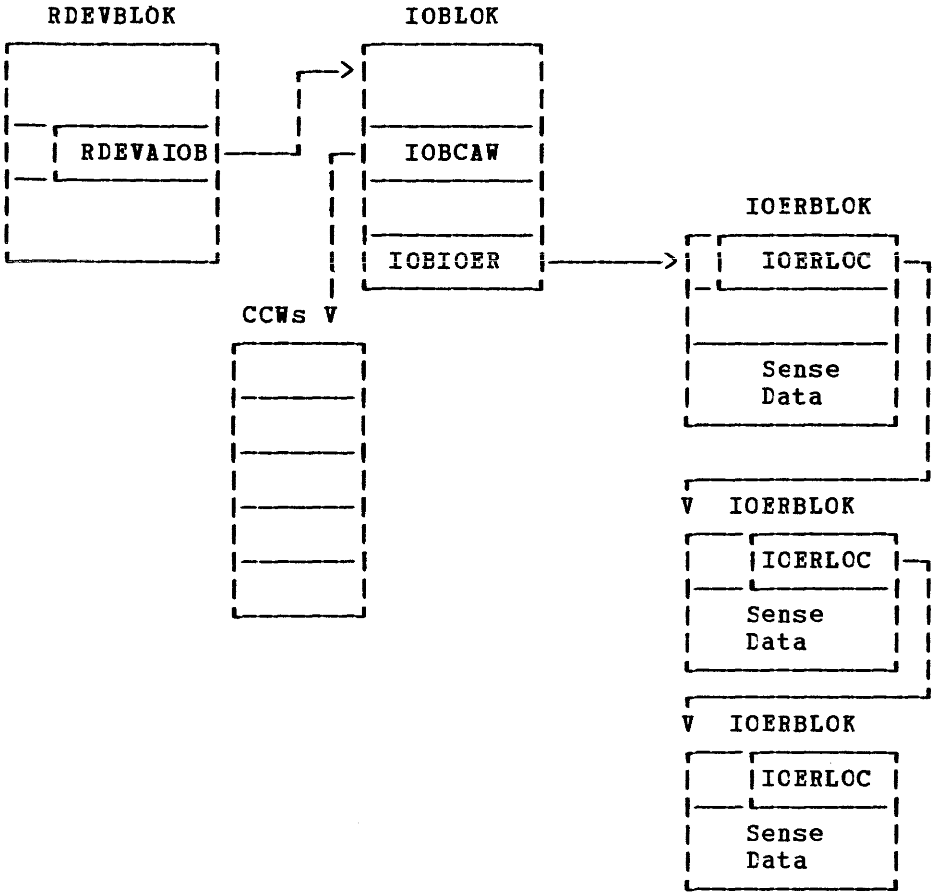


Figure 15. 3330/3340/3350 Control Block Structure--Environmental Data Recording

The IOERLOC pointer in the IOERBLOK points to the next IOERBLOK cn the string. The error recovery procedures obtain free storage and construct IOERBLOKs to be placed on the string until the buffer on the control unit is completely unloaded. The I/O error recording routine builds an environmental data record in the prcper format, queues the request for recording, and returns to the I/O supervisor. The error recovery procedure retries the operation and normal processing continues.

Intensive Mode Recordings

On any unit check occurrence the I/O supervisor invokes the I/O error recording routines to determine if the conditions for intensive mode recording are satisfied. Intensive mode is an error recording mode whereby errors are recorded for a specific device that achieves a unit check condition and sense data that matches previously defined sense data values. The SET RECORD command starts intensive mode. If intensive mode recording conditions are satisfied, an I/O error record is constructed, formatted, and recorded in the I/O error area of the VM/370 system residence device, and a flag is set in the IOERBLOK to indicate that the error has been recorded (IOERFLG2 = IOERCEMD). This recording is done for CP-owned devices as well as dedicated devices attached to virtual machines. The user who initiated the intensive mode operation must run the CPEREP program to retrieve the records created while this option was active. No messages occur to inform either the VM/370 system operator, or the virtual machine user when a recording is made or when intensive mode is disabled by the I/O error recording routines after the tenth recording. Intensive mode (SET RECORD option) can be invoked only on one real hardware device at any time and only by a user with the privilege class F command usage.

Note: For the privilege class F virtual machine all normal error recording is suspended except for the 'intensive mode' selected device. If however, the F class user invokes SVC 76 to pass a record to CP to record, CP will honor such a request.

VM/370 I/O Error Recordings

OBR records are written if any of the following conditions exist for all but the privilege class F user (unless intensive mode is specified for a particular device).

- An unrecoverable (permanent) I/O error which was initiated as a VM/370 I/O task (CP request).
- Counter overflow statistics (SDR count).
- SHUTDOWN and NETWORK SHUTDOWN commands (devices with SDR counters).
- VARY OFFLINE (devices with SDR counters).
- 2305/3330/3340/3350--Equipment Check.
- 2305/3330/3340/3350--Busout Check.
- 2305/3330/3340/3350--MDR record on BUFFER UNLOAD command (X'A4' or X'24') to a nondedicated DASD storage device by a virtual machine.
- 3340--Seek Check.
- 2305/3340/3350--Data Check.
- 2305/3330/3340/3350--Overrun.

Error Recording Record Layout

Error recordings vary in length and format depending on the malfunction or the device encountered. Data that relates to channel check, machine check, or unit check conditions is arranged in byte-formatted records in the error recording area. Figures 16, 17, 18, 19, and 20 describe the layout and data length of the fields within defined record types. Use Figure 21 with these charts to ascertain the origin of particular fields of data.

The paired alphabetical characters shown in the fields of Figures 16, 17, 18, 19, and 20 correspond to the location codes in Figure 21. The location code in conjunction with the type of error (MC, UC, CC) indicates the availability of that data and what data block or function contains or generates this data.

Figures 22 and 23, using the same paired alphabetical character scheme, describe the 24-byte header that precedes the error record. Figure 23 describes the contents and source of the fields indicated in Figure 22.

For additional information on error record layout as used by the CP component of VM/370 refer to *VM/370 Data Areas and Control Block Logic*. For information on the printout format of supported error record types, refer to the *OS/VS*, *DOS/VSE*, *VM/370 Environmental Recording Editing and Printing* (*EREP*) *Program*. Support logic for this program is contained in the *OS/VS*, *DOS/VSE*, *VM/370 Environmental Recording Editing and Printing* (*EREP*) *Program Logic*.

```
┌────────────────────────────────────────────────────────────┐
|          Header Record        24 Bytes              AA      |
|────────────────────────────────────────────────────────────|
|          Job ID                8 Bytes              AC      |
|────────────────────────────────────────────────────────────|
|          Failing CCW           8 Bytes              AJ      |
|────────────────────────────────────────────────────────────|
|          CSW Stored            8 Bytes              AK      |
|          When error was detected                            |
|────────────────────────────────────────────────────────────|
|Device Depletion|Channel & Unit Address| Device Type         |
|Count of Double-| of Failing Device    | Characteristic      |
|   Word Size    |                      |                     |
| 1 byte    BD   | (SEC) 3 bytes     AD |   4 bytes  AN       |
|────────────────|──────────────────────|─────────────────────|
| SDR Work Area  |Channel & Unit Address|Retry   |Sense Byte|
|     Count      | Physical Location of |Count   |  Count   |
|                | Failing Spindle (PRI)|2 Bytes| 2 Bytes  |
|  1 byte   BE   |           3 bytes  AM | AQ     |    BF    |
|────────────────────────────────────────────────────────────|
|   Volume ID Associated With I/O Error   6 bytes     AR      |
|────────────────────────────────────────────────────────────|
|   Last Seek Address of DASD             8 bytes     AS      |
|────────────────────────────────────────────────────────────|
|   Actual Home Address Read              6 bytes     AT      |
|────────────────────────────────────────────────────────────|
|   SDR Work Area (RDEVCNTS) variable length          BH      |
|────────────────────────────────────────────────────────────|
|   Sense Data           variable length                      |
|   Physical identity of device when available        AU      |
└────────────────────────────────────────────────────────────┘
```

Figure 16. Unit Check Record Layout (Long)

```
┌────────────────────────────────────────────────────────────┐
|   Header Record                                            |
|                                31 bytes             AA      |
|────────────────────────────────────────────────────────────|
|   Device Dependent Data                                    |
|                                variable length      BG      |
└────────────────────────────────────────────────────────────┘
```

Figure 17. Ncnstandard Record Layout

```
┌────────────────────────────────────────────────────────────┐
|   Header Record   24 bytes                         AA       |
|────────────────────────────────────────────────────────────|
| Device Type      | SDR Work Area| Channel and Unit Address|
|  Characteristic  |  Count       |  of Failing Device      |
| 4 bytes   AN     | 2 byte   BE  |   3 Bytes          AD   |
|────────────────────────────────────────────────────────────|
|               SDR Work Area                                |
|                          Variable Length           BH      |
└────────────────────────────────────────────────────────────┘
```

Figure 18. Unit Check Record Layout (Short)

```
 _________________________________________________________
|             Header   24 Bytes                  AA       |
|---------------------------------------------------------|
|         Program Identity  8 Bytes              AB       |
|---------------------------------------------------------|
|           Job Identity  8 Bytes                AC       |
|---------------------------------------------------------|
|       Machine Check Old PSW 8 Bytes            AE       |
|---------------------------------------------------------|
|         Machine Independent Logout             AF       |
|                  280 Bytes                              |
|---------------------------------------------------------|
|          Machine Dependent Logout              AG       |
|       (Extended Logout) variable length                  |
|---------------------------------------------------------|
|             Damage Assessment                  AH       |
|                  80 Bytes                               |
|       Identifies the Extent of Damage Found             |
|         by a Recovery Management Program                |
|_________________________________________________________|
```

Figure 19. Machine Check Record Layout

```
 ___________________________________________________________
|        Header          24 Bytes                      AA   |
|-----------------------------------------------------------|
|        Job ID          8 Bytes                       AC   |
|-----------------------------------------------------------|
|                                                           |
|        Active I/O Units          16 Bytes            AI   |
|                                                           |
|      Contains up to eight devices on failing channel      |
|                                                           |
|-----------------------------------------------------------|
|        Failing CCW          8 Bytes                  AJ   |
|-----------------------------------------------------------|
|    CSW Stored when Error Detected   8 Bytes          AK   |
|-----------------------------------------------------------|
|  Extended Channel Status  |                               |
|         Word          AL  |    Device Type            AN  |
|         4 Bytes           |      4 Bytes                  |
|---------------------------|-------------------------------|
|            | Channel & Unit|      Multiprccessing          |
| Channel ID |   Address     |        Information       AW   |
|   1 Byte   |   3 Bytes     |Fail-| Re-    | CPU-1 | CPU-2 |
|            |               |ing  |served  |Channel|Channel|
|            |               |CPU  | for    |       |       |
|            |               |and  | IBM    |       |       |
|            |               |Stat-| use    |       |       |
|            |               |us   |        |       |       |
|    AO            AM         1Byte  1Byte   1 Byte  1 Byte |
|-----------------------------------------------------------|
|                      Channel Logout                       |
|                                                           |
⊥                     Variable  Length                      ⊥

 _     Reflects the Internal Hardware Status of            _
|      the Failing Channel at I/O Interrupt Time            |
|                                                      AP   |
|___________________________________________________________|
```

Figure 20. Channel Check Record Layout

Location Code	Header	MC	UC	UC Short	CC	Non Std	Data Recorded	From
AA	x	x	x	x	x	x	Chart	
AB		x					Program ID	NA
AC		x	x		x		Job ID (USERID)	VMBLOK
AD			x	x	x		Channel & Unit	RDEVBLOK
AE		x					Mach Ck Old PSW	MCH Buffer
AF		x					Mach Ck Independent Logout	MCH Buffer
AG		x					CPU Hardware Logout	MCH Buffer
AH		x					Damage Assessment	NA
AI					x		Active I/O Units on Channel	CCH
AJ			x		x		Failing CCW	IOBLOK
AK			x		x		CSW	IOERBLOK
AL					x		Extended CSW	CCH
AM			x	x	x		Physical Spindle or Channel & Unit	IOERBLOK IOBLOK
AN			x	x	x		Device Type	RDEVBLOK
AO					x		Channel ID	RCHBLOK
AP					x		Channel Logout	CCH
AQ			x				I/O Retries	IOBLOK
AR			x				Volume ID	RDEVBLOK
AS			x				Last Seek Address	IOBLOK IOERBLOK
AT			x				Actual Home Address	IOERBLOK
AU			x				Sense Data	IOERBLOK
AV							NA	
AW					x		Multiprocessing	NA
AX								
AY								
AZ								
BA								
BB								
BC								
BD			x				Device-Dependent Data Count	IOERBLOK
BE			x	x			Statistical Data Work Count	Recorder
BF			x				Sense Byte Count	IOERBLOK
BG						x	Device-Dependent	
BH			x	x			Stat. Data Ctrs	SDRBLOK

Legend:
MC = Machine Check
UC = Unit Check
CC = Channel Check
Non Std = Nonstandard

CCH = Channel Check Handler
MCH = Machine Check Handler
NA = Not Applicable

Figure 21. Record Breakdown Table (Except Header)

RECORD HEADER - 24 BYTES

```
Bits    0     8         16        24        32     40       48      56       63
        ┌────────────────────────────────────────────────────────────────────┐
        |Record|Operating|      Switches          |Re-      |Record  |Re-     |
        | Type | System  |Independent-Dependent   |served   |Count   |served  |
        |      |         |byte 0 byte 1  byte 2   |for IBM  |        |for IBM |
        | CA   |   CB    |         CC             |Use      |  CD    |Use     |
        |      |         |                        |         |        |        |
        |1 byte| 1 byte  |       3 bytes          |1 byte   |1 byte  |1 byte  |
        |────────────────────────────────────────────────────────────────────|
        |                    Date and Time   CE                               |
        |                                                                     |
        |                        8 bytes                                      |
        |────────────────────────────────────────────────────────────────────|
        |Ver-  |        CPU Serial      |   CPU Model     |   MAX MCEL        |
        |sion  |        Number          |   Number        |   Length          |
        |No. CF|          CF            |     CG          |     CG            |
        |      |                        |                 |                   |
        |1 byte|        3 bytes         |   2 bytes       |   2 bytes         |
        └────────────────────────────────────────────────────────────────────┘
```

Figure 22. Layout of the Header Portion of the Error Records (24 Bytes)

Record Header Field	Source of Data	Location
Record Type	From calling routines or type of entry	CA
Operating System	System description module	CB
Switches (dependent/independent)		CC
Byte 0		
Bit 0 Multiple Record Recording	NA	
1 NS Machine	Always 0 (using NS Clock Binary)	
2 EC Mode	PSW	
3 Reserved for IBM Use	–	
4 Time Macro Used (HHMMSS)	Always 1	
Byte 1 MACHINE CHECK		
Bit 0 Short Form	NA	
1 Record Incomplete	NA	
2 System Terminated	MCH	
3 First Record of two	NA	
4 Channel Record Included	NA	
5 Data Overlaid	NA	
6 External Machine Check	NA	
Byte 1 CHANNEL CHECK		
Bit 0 Operator Message	NA	
1 Record Incomplete	NA	
2 System Terminated	CCH	
3 Channel unsupported or failed to log.	CCH	
4 Invalid CUA	CCH	
5 Data Overlaid	CCH	
6 ERP in Progress	NA	
Byte 1 UNIT CHECK		
Bit 0 SDR dump (EOD)	RECORDER	
1 Temporary error	IOBLOK	
2 Short record	RECORDER	
3 MP system	NA	
4 CPU B	NA	
5 Volume dismount	NA	
6 SVC requested	NA	
Byte 2 MISCELLANEOUS DATA Recorder (Nonstandard)		
Record ID Code		
01 = 3330		
02 = 2305-2		
03 = 3270		
04 = 3211		
05 = 3705		
08 = 2715		
09 = 3340		
0A = 3330-11		
11 = 3350		
12 = 2305-1		
FF = Reserved for IBM Use		
Record Count	Always 01	CD
Date & Time	RECORDER	CE
CPU ID and Serial	Store CPU ID	DF
Max MCEL Length	Store CPU ID	CG
Version Number	Store CPU ID	CF
NA = Not Available		

Figure 23. Header Record Table

VM/370 Recovery Features -- Introduction

The primary objectives of VM/370's recovery management support are:

- To reduce the number of system terminations that result from machine malfunctions.

- To minimize the impact of such incidents.

The programmed recovery, which accomplishes these objectives, allows system operations to continue whenever possible, and records the system status for all errors. The MCH (Machine Check Handler) and CCH (Channel Check Handler) provide the recovery management functions of VM/370.

MACHINE CHECK HANDLER

A machine malfunction can originate from a processor, processor storage, control storage, or a channel group. When any one of these fails to work properly, the hardware tries to correct the malfunction. If the machine recovers from the error through its own recovery facilities, a machine check interruption notifies the appropriate machine check handler routine. The machine check handler records the fact that the machine operated improperly. Concurrently with the machine check interruption, the processor logs out fields of information in processor storage. This information describes the cause and nature of the error. MCH analyzes this information and builds the machine check record.

If the machine fails to recover from the error through its own recovery facilities, a machine check interruption occurs, and an interruption code indicates that the recovery attempt failed. The machine check handler then analyzes the data and tries to keep the system as fully operational as possible. The cause of the malfunction determines what action the machine check handler takes:

- Resume operations, leaving no adverse effects on the system.

- Resume system operations by terminating the virtual machine that was interrupted.

- Isolate the failure to a page and flag the page as invalid or unavailable for use by the paging supervisor.

- Place the system in a disabled wait state.

- If the 158 AP or 168 AP operating in attached processor mode had an unrecoverable malfunction occur on the attached processor in problem program state, resume operations in uniprocessor mode.

CHANNEL CHECK HANDLER (CCH)

The channel check handler is a resident program that receives control from the I/O supervisor when a real channel error occurs. CCH records the error. CCH reflects channel control checks, channel data checks, interface control checks, and channel interface inoperative (for a dedicated channel) to the virtual machine to allow the SCP in that virtual machine to attempt recovery, and/or initiate appropriate termination procedures. If CCH determines that system integrity has not been damaged, channel errors associated with an input/output operation initiated by CP (for example, paging or spooling) are retried by the appropriate device-dependent error recovery procedure.

If CCH determines that system integrity has been damaged (for example, if the channel has been reset, or if the device address stored is invalid), CCH places the system in a disabled wait state and sends a message to the VM/370 primary system operator. For the 4331 and 4341 processors, limited channel logout is still available, but no fixed or I/O extended logout area exists.

HANDLING OF HARD MACHINE CHECKS

If a permanent error (hard machine check) occurs on the main processor (or attached processor), the error is analyzed to determine whether or not it is correctable by programming. Time-of-day clock and timer errors that result in a machine check interruption that are not correctable and cannot be circumvented place the real computing system in a disabled wait state.

Uncorrectable or unretryable processor errors, storage errors, and storage protect key failures are handled as discussed in the following paragraphs.

Processor Errors

When a machine check interruption indicates that a processor error associated with VM/370 cannot be corrected or retried the system operator is informed of the error and the system is put in a disabled wait state. All virtual machine users must log on again. If the error is associated with a virtual machine, the user is informed of the error and the virtual machine is reset, unless it is using the virtual=real option. In that case, the virtual machine is terminated, and the user must then log on and reinitialize (via IPL) his machine.

If VM/370 is being run in attached processor mode and an uncorrectable error is encountered on the attached processor while executing in problem program state, system operation continues in uniprocessor mode on the main processor.

Storage Errors in a Virtual Machine Page

When the control program (CP) detects a permanent storage error (hard machine check) in a real storage page frame that is being used by a virtual machine, the frame is marked invalid if the error is intermittent, or unavailable if the error is solid. If the page frame has not been altered by the virtual machine, a new page frame is assigned to the virtual machine and a backup copy of the page is brought in the next time the page is referenced. All storage errors are transparent to the virtual machine user.

If the page frame has been altered, VM/370 resets the virtual machine, clears its virtual storage to zeros, and sends an appropriate message to the user. If the virtual machine is using the virtual=real option, it is terminated. In either case, normal system operation continues for all other users.

Storage Errors in the CP Nucleus

Multiple-bit storage errors in the CP nucleus cannot be corrected; they cause VM/370 to terminate. (Single-bit storage errors are corrected by ECC, as noted above.)

Storage Protect Key Failures

When intermittent storage protect key failures occur, whether associated with VM/370 or a virtual machine, the key is corrected and operation continues.

If the storage protect key error is uncorrectable (solid) and is associated with a virtual machine, the user is notified and the virtual machine is terminated. The page frame is marked unavailable. Uncorrectable storage protect key failures associated with VM/370 cause the VM/370 system to be terminated. An automatic restart reinitializes VM/370.

HANDLING OF SOFT MACHINE CHECKS

Although hard machine checks always cause a machine check interruption to occur and logouts to be written, soft machine checks are handled in one of two operating modes -- recording mode or quiet mode.

- In recording mode, soft machine checks cause machine check interruptions and write logouts.

- In quiet mode, only hard machine checks cause machine check interruptions and write logouts.

The normal operating state of VM/370 for CPU retry reporting is recording mode. For ECC (error checking and correction) reporting, the initialized (normal) state of VM/370 is model-dependent: quiet mode for all VM/370-supported processors except Models 155II and the 165II. The initial state for the 155II and 165II is record mode.

A change from recording mode to quiet mode can occur in one of two ways: when 12 soft machine checks have occurred, or when the SET MODE RETRY/MAIN QUIET command is executed by maintenance personnel.

To revert to record mode again, the command SET MODE RETRY/MAIN RECORD must be issued.

In attached processor applications, soft error recording can be set or reset for the selected processor if so desired.

If a soft machine check (a transient error) occurs while the system is in recording mode, a machine check record containing information about the error is written on the error recording cylinders. This record includes the data in the fixed logout area, the date, the time of day, and other pertinent data. The operator is not informed that a soft machine check has occurred.

If a transient error occurs while the system is in quiet mode, no machine check interruption occurs, and no logouts are written. The hardware, which had gained control when the soft machine check occurred, returns control to either VM/370 or the problem program, depending on which had control at the time the machine check occurred.

Multiple-bit ECC storage errors that occur cn a 3031, 3032 or 3033 processor are not recorded as soft errors, but rather as solid errors. If the storage frame that incurred the error is assigned to a virtual machine, it is removed from system use without any attempt to determine whether the error is intermittent. The SET MODE MAIN command is treated as invalid on these processors.

ERROR RECOVERY PROCEDURES

VM/370 includes device-dependent error recovery procedures for all devices supported by VM/370. Functionally, these procedures perform as their counterparts do in an OS or DOS system. VM/370 uses the standards used by OS cr DOS for priority of error testing, recommended retry action, and number of retry attempts for a particular error type. The error recovery procedures accept and use the extended channel status word, determine if retry is possible, and start retry actions.

CP Input/Output Errors

An appropriate error recovery procedure is invoked whenever an errcr occurs that is related to a CP input/output operation, such as paging or spooling. If VM/370 cannot correct the error, VM/370 records the errcr and notifies the system resource operator of the error.

Handling of Virtual Machine Input/Output Errors

VM/370 passes input/output errors associated with virtual machine START I/O requests to the virtual machine. The machine operating system assesses the error and attempts retry.

Note that CMS uses the DIAGNOSE interface to request VM/370 to perform input/output operations, and VM/370 then performs any necessary recovery operations for errors associated with the request.

Recording Virtual Machine Input/Output Errors

By use of the SVC 76 error recording interface, VM/370 provides uniform recording of errors encountered by operating systems running in virtual machines. VM/370 records the real address (rather than the virtual address) of a device that has an error, to allow it to be located by support personnel. The operating systems that use the SVC 76 interface are:

VM/370 Release 2 and above
  (running in a virtual machine)
DOS/VS
OS/VS1
OS/VS2
OS Release 21.0 and above
DOS Release 27.0 (requires PTF #1124)
DOS Release 27.1 (requires PTF #2051)

When an SVC 76 is issued, CP examines the error record built by the virtual machine operating system. If the information is valid, CP translates frcm virtual to real device addresses and then records If this information is invalid, CP reflects the SVC to the virtual machine and no recording takes place. Duplicate recording of errors is thus avoided.

In case of a permanent I/O error, VM/370 sends a message to the primary system operator.

If a virtual machine is using one of the above operating systems and is also using the virtual machine assist feature, then all SVCs are handled by the assist feature (except SVC 76, which is always handled by CP). However, the user can specify that CP handle all SVCs by issuing SET ASSIST NOSVC, or by including the SVCOFF option in his directory entry.

If a virtual machine is using an operating system that does not use the SVC 76 interface, both CP and the virtual machine record errors, but CP does not record all errors associated with the virtual machine.

RECORDING FACILITIES

The OS/VS environmental recording, editing, and printing program (EREP) is executed when the CMS CPEREP command is invoked. The output produced by the command is determined by information contained in the VM/370 error recording area and/or SYS1.LOGREC data on tape and the supplied CPEREP operands. The printed output from CPEREP under VM/370 has the same format as that generated by OS/VS EREP. The system can:

- Edit and print all or specific error records contained in the system error recording area or tape history file

- Create a history of records on an accumulation tape

- Erase the error recording area and, optionally, the SRF frame records on a 3031, 3032 or 3033 processor

For additional information on OS/VS EREP and the CPEREP command, a tool for software and hardware problem analysis, see Section 4.

VM/370 Repair Facilities

The Online Test Standalone Executive Program (OLTSEP) and online tests (OLT) execute in a virtual machine that runs concurrently with normal system operations. These programs provide online diagnosis of input/output errors for most devices that attach to the IBM System/370.

The service representative can execute online tests from a terminal as a user of the system; VM/370 console functions, including the ability to display or alter the virtual machine storage, are available when these tests are run. Those tests that violate VM/370 restrictions may not run correctly in a virtual machine environment.

VM/370 Restart Facilities

When either MCH or CCH determines that an error has damaged the integrity of VM/370 the system is placed in a disabled wait state. On a subsequent reloading of VM/370, the system operator can elect to execute a warm start, thus allowing terminal lines to be re-enabled automatically and completed spool files to be maintained. Storage reconfiguration data (such as page frames marked unavailable or invalid) that is acquired during the process of recovering from real storage errors is lost. After a VM/370 system failure, each user must reinitialize his virtual machine.

The same philosophy of malfunction handling is evident in Models 158 and 168 operations in attached processor mode. However, if error analysis determines that a nonrecoverable fault is associated with the attached processor while it was running in problem state, the system continues operating in uniprocessor mode on the main processor. In addition, virtual machines associated with the attached processor (AFFINITY option set to the attached processor) are set for execution on the main processor. Such virtual machines are notified of system action and their virtual machine consoles are placed in console function mode.

Resetting of a virtual machine, whether caused by a real computing system malfunction or by a virtual machine program error, does not affect the execution of other virtual machines, unless they are sharing the area in which the malfunction occurred.

Hardware Errors and Recovery Management Support

The System/370 systems supported by VM/370 have built-in error detection logic in the processor, channels, and main storage. This detection logic, working with additional hardware logic, allows the system to attempt the correction of certain error conditions. When errors are correctable, they are referred to as soft errors and have no adverse effect on VM/370. They are also usually not apparent to the virtual machine's operating system.

The following errors are not corrected by the system: channel control checks, channel data checks, and interface control checks for user SIO-initiated channel programs; and channel interface inoperative on a dedicated channel for user SIO-initiated I/O. These errors are reflected to the virtual machine.

When errors are not correctable, hardware-initiated machine check interruptions invoke the Recovery Management Support (RMS) of VM/370. RMS is part of the VM/370 Control Program, and is provided on all processors supported by VM/370 and on their supported channels.

The two primary objectives of RMS are (1) to reduce the number of system terminations that result from machine malfunctions and (2) to minimize the impact of such incidents when they occur (see Figure 24). These objectives are accomplished by programmed recovery to allow system operations to continue whenever possible and by the recording of system status for both transient (corrected) and permanent (uncorrected) hardware errors.

Function	Explanation	System Program Module
Machine Check Handler	To record all machine checks and recover from hard machine checks, or to reset or terminate virtual machines or terminate System/370 operations or if attached processor mode change to uniprocessor operations when necessary.	DMKMCH[1] DMKMCT[2]
Channel Check Handler	To record channel checks and effect proper recovery or terminate System/370 operations when necessary.	DMKCCH

[1] Both the machine check and channel check modules, where pertinent and possible, post messages to the primary system operator informing him of the status of the system.

[2] Machine check handler operations exclusive to attached processor mode termination situations, malfunction alerts, and automatic processor recovery are contained in the module DMKMCT.

Figure 24. Summary of RMS Functions

Machine Check Handler--An Overview

A machine malfunction can originate in the processor, main storage, or control storage. When any of these fails to work properly, an attempt is made by the machine to correct the malfunction. Whenever the malfunction is corrected, the machine check handler is notified by a machine check interruption. The machine check handler records the fact that the machine has failed to operate properly. Concurrently with the machine check interruption, the processor logs out fields of information in main storage detailing the cause and nature of the error. The model independent data is stored in the fixed logout area and the model dependent data is stored in the extended logout area. The machine check handler uses these fields to analyze the error and to produce the error report.

If the machine fails to recover from the error through its own recovery facilities, a machine check interruption occurs, and the fixed logout contains an interruption code that indicates the recovery attempt was unsuccessful. The machine check handler then analyzes the data and attempts to keep the system as fully operational as possible. The cause of the malfunction determines what actions MCH takes:

- Resume operations leaving no adverse effects on the system.
- Resume system operations by terminating the user that was interrupted.
- Isolate the failure to a page and flag the page as invalid or unavailable for use by the paging supervisor.
- Place the system in a disabled wait state.
- In VM/370 attached processor operations, processing may continue in uniprocessor mode if the attached processor malfunctions while in problem program state and recovery is not possible.

Note: Loss of system integrity prevents the recording of hard machine checks in the supervisor (CP). Error information of this type may be obtained through the use of the processor's hard stop facility if the machine check is repetitive.

LEVELS OF ERROR RECOVERY

Recovery from machine malfunctions can be divided into the following categories: functional recovery, system recovery, operator-initiated restart, and system repair. These levels of error recovery are discussed in order from the easiest type of recovery to the most difficult.

Functional Recovery

Functional recovery is recovery from a machine check without adverse effect on the system or the interrupted user. This type of recovery can be made by either the processor retry or the ECC facility, or the machine check handler. The processor retry and ECC error correcting facilities are discussed separately in this section since they are significant in the total error recovery scheme. Functional recovery by the machine check handler is made by correcting Storage Protect Feature (SPF) keys and intermittent errors in main storage.

System Recovery

System recovery is attempted when functional recovery is impossible. System recovery is the continuation of system operations by terminating the user who experienced the error. System recovery can take place only if the user in question is not critical to continued system operation. A system routine containing an error that is considered to be critical to system operation precludes functional recovery and would require logout and a system dump followed by reloading the system.

Operator Initiated Restart

When the errors may have caused a loss of supervisor or system integrity, the system is put into a disabled wait state. The operator must then reload the system.

System Repair

If system recovery is not possible, the system may require the services of maintenance personnel to effect a system hardware repair. System repair by this method occurs when the error is so critical to system operations that the system cannot be used to record the error.

Machine Check Handler--Summary

The machine check handler (MCH) consists of entirely resident routines in the CP nucleus.

Recovery from most machine malfunctions on System/370 is initially attempted by the instruction retry, and the error checking and correction (ECC) machine facilities. However, if the retry or storage correction is unsuccessful, if the interrupted instruction is non-retryable, or if the storage failure cannot be repaired, RMS will assess the damage and do the following:

- If the fault is an SPF key failure, refresh the key if conditions warrant such action.

- If the fault is related to main storage, either (1) refresh that page or (2) have CP flag that page as unusable and assign a new page; then refresh the data if valid to do so.

- Terminate or reset the virtual machine if the malfunction cannot be repaired but is traceable to a particular virtual machine.

- Terminate all SCP operations and post a wait state code if system integrity is lost and nonrecoverable.

- In attached processor applications, if the malfunction is associated with the attached processor while running in problem program state and attached processor recovery is not possible, cease all operations on the attached processor and allow the system to continue in uniprocessor mode on the main processor.

- If the error is a channel group inoperative on a 3031, 3032 or 3033 processor, place the system in a disabled wait state.

Any of the above conditions can produce one or more of the following results:

-- Wherever possible, a record of the error is produced in the system's error recording area.

-- Wherever possible, the primary system operator is informed of the error.

Errors corrected by instruction retry and main storage errors corrected by ECC are not reflected to the system operator's console, and these errors may or may not be recorded. See "Recovery Modes" in this section for a discussion of this.

The messages produced by the machine check handler on supported VM/370 systems are described in _VM/370 System Messages_. Wait state codes 001 and 013 produced by the machine check handler routines are also described in _VM/370 System Messages_.

The action that the machine check handler takes for a given situation is determined by the error itself, the operating environment of VM/370, and whether the system was performing a CP function, a virtual machine function, or no function at all (a loaded wait state condition when the error occurred). Figure 25 clarifies the action the system takes for the given situations.

	VM/370 Processing (CP)			Virtual Machine Processing		
Error Condition	Uniprocessor Operation	Attached Processor Operation		Uniprocessor Operation	Attached Processor Operation	
		Main	Attached		Main	attached
Invalid machine check interrupt code	1	1	1	1	1	1
Invalid PSW data	1	1	1	1	1	1
Register, Program mask instruction address invalid	!	1	1	1	3	3
System damages	1	1	1	1	3	3
TOD or CPU Clock Errors	1	1	1	1	1	3,4
Multibit (solid) Storage error	1	1	1	3,2	3,2	3,2
Multibit (intermittent) storage error	1	1	1	3,2	3,2	3,2
Storage Protect Key (solid) failure	1	1	1	3	3	3
Storage Protect (intermittent) failure	2	2	2	2	2	2
Malfunction alert	5	1	1	5	1	3,4
Channel group inoperative	1	1	1	1	1	1

<u>Legend</u>:
1 = load wait state PSW
2 = refresh for retry operation
3 = terminate the virtual machine
4 = automatic processor recovery
5 = Not applicable

Figure 25. Condition/Action Table for Uncorrectable Errors

The System/370 processors and main storage have error detection circuitry integrated into their logic. This error circuitry has additional hardware logic that allows the correction of some generated error conditions. They are:

- Certain processor error conditions
- Main storage single-bit errors (within a doubleword)

The detected processor errors cause the system to retry or circumvent the failing function, while main storage single-bit failures are corrected by error correction code (ECC) hardware logic. These errors (called soft errors), when detected and corrected, impose no adverse conditions upon the operating system. These errors are also generally not apparent to the users of the system.

Because soft errors are automatically rectified and are related to the fastest part of system hardware, they could, if no controls were imposed upon them, quickly fill the error recording area. To prevent this from happening, VM/370 maintains a program counter to record the number of soft errors that are recorded on the error recording area. This counter, initially reset on system initialization, can accumulate up to a count of 12. At the count of 12, control register (CR) 14 bit 4 (also initiated to the ON condition upon system initialization) is turned off. With the turning off of this bit, soft errors are no longer recorded in the error recording area. The system operator receives a message informing him that soft errors are no longer being recorded.

Not all of the various System/370-supported systems initiate soft error recording in the same way. All VM/370 supported processors, with the exception of the 155 II and 165 II, are disabled for ECC (error checking and correction) at system initialization. All processors, including the 4331, 4341, and the 3031 AP, are enabled at system initialization to record processor retry.

After system initialization, in order to change the mode of soft error recording, the SET MODE command must be invoked. In attached processor applications, SET MODE values can be set for either the main or the attached processor or both processors if desired. The SET MODE command can only be used by privilege class F users.

<u>Note</u>: The SET MODE MAIN command is treated as invalid on the 3031, 3032, or 3033 processors, and the 3031 AP.

On all other processors, SET MODE may be invoked in any of the following ways:

```
SET MODE MAIN RECORD [cpuid]
```

This instruction resets the error recording counter and turns CR14 bit 4 on, so that VM/370 can record ECC-corrected errors.

```
SET MODE RETRY RECORD [cpuid]
```

This instruction resets the error recording counter and turns CR14 bit 4 ON, so that VM/370 can record processor errors that were rectified by retry or circumvention techniques.

```
SET MODE MAIN QUIET
```

This instruction inhibits the recording of ECC-corrected storage errors only.

SET MODE RETRY QUIET

This instruction turns CR14 bit 4 off, thus inhibiting the recording of all soft errors.

By specifying the cpuid (valid for attached processor operations only), SET MODE values can be specified for a particular processor. By not specifying the cpuid, the SET MODE values are applicable to both processors.

While in record mode, corrected soft errors are formatted and recorded in the VM/370 error recording area. The primary system operator is not informed of the occurrence of these recordings until the recording of such errors is stopped by a command or, automatically, by count control.

Channel Check Handler

There are four types of channel checks caused by hardware errors:

- Channel data check - (Bit 44 set in the CSW).
- Channel control check - (Bit 45 set in the CSW).
- Interface control check - (Bit 46 set in the CSW).
- Interface inoperative - (Bit 46 is set in the CSW with bit 27 of the limited channel logout (LCL) set at the same time). Interface inoperative is a rare but usually persistent hardware problem with one control unit that affects the entire channel. Note: This condition is only recognized on the 3031, 3032 and 3033 processors.

The channel check handler receives control from the I/O supervisor when any of the channel checks listed above is detected. For these channel conditions, CCH does the following:

- Records the results of CCH error analysis in the IOERBLOK (I/O error block). If the error is an interface control check or a channel control check, device-dependent error retry procedures (ERP) will use the data in the IOERBLOK for the subsequent retry operation.
- Constructs a record describing the error environment.
- Informs the proper module so the error record will be written in the error recording area.
- Sends a message to the system operator regarding the error incident.
- Sets logout areas and the ECSW to all ones.
- Reflects the error to the virtual machine if it is the result of a SIO issued by a virtual machine. The manner of reflection depends on the processor and channel models; in addition to the CSW, the limited channel logout (LCL) and extended channel logout are reflected as appropriate, depending upon the model. If the setting of the virtual machine's control register 14 masks out the extended channel logout, the extended channel logout data is not kept pending and is lost to the virtual machine, but still is recorded in the VM/370 error recording cylinders. Figures 26 and 27 show, in greater detail, under what circumstances the various channel checks are reflected to the virtual machine.

	Non-Dedicated Channel	Dedicated Channel
CP I/O	CP attempts recovery	CP attempts recovery
Virtual Machine SIO I/O	Reflected to virtual machine	Reflected to virtual machine
Virtual machine DIAGNOSE I/O	CP attempts recovery	CP attempts recovery
Unsolicited Interrupt	CP attempts recovery	Reflected to virtual machine

Figure 26. Handling of Channel Check, Channel Control Check, and Interface Control Check

	Non-Dedicated Channel	Dedicated Channel
CP I/O	VM/370 wait state	VM/370 wait state
Virtual machine SIO I/O	VM/370 wait state	Reflected to virtual machine
Virtual Machine DIAGNOSE I/O	VM/370 wait state	VM/370 wait state
Unsolicited interrupt	VM/370 wait state	Reflected to virtual machine

Figure 27. Handling of Interface Inoperative

CHANNEL CHECK HANDLER--INITIALIZATION

To be effective, CCH must be tailored to the resident system operating environment. This is done during the CP initialization phase by the use of the Store Channel ID instruction (STIDC) and the Store Processor ID instruction (STIDP).

By using the STIDP instruction, it can be determined whether the processor is a 165 II, or 168 or some other VM/370-supported system. If it is a 165 II or 168, then a determination must be made to find out what type of standalone channels are attached to the system. This is done by using the STIDC instruction. When the type of channels is determined, the related standalone channel program modules are loaded and locked into main storage. If the system is not a 165 II or 168, support for the integrated channels is provided.

Besides determining the processor and channel types, CP initialization does the following:

- Obtains storage for maximum I/O extended logout area for the VM/370-supported system.
- Initializes logout and ECSW to all ones.
- Sets up the I/O extended lcgout pointer, if one exists for the supported system.

It is only after this initialization that CCH can assist the system in its error recovery function.

CHANNEL CHECK HANDLER--SUMMARY

CCH receives control from the I/O supervisor when a channel check occurs. CCH produces an I/O error block (IOERBLOK) for the error recovery procedure and a record to be written in the error recording area for the system operator or customer engineer. The VM/370 system's operator or customer engineer may obtain a copy of the record by using the CMS CPEREP command. A message about the channel error is issued to the system's operator each time a record is written in the error recording area.

When the input/output supervisor program detects a channel error during routine status examination (following the issuance of an I/O instruction or following an I/O interruption), it passes control to the channel check handler. If the error is a channel control check or interface control check, CCH analyzes the channel logout information and constructs an IOERBLOK, and, if the error is not a channel data check, an ECSW is constructed and placed in the IOERBLOK. The IOERBLOK provides information for the device-dependent error recovery procedures. CCH also constructs a record to be recorded in the error recording area. Normally, CCH returns control to the I/O supervisor after constructing an IOERBLOK and a record. However, if CCH determines that system integrity has been damaged (system reset or invalid unit address), then system operation is terminated. For system termination, CCH issues a message directly to the system operator and places the processor in a disabled wait state with a recognizable wait code in the processor instruction counter.

Normally, when CCH returns control to the I/O supervisor, the error recovery procedure is scheduled for the device that experienced the error. When the ERP receives control, it prepares to retry the operation if analysis of the IOERBLOK indicates that retry is possible. Depending on the device type and error condition, the ERP either recovers or marks the event fatal and returns control to the I/O supervisor. The I/O supervisor calls the recording routine to record the channel error. The primary system operator is notified of the failure, and the recording routine returns control to the system and normal processing continues.

If the channel check is associated with an I/O event initiated by a SIO in a virtual machine, the logout is reflected to the virtual machine in one of two ways, depending on whether the channel check occurred at SIO time, or later in an interrupt. If it occurred at SIO time, the SIO routine calls CCH to reflect the logout. If it occurred in an I/O interrupt, the dispatcher notices the channel check as it is reflecting the I/O interrupt to the virtual machine, and at that time the dispatcher calls CCH to reflect the channel logout.

VM/370 Channel Check Handler action is summarized in Figure 28.

Channel Address Valid	Retry Codes Valid	Channel Has Been Reset	Start I/O Time	Unit Address Valid	Action Code[1]
No					2
Yes	No				2
Yes	Yes	Yes			2
Yes	Yes	No	Yes		1,3
Yes	Yes	No	No	No	1
Yes	Yes	No	No	Yes	1,3

[1]Action Codes:
1 = Schedule recording.
2 = Schedule system termination with proper message (the error will be recorded if SEREP is invoked).
3 = Error can be isolated to a device for retry.

Figure 28. Channel Check Action Table

All messages that are the result of the channel check handler are prefixed by the designation DMKCCH and are described in the publication *VM/370 System Messages*. Action by the channel check handler can also force the system into wait state 002. Operator action for the wait state condition is also described in *VM/370 System Messages*.

Other Error Messages and Wait State Codes

There are three critical phases of VM/370 CP operations where continuous system operation is vulnerable and may degenerate to wait state codes as a result of machine check or fatal I/O error conditions. They are: during VM/370 CP initialization, during system checkpoint activity and during the occurrence of system dump operations.

The resultant messages and wait state codes are produced by other system modules (other than DMKCCH and DMKMCH). Consult *VM/370 System Messages* for a description of these messages and wait state codes.

FIXED STORAGE ASSIGNMENT AND LOGOUT AREAS

The storage areas that concern CCH and MCH for error analysis are:

- Permanent storage assignments
- I/O communications areas
- Fixed logout area
- Extended logout area

Figure 29 shows details of these areas.

	logs out at					
channel	fixed location	location pointed to by location	length of logout in bytes	CSW at	LCL (ECSW) at	unit address at
2860	304		24	64		
2870	304		24	64		
2880		172	112	64		
145/148 145-3		172	96 maximum	64	176	186
135/138 135-3	256		24 maximum	64	176	186
155/158	155 & 158 channels do not log out			64	176	186
4331 4341	No fixed or I/O extended logout areas			64	176	186
3031 3032 3033		172	640	64	176	186

NOTE: All numbers in this figure are decimal. 3031, 3032 and 3033 have integrated channels. The 2880, 2870 and 2860 channels cannot be attached to these processors. Their channels are similar to M145 channels in that both a LCL and an IOEL are produced.

Figure 29. VM/370 Fixed Storage and Logout Areas

Section 4. Additional CE Aids

- CPEREP and OS/VS EREP
- SET RECORD and SET MODE Facility
- TRACE Facility
- VMFDUMP

VM/370 CPEREP and OS/VS EREP

CPEREP reads system error records from the VM/370 error recording area and produces printed reports. CPEREP can also be used to copy the error records to tape and to clear the VM/370 error recording area.

The OS/VS Environmental Recording, Editing, and Printing Program (EREP) is executed when the CMS CPEREP command is invoked. CPEREP provides the virtual machine user with all the facilities of OS/VS EREP. The reports generated by CPEREP have the same format as those generated on an OS/VS system. The content of the reports depends upon the specified (or defaulted) CPEREP operands and upon the input system error records. The input system error records may be from the VM/370 error recording area or from a history tape. The history tape may have been produced earlier by CPEREP from VM/370 error recording area data or by an OS/VS system from SYS1.LOGREC data. Unlabelled tapes produced on OS/VS systems by OS/VS EREP and on VM/370 systems by CPEREP are compatible and can be transported between systems. Data from multiple systems can even be accumulated on the same tape.

OS/VS EREP is documented in existing OS/VS publications, but the VM/370 CPEREP command is not described there. Therefore, the function of this chapter is to:

- Describe briefly the capabilities of OS/VS EREP
- Describe in detail how the CPEREP command is invoked
- Describe the CPEREP interface to OS/VS EREP
- Refer to the OS/VS publications for details on operands and the reports OS/VS EREP produces.

Publications

Because OS/VS EREP is not a program exclusively used with VM/370 and because it is not part of the VM/370 system reference library, the user must use the latest OS/VS1 or OS/VS2 publication that describes EREP. Changes and enhancements to EREP documented in the OS/VS Publications, because of the level of the detailed information and the fact that OS/VS1, OS/VS2 and VM/370 do not adhere to the same publication print cycle, may not be documented in the VM/370 publications.

The VM/370 user requires the following publications:

VM/370 OLTSEP and Error Recording Guide, Order No. GC20-1809

OS/VS, DOS/VSE, VM/370 Environmental Recording Editing and Printing (EREP) Program, Order No. GC28-0772

The above publications give details on operands used by CPEREP and describes the outputs.

VM/370 System Messages, Order No. GC20-1808

The user should be aware that error messages may originate from the CPEREP program with a *DMS* prefix or they can originate from OS/VS EREP with an *IFC* prefix. In addition to its *DMS* section for CMS, *VM/370 System Messages* contains an *IFC* message section devoted to EREP. This section does not describe all possible EREP messages, but it does describe all EREP messages that are likely to be issued in the VM/370 environment.

The VM/370 user, in rare instances, may require the following publication which describes the full set of possible EREP messages:

<u>OS/VS, DOS/VSE, VM/370 EREP Messages, Order No. GC38-1045</u>

Logic information pertaining to CPEREP and OS/VS EREP is contained in the following publications:

<u>VM/370 Service Routines Program Logic</u>, Order No. SY20-0882

This describes the CPEREP modules, DMSIFC and DMSREA.

<u>OS/VS, DOS/VSE, VM/370 Environmental Recording Editing and Printing (EREP) Program Logic</u>, Order No. SY28-0773

If error records from a 3850 Mass Storage System (MSS) are to be processed, then one or both of the publications listed below are required. These publications describe the use of the VS1/VS2 Subsystem Data Analyzer (SDA) program (ISDASDAO). This program runs under VS1 or VS2 operating systems and is used with SYS1.LOGREC data set information for the generation of analysis and performance reports for MSS hardware.

<u>OS/VS1 SYS1.LOGREC Error Recording</u>, Order No. GC28-0668

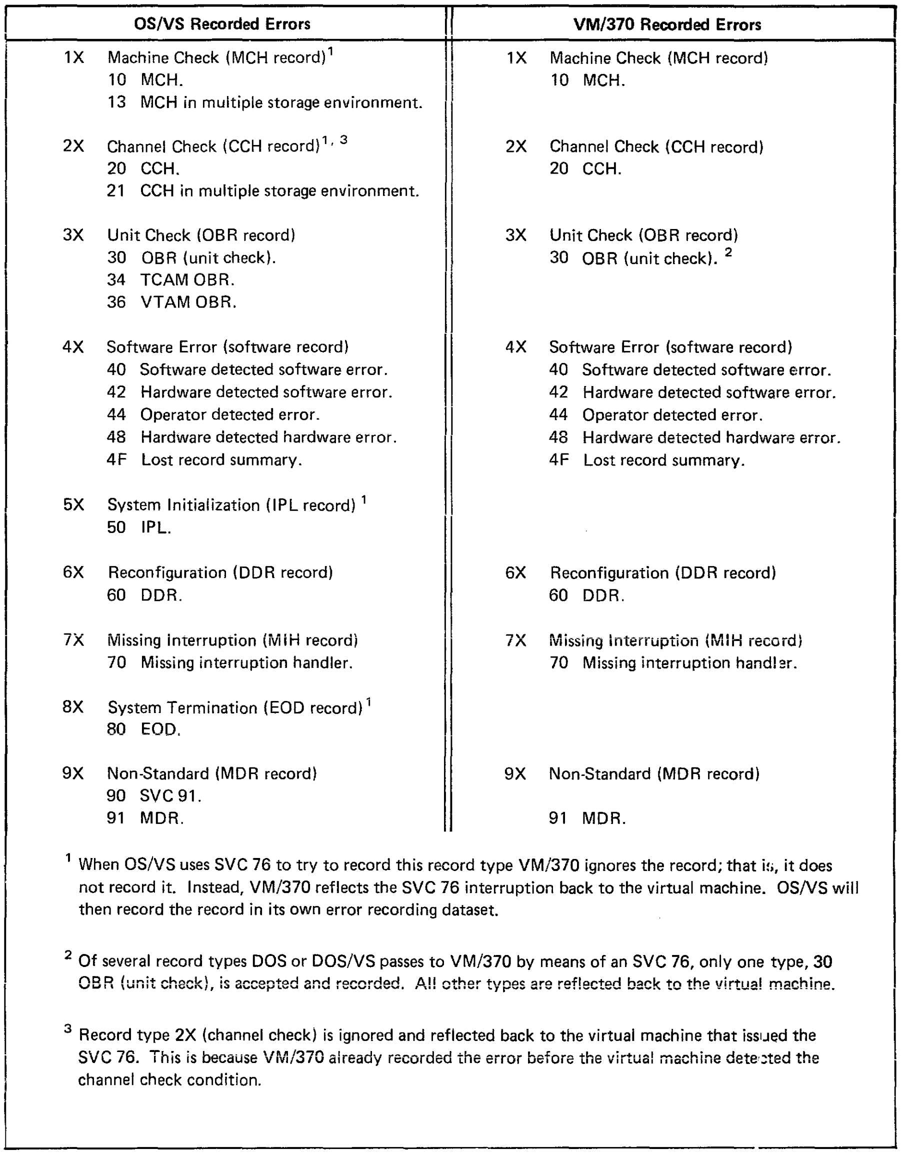
<u>OS/VS2 System Programming Library: SYS1.LOGREC Error Recording</u>, Order No. GC28-0677

CPEREP and OS/VS EREP -- An Overview

In the VM/370 system, the CMS CPEREP command provides access to the OS/VS EREP program. Operands are supplied to CPEREP via a control file and/or prompted console input. The CMS module DMSIFC called by the CPEREP command provides an initial screening and edit of the operands that are passed to IFCEREP1 (a module of OS/VS EREP) for the edit and printing phase of EREP. IFCEREP1 does the final screening of the supplied operands and initiates error record retrieval activity and the requested edit and print function. Because the format of the VM/370 error recording area differs from the format of a SYS1.LOGREC data set, the method of error record retrieval and erasure from DASD differs. To circumvent format incompatibilies, DMSIFC causes EREP's I/O operations to the OS/VS SYS1.LOGREC data set to be trapped and simulated. DMSIFC performs the simulation and in the process it calls on DMSREA to read records from the VM/370 error recording area. For other files required by EREP, DMSIFC does not perform the I/O simulation; it merely issues FILEDEFs for them. For these files the standard simulation of OS files provided by CMS is adequate.

The formats of the individual records in the OS/VS SYS1.LOGREC data set and the VM/370 error recording area are identical; however, VM/370, through the medium of SVC 76, does not record on its error recording area all error record types. On the VM/370 system, errors passed to VM/370 for error recording (via SVC 76) that do not adhere to VM/370 standards are reflected to the virtual machine to record the error on its own error recording data set.

The error record types recorded by VM/370 as opposed to the record types recorded by OS/VS1 and OS/VS2 operating systems are shown in Figure 30. Although the process of recording errors has been summarized here, it should be noted that CPEREP and OS/VS EREP are only involved in reading the error records for reporting purposes and are not involved in writing the error records at the time of their occurrence to the recording medium be it SYS1.LOGREC data set or VM/370's error recording area.



OS/VS Recorded Errors	VM/370 Recorded Errors
1X Machine Check (MCH record)[1] 10 MCH. 13 MCH in multiple storage environment.	1X Machine Check (MCH record) 10 MCH.
2X Channel Check (CCH record)[1, 3] 20 CCH. 21 CCH in multiple storage environment.	2X Channel Check (CCH record) 20 CCH.
3X Unit Check (OBR record) 30 OBR (unit check). 34 TCAM OBR. 36 VTAM OBR.	3X Unit Check (OBR record) 30 OBR (unit check).[2]
4X Software Error (software record) 40 Software detected software error. 42 Hardware detected software error. 44 Operator detected error. 48 Hardware detected hardware error. 4F Lost record summary.	4X Software Error (software record) 40 Software detected software error. 42 Hardware detected software error. 44 Operator detected error. 48 Hardware detected hardware error. 4F Lost record summary.
5X System Initialization (IPL record)[1] 50 IPL.	
6X Reconfiguration (DDR record) 60 DDR.	6X Reconfiguration (DDR record) 60 DDR.
7X Missing Interruption (MIH record) 70 Missing interruption handler.	7X Missing Interruption (MIH record) 70 Missing interruption handler.
8X System Termination (EOD record)[1] 80 EOD.	
9X Non-Standard (MDR record) 90 SVC 91. 91 MDR.	9X Non-Standard (MDR record) 91 MDR.

[1] When OS/VS uses SVC 76 to try to record this record type VM/370 ignores the record; that is, it does not record it. Instead, VM/370 reflects the SVC 76 interruption back to the virtual machine. OS/VS will then record the record in its own error recording dataset.

[2] Of several record types DOS or DOS/VS passes to VM/370 by means of an SVC 76, only one type, 30 OBR (unit check), is accepted and recorded. All other types are reflected back to the virtual machine.

[3] Record type 2X (channel check) is ignored and reflected back to the virtual machine that issued the SVC 76. This is because VM/370 already recorded the error before the virtual machine detected the channel check condition.

Figure 30. VM/370 vs OS/VS1 and OS/VS2 Error Record Types Recorded

Figure 30. VM/370 vs OS/VS1 and OS/VS2 Error Record Types Recorded

The OS/VS emergency offload program (IFCOFFLD) of EREP modules is not supported by VM/370. Function performed by IFCOFFLD (that is, the program that is used to quickly dump a SYS1.LOGREC error recording data set to a history tape during an emergency when time and circumstance do not permit an EREP to printer run) can be performed by CPEREP in conjunction with a user created control file (details on how this is done are described further on in the text).

Using CPEREP and the Facilities of OS/VS EREP

To use the CPEREP command, the user must invoke the command from the CMS environment; the user must have privilege class C, E, or F to access records in the VM/370 error recording area. To erase the VM/370 error recording area, the user must have privilege class F.

CPEREP as depicted in this text is consistent with the same notational conventions as used in the VM/370 publications to describe other commands. Briefly, operands in brackets ([ ]) indicate one operand may be selected; for operands nested in braces ({ }) one operand must be selected. Operands that are underscored (_) are default values. Parentheses, commas, periods, and equal signs must be entered as shown. All lowercase operands indicate variable values that are to be entered. Uppercase values indicate the keywords that must be entered for the functions that are to be performed.

For a full description of VM/370 notational conventions, refer to the *VM/370 CP Command Reference for General Users*.

The format of the CPEREP command with available operands is shown in Figure 31. "CPEREP Command Format". Syntactical rules for specifying operands are given under the following topic: "Entering CPEREP and EREP Operands". A brief description of the keywords used with CPEREP is contained in Figure 32. No attempt is made in this publication to describe the precise meaning and use of operand values that are required with the user supplied keywords. Also (except for Figure 33, which shows the reports that can be generated and the operands allowed with each report), no attempt is made to explain the relationship of keywords to one another. Some keywords are used with each other and at the same time disallow the use of other keywords. For details on OS/VS operands and the unique operand to operand relationships, refer to the OS/VS EREP publication. Details on the operands (CLEAR, CLEARF, and TERMINAL) that are unique to VM/370 are covered only in the present publication, not in the OS/VS publication.

```
┌─────────┬──────────────────────────────────────────────────────────┐
│         │                                                          │
│CPEREP   │ [filename filetype [filemode]]                           │
├─────────┴──────────────────────────────────────────────────────────┤
│The operands that follow cannot be entered on the command line.     │
│Enter operands via prompting technique or from file specified as    │
│above.                                                              │
├─────────┬──────────────────────────────────────────────────────────┤
│         │ ┌   ┌    ┐┐                                              │
│ │       │ │ACC│={Y}││                                              │
│         │ │   │ {N}││                                              │
│         │ └   └    ┘┘                                              │
│         │ ┌         ┐1                                             │
│         │ │CLEAR    │                                              │
│         │ │CLEARF   │                                              │
│         │ └         ┘                                              │
│         │ [CPU=serialno.modelno[,serialno.modelno,...]]            │
│         │ [CPUCUA=(serial.addr,serial.addr[,serial.addr,...])]     │
│         │ ┌                                          ┐2            │
│         │ │CTLCRD{date1[date2[interval[title...]]]}│               │
│         │ │      {date1,[date2],[interval],[title }│               │
│         │ └                                        ┘               │
│         │ [CUA=(addr[,addr,...])]                                  │
│         │ [DATE=(yrday[,yrday])]                                   │
│         │ [DEV=(devtype[,devtype,...])]                            │
│         │ [DEVSER=(serial[,serial,...])]                           │
│         │ [ERRORID=(seqno[,cpuid,asid,hh,mm,ss,t])]                │
│         │ ┌     ┌    ┐┐                                            │
│         │ │EVENT│={Y}││                                            │
│         │ │     │ {N}││                                            │
│         │ └     └    ┘┘                                            │
│         │ ┌    ┌    ┐┐                                             │
│         │ │HIST│={N}││                                             │
│         │ │    │ {Y}││                                             │
│         │ └    └    ┘┘                                             │
│         │ [LIBADR=addr]                                            │
│         │ [LINECT=nnn]                                             │
│         │ ┌     ┌    ┐┐                                            │
│         │ │MERGE│={Y}││                                            │
│         │ │     │ {N}││                                            │
│         │ └     └    ┘┘                                            │
├─────────┴──────────────────────────────────────────────────────────┤
│1Operand exclusive to VM/370.                                       │
│2After entering the CTLCRD operand and associated data, no further  │
│ operand can be entered on the same command line; the next operand  │
│ must begin on a new line.                                          │
└────────────────────────────────────────────────────────────────────┘
```

Figure 31. CPEREP Command Format (Part 1 of 2)

```
|CPEREP  | ┌    ┌     ┐┐                                          |
|(cont.) | |MES |=(N) ||                                          |
|        | |    | {Y}||                                          |
|        | └    └     ┘┘                                          |
|        | [MOD=(modelno[,modelnc...])]                           |
|        | ┌             ┐                                        |
|        | |PRINT=(PS) |                                         |
|        | |       {PT} |                                         |
|        | |       {SU} |                                         |
|        | |       (NO) |                                         |
|        | └             ┘                                        |
|        | ┌       ┌     ┐┐                                       |
|        | |RDESUM|=(Y) ||                                       |
|        | |       | {N}||                                       |
|        | └       └     ┘┘                                       |
|        | ┌                                            ┐          |
|        | |SHARE=          (cua)          (cua)       |          |
|        | |      (serial.{cuX},serial.{cuX},...)     |          |
|        | └                                            ┘          |
|        | ┌      ┌     ┐┐                                        |
|        | |SHORT|=(Y) ||                                        |
|        | |      | {N}||                                        |
|        | └      └     ┘┘                                        |
|        | ┌                ┐                                     |
|        | |SYMCDE=(nnnn) |                                      |
|        | |        {nnnX} |                                      |
|        | |        {nnXX} |                                      |
|        | |        (nXXX) |                                      |
|        | └                ┘                                     |
|        | ┌      ┌     ┐┐                                        |
|        | |SYSUM|=(Y) ||                                        |
|        | |      | {N}||                                        |
|        | └      └     ┘┘                                        |
|        | [TABSIZE=sizeK]                                        |
|        | ┌         ┌     ┐┐¹                                     |
|        | |TERMINAL|=(Y) ||                                      |
|        | |         | {N}||                                      |
|        | └         └     ┘┘                                     |
|        | [TERMN=termname]                                       |
|        | [THRESHOLD=(tempread,tempwrite)]                       |
|        | [TIME=(hhmm,hhmm)]                                     |
|        | ┌       ┌     ┐┐                                       |
|        | |TRENDS|=(Y) ||                                       |
|        | |       | {N}||                                       |
|        | └       └     ┘┘                                       |
|        | [TYPE=[C][D][E][H][I][M][O][S][T]]                     |
|        | [VOLID=(volid1[,volid2[,volid3[,volid4]]])]            |
|        | ┌     ┌     ┐┐                                         |
|        | |ZERO|=(Y) ||                                         |
|        | |     | {N}||                                         |
|        | └     └     ┘┘                                         |
|-----------------------------------------------------------------|
|¹Operand exclusive to VM/370                                     |
```

Figure 31. CPEREP Command Format (Part 2 of 2)

Operand	Description
ACC=	Indicates that selected error records are to be accumulated in an output data set. The particular error records selected and the source of these records (either the VM/370 error recording cylinders, or a history file, or both) depends on what other operands are coded. The output accumulation data set is normally a tape mounted on tape drive 181, but this can be changed (see the section "CPEREP FILEDEFS"). When output is accumulated on tape 181, the output is added as an extension of the existing file: the tape is rewound and then spaced out to the end of the first file prior to writing. Therefore, if a tape is to be used for the first time, the user should write a tape mark at the beginning of the tape before invoking CPEREP (the CMS TAPE command can do this). When output is accumulated on a tape, the tape should be mounted, readied, and attached to the user's virtual machine as tape 181 prior to invoking CPEREP. Note that for most types of reports, ACC=Y is the default.
CLEAR CLEARF	CLEAR clears all error records from the error recording area, but does not clear the SRF frame records. CLEARF clears the SRF frame records from the error recording area, as well as error records subsequent to re-reading the SRF frames, and writing the frame records at the beginning of the error recording area. A CLEAR or CLEARF operand cannot be invoked with other operands. It must be invoked in a standalone manner. Therefore, the user should capture pertinent error area information before invoking one of these operands. It is recommended that the user acquaint himself with the ZERO operand for erasing the VM/370 error recording area. The ZERO operand is used in conjunction with report generation operands and does not execute the erase process until the report generation process is complete. The service support console must be in SRF mode. Note: The CLEARF operand is designed for the 3031, 3032, and 3033 processors. CPEREP should be invoked with CLEARF specified after the installation of engineering changes to the processor and channels. To access the SRF (1) enable the I/O interface for the service support console, (2) activate the C1 frame, (3) select SRF mode (A2) for 3031 and 3033 processors (SRF appears disabled until accessed on the 3032), (4) vary on SRF, and (5) attach the SRF to the console of the class F user running CPEREP. CLEARF clears error records on a 158 or 168 processor.
CPU	An error record selection operand.--It allows the selection of error records by the central processor unit's serial and model number. Multiple processor values may be specified as multiple sub-operands of CPU.
CPUCUA	An error record selection operand.--It allows the selection of error records that relate to a specific processor (serial address) and an attached device (cuu) address. Multiple processor and devices can be specified as multiple sub-operands of CPUCUA.

Figure 32. Operands Used with CPEREP (Part 1 of 5)

Operand	Description
CTLCRD	An error record selection operand.--When the RDESUM operand requests an IPL report, CTLCRD controls the selection of error records via its span of dates and IPL clustering interval. Note: This operand and the date1, date2, interval, and title operands associated with it must be completed on one line of input and must not be followed by any other operands on this one line of input.
CUA=	An error record selection operand.--It allows the selection of error records by specific device address a range of device addresses, all the devices on a particular control unit, and all the devices on a particular channel. Multiple address values or ranges of values can be specified as multiple sub-operands of CUA.
DATE=	An error record selection operand.--It allows the selection of error records by the date or span of dates (Julian day values) specified.
DEV=	An error record selection operand.--It allows the selection of error records by device type (for example, 2314, 3330). Multiple device types can be specified as multiple suboperands of DEV.
DEVSER=	An error record selection operand.--It allows the selection of error records by the specific device serial number in the service data field in the error record. This operand is valid for only 3410/3420 devices. Multiple device serial numbers can be specified as multiple suboperands of DEVSER.
ERRORID=	An error record selection operand.--It applies only to MCH and software records generated by OS/VS2 MVS. It allows selection by the five digit error identifier alone or by the five digit error identifier, processor identifier, address space identifier, and date/time values.
EVENT=	A report generation operand. -- This operand generates one line abstracts of all or selected error records in chronological order.
HIST=	Indicates that the source of the error records for this run is to be a history data set rather than the VM/370 error recording cylinders. A history data set is a data set that was created as an accumulation (ACC) data set during an earlier session. Usually, the history data set is a tape mounted on tape drive 182, but this can be changed (see the section "CPEREP FILEDEFs"). When input is from a history tape, the tape should be mounted, ready, and attached to the user's virtual machine as tape 182 prior to invoking CPEREP.

Figure 32. Operands Used with CPEREP (Part 2 of 5)

Operand	Description
LIBADR=	An error record selection operand.--It allows the selection of error records by the four-digit hexadecimal line interface base address.
LINECT=	An output report formatting operand.--This operand defines the number of lines of error data that are to be printed on a page.
MERGE=	Indicates that the source of the error records for this run is to be both a history data set and the VM/370 error recording cylinders. The history data set is as described earlier for the HIST operand.
MES=	A report generation operand.--This operand allows the generation of a Media Error Statistics Report. This operand is valid for processing 3410 and 3420 magnetic tape subsystem records.
MOD=	An error record selection operand.--It allows the selection of error records by specified processor model designation; for example, 158 or 3062. Multiple processor model numbers may be specified as suboperands of MOD.
PRINT=	A report formatting operand.--Values supplied with this operand in conjunction with other operands produce: SU - A printed summarization of all selected error records. PS - A printed summarization as well as the full printout of all selected errors. PT - Full printing of all selected error records only. NO - No printed output.
RDESUM=	A report generation operand -- This operand allows the generation of the IPL Report produced by the RDE Summary. The IPL report contains each IPL in the sequence of occurrence with the date, time, and reason for the IPL and the subsystem, if any, that was responsible for the IPL action. Note: With VM/370 error recording area data there will be nothing to report as VM/370 does not record IPL records to its error areas. However, RDESUM may be invoked when processing history tapes generated by another operating system which include the RDE option.

Figure 32. Operands Used with CPEREP (Part 3 of 5)

Operand	Description
SHARE=	In an installation where several processors share a device or where channel switches (or similar features) provide multiple paths to a device, this operand identifies all cf the equivalent addresses of a particular device or contrcl unit. From 2 to 6 equivalent addresses may be specified as multiple sub-operands of SHARE. The SHARE operand may be specified more than once and is generally used once fcr each shared device or control unit. When a device is shared by more than one processor, I/O errors are recorded in the error log of the processor in control at the time of the error. If the error logs cf several processors are accumulated on a single tape, a SHARE operand for the shared device allows EREP to bring together and report all of the errors for the device regardless of where they were recorded.
SHORT=	An error record selection operand.--When specified in conjunction with the PRINT operand, this operand suppresses the printing of short OBR (outboard recordings) records.
SYMCDE=	An error record selection operand that allows the selecticn of recorded error records whose sense tyte bits match supplied values. This operand is valid for DASD 33xx devices.
SYSUM=	A report generation operand.--Selection of this operand produces a System Summary Report. This report is a comprehensive condensed report on the principal elements cf a system, that is, the processor, channels, storage, I/C, and the system control program.
TABSIZE=	An EREP processing operand. The value supplied with this operand defines the size of the sort table to be used in processing error records. Note: If a value substantially greater than 24K is specified, it may be necessary to increase the storage size of the virtual machine in order to execute EREP programs.
TERMINAL=	A CPEREP operand that is not passed on to EREP.--When CPEREP encounters this operand while reading operands from a file of operands, it causes CPEREP to discontinue reading records from the file and to begin prompting for operands at the terminal instead. All operands on the current record are processed before going to the terminal, but subsequent records in the file, if any, are ignored. If CPEREP encounters this operand while reading from the terminal, it is ignored; control cannot be switched back to the disk file.
TERMN=	An error record selection operand.--Values supplied with this operand are the 1 to 8-character alphameric names applied to terminals as used via VTAM and TCAM operations.

Figure 32. Operands Used with CPEREP (Part 4 of 5)

Operand	Description
THRESHOLD=	A report generaticn operand -- This operand, used with 3410 and 3420 series tape systems, allows the user to set THRESHOLD values for the number of temporary read and temporary write errors that occur. With these values established, only those devices that exceed the THRESHOLD values are printed.
TIME=	An error record selection operand.--Supplied values allcw the user to select a time span for the processing of errcr records that occurred within that time span.
TRENDS=	A report generaticn operand.--The IREND Report provides a summarizaticn of error ccunts per day cn the component groups that comprise the system installation.
TYPE=	An error record selection operand.--Values supplied with this operand allow the user to select the record types to te processed. Valid record types are: channel check records (code C), dynamic device reconfiguration records (code D), end-of-day records (code E), missing interrupt handler records (code H), initial program load records (code I), machine check records (code M), outboard (I/C) records (code O), program error records (code S), and miscellaneous data records (code T). If no record type is specified, all record types are processed.
VOLID=	An error record selection operand.--Allows error record selection by defined volume serial number values. This cperand is valid for 34xx and 33xx subsystems.
ZERO=	This operand is invoked to erase the error records from the VM/370 error recording area. The error area is erased after all other functions/operations of EREP have been successfully written out to the printer or to an accumulaticn tape.

Figure 32. Operands Used with CPEREP (Part 5 of 5)

EREP Reports			Print							
Keyword	Event	MES	=PS	=PT	=SU	=NO	RDESUM	SYSUM	Trend	Threshold
ACC	X		X	X	X	X	X	X		
CLEAR										
CLEARF										
CPU		X	X	X	X	X				X
CPUCUA			X	X	X	X				
CTLCRD							X			
CUA	X	X	X	X	X	X			X	X
DATE	X	X	X	X	X	X		X	X	X
DEV	X	X[1]	X	X	X	X		X	X[1]	
DEVSER		X								X
ERRORID			X	X	X	X				
HIST	X	X	X	X	X	X		X	X	X
LIBADR			X	X	X	X				
LINECT	X	X	X	X	X		X	X	X	X
MERGE	X	X	X	X	X	X	X	X	X	X
MOD			X	X	X	X				
SHORT			X	X	X	X				
SYMCDE			X	X	X	X				
TABSIZE	X	X	X	X	X	X		X	X	X
TERMINAL	X	X	X	X	X	X	X	X	X	X
TERMN			X	X	X	X				
TIME	X	X	X	X	X	X		X	X	X
TYPE	X		X	X	X	X			X	
VOLID		X	X	X	X	X				X
ZERO			X[2]	X[2]		X[2]		X[2]		
PRINT=PT		X								
PRINT=PS		X								
PRINT=SU		X								

[2]The 3410 and/or 3420 devices are permitted.
[1]The ZERO keyword is acceptable only when no selective operands are requested and all records are either printed or accumulated on tape.

Note: During an execution of EREP one of the above PRINT functions is performed. The default function is PRINT=SU, which is underscored and generates summary reports for all error recorded data; this is the defaulted input.

Figure 33. Types of Reports, Showing Operands Allowed with Each

Entering CPEREP and EREP Operands

As mentioned previously, the class C, E, or F user must be in the CMS environment to invoke CPEREP. EREP operands can be supplied to CPEREP (CMSIFC) by means of a console prompting technique or from a previously generated file that contains the operands required for the desired EREP record output.

The sequence for invoking CPEREP is as follows:

1. Log on the CE's virtual machine.
2. IPL CMS.
3. Have the system operator attach any required tape devices to the virtual machine to serve for input and/or output data set use. (See description of the HIST and ACC operands).
4. Enter CPEREP and EREP operands via the file entry method or by the prompting method.

   Note: The typical method of entering commands and operands on the same input line, as is done by other CP and CMS commands, is not valid. The reason that such action is disallowed is that many functions of EREP can exceed the maximum input line length allowed by VM/370.

PROMPTING METHOD

The CPEREP command is typed on the terminal followed by pressing the ENTER key. After a short pause, the system prompts the user with:

```
ENTER:
```

The user then enters CPEREP operands. If the user's needs exceed one line of input (limited by terminal line length), the user types a few operands and then presses the ENTER key again. The system then responds with the ENTER: prompt message again. The user may then enter more operands. The process is repeated until no more operands are required. When this occurs, the user presses the ENTER key to signal with a null line the end of the current string of operands.

In entering operands, the following rules must be observed:

- Keywords and their associated values must be separated from a following keyword operand by a blank (space), or multiple blanks, or by a comma.
- Embedded commas, periods, and parentheses that define the extent of variable operands must be entered as indicated in the CPEREP command format structure previously described.
- Keywords and keywords with their related variable operand(s) may be entered on the command input lines in any order.
- When specifying a keyword where the allowed values are Y and N, the =Y may optionally be omitted, with the keyword alone being specified. This form of the operand will always be interpreted as a Y specification regardless of the normal default value.
- To initiate CPEREP with system default values, respond to the first ENTER: prompt message by pressing the ENTER key (enter a null line).

A sample illustrating the previous points is described later in the text.

FILE ENTRY METHOD

The CPEREP command is entered followed by the filename, filetype, and filemode identity of a file that contains a "package" of CPEREP operands arranged in the format as described in the prompt method. The same rules regarding blanks, commas, and parentheses still apply. In addition, card images are truncated at column 71.

In practice, a VM/370 installation would probably have multiple files containing various operand "mixes" to satisfy the installation CPEREP report needs. To create and generate the necessary CPEREP files for this method of entry, use the CMS EDIT command. File generation using the CMS EDIT command is described in the _VM/370 CMS User's Guide_.

MIXED METHOD

The CPEREP command is entered followed by the filename, filetype, and filemode identity of a file of CPEREP operands, one of which is the TERMINAL operand. The operands are read from this file until the TERMINAL operand is encountered. At this point no further input is read from the file. Prompting begins at the terminal where additional operands may be entered.

LOGON TO CPEREP EXECUTION--AN EXAMPLE

The following example shows a complete CPEREP operation as initiated from the virtual machine console from the logon step to CPEREP completion. Lowercase letters indicate user entries, uppercase letters indicate VM/370 system response. The lozenge indicates an ENTER key action.

```
Console Listing                   Comments

logon ce □                        Logon initiated

ENTER PASSWORD:

        □

LOGMSG-10:14:15 04/13/76          Logon process completed
*RUNNING IPL5
LOGON AT 10:54:13
THURSDAY 04/13/76

msg operator attach tape as 181 □ User requesting tape for CPEREP use.

TAPE 181 ATTACHED

                                  Note that if HIST and ACC functions
                                  are to be invoked, two tape drives
                                  must be attached before invoking
                                  CPEREP.

ipl cms □
CMS VERSION 4.1-04/30/76          Loads CMS.  CPEREP can only be
                                  invoked after the CMS environment is
                                  entered.
```

```
cperep □                           CPEREP is invoked.
                                   Note: CPEREP was invoked with nc
                                   control file operand; therefore, the
                                   operation defaults to the prompting
                                   technique.

GLOBAL TXTLIB ERPTFLIB EREPLIB     The system response indicates that an
                                   EXEC has been executed and now EREP
                                   library members are available to
                                   process CPEREP requests.

CPEREP                             This system response indicates that
                                   CPEREP initialization is in progress.

ENTER:                             Message from CPEREP prompting fcr
                                   operand input.

print=ps dev=(3340) □              First line of CPEREP operand entry
                                   followed by pressing the ENTER key.

ENTER:                             CPEREP prompts for more operand
                                   input.

□                                  Operand entry has been completed so
                                   the ENTER key is pressed again. This
                                   creates a null line. The null line
                                   indicates to the system that the EREP
                                   execution phase is to begin.

DATE - 116 77                      EREP INFORMATIONAL MESSAGES

INPUT PARAMETER STRING             PRINT = PS, DEV=(3340)

PARAMETER OPTIONS VALID FOR THIS EXECUTION
  RECORD TYPES(MCH,CCH,ORD,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)-3340(200A)
  DEVICE TYPES(MDR)-3340(09)
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

                                   The above represents information frcm
                                   OS/VS EREP to the user indicating
                                   operand selection and OS/VS default
                                   values. Also indicated is a synopsis
                                   of the EREP exercise.
```

Though the previous example does not show a complex string of OS/VS EREP and CPEREP operands being used, it does show that the procedure itself is not difficult to use. Indicated in the example are informaticn messages issued by the OS/VS EREP. CMS messages as well as other OS/VS EREP messages may also appear in the course of EREP execution. Consult the VM/370 System Messages for the meaning of messages prefixed by DMSIFC, DMSREA, and IFCxxxx.

Further insight into the functions performed by DMSIFC is realized when you examine a typical example of an OS/VS EREP entry from a standalone OS/VS1 system as shown below.

```
//EVENT      JOB
//STEP1      EXEC     PGM=IFCEREP1,PARM=('EVENT,DATE=(76130,76150),HIST',
//  'ACC=N')
//ACCIN      DD       DSN=EREP.HIST,DISP=OLD
//DIRECTWK   DD       UNIT=SYSDA,SPACE=(CYL,(5))
//TOURIST    DD       SYSOUT=A,DCB=BLKSIZE=133
//EREPPT     DD       SYSOUT=A,DCB=BLKSIZE=133
//SYSIN      DD       DUMMY
//
```

The EXEC job control statement, indicated by the arrow, shows a selection of OS/VS operands; it also shows the comma that is used as an operand separator. Job control statements are needed to define those data sets that are required for the OS/VS EREP execution process. In VM/370, CPEREP provides the data set requirements automatically. For the VM/370 CPEREP process, operand separators are still required and VM/370 traditionally uses blanks. However, because much of the documentation and examples used are in OS/VS1 or OS/VS2 publications that indicate the comma separator, CPEREP uses either the comma or blank.

CPEREP FILEDEFs

CPEREP issues the FILEDEFs listed below prior to invoking OS/VS EREP. These allow the corresponding EREP files to be simulated by CMS.

```
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
```

When a mode letter of X is shown, X represents the read/write disk that has the most free space when CPEREP is invoked. At the end of the run, the FILEDEFs listed above are cleared with the exception of the EREPPT, ACCIN, and ACCDEV FILEDEFs. For these, it is expected that the user may sometimes be supplying them and so they are left intact.

For those FILEDEFs listed above where NOCHANGE is an option, the user can supply an overriding FILEDEF of his own prior to invoking CPEREP (but see explanations below). The NOCHANGE option in CPEREP's FILEDEF means that it cannot change the user's prior FILEDEF.

EREPPT This is EREP's printer file to which the report output is sent. The user can override this FILEDEF with a FILEDEF of his own which he can issue prior to invoking CPEREP.

SYSIN This is a workfile, built by CPEREP, and read by OS/VS EREP. The user is not concerned with it. It generally is a file containing only a few records. It is placed on the read/write disk having the most available space, and it is erased automatically at the end of the run because its filemode number is 3. In those runs where there is no data to go into SYSIN, CPEREP issues FILEDEF SYSIN DUMMY for it rather than the FILEDEF shown above.

SERLOG — This represents the SYS1.LOGREC data set of OS/VS. Since EREP's I/O to SYS1.LOGREC is trapped by DMSIFC, no I/O is ever performed with this FILEDEF. Nevertheless, the FILEDEF is required to satisfy the requirements of the OPEN and CLOSE issued by EREP since these are not trapped. Although a FILEDEF is defined, no corresponding file ever exists on any disk.

TOURIST — This is the message data set, directed to the user's terminal. EREP writes messages and diagnostics to this file.

DIRECTWK — This is a workfile, built and read by OS/VS EREP. It is not always created; whether it is created or not depends upon the particular report and whether or not the input comes from a history tape. This file may be quite large since it contains all input error records. The file is placed on the read/write disk having the most available space and is erased at the end of the run.

ACCDEV — This is the accumulation file; normally, it is a tape on tape drive 181. This file is used if ACC=Y is specified either explicitly or implicitly. If ACC=Y is specified, tape 181 is rewound and spaced forward over the existing file and then backspaced over the tape mark before any writing is done. In this way, the tape is positioned to write new records at the end of the accumulation file.

By issuing his own FILEDEF for ACCDEV before invoking CPEREP, the user can override CPEREP's FILEDEF; thus, he can accumulate to another tape drive or to a disk file. However, the positioning of tape 181 is independent of the user's FILEDEF and this causes two problems: (1) Regardless of the user's FILEDEF, CPEREP attempts to position tape 181 as long as there is a tape 181 attached to the virtual machine (and provided that ACC=Y). If 181 is attached and ready, it is positioned; if attached but not ready, the operator is notified and CPEREP waits for him to make it ready. The solution to this problem is to DETACH tape 181 before running CPEREP. (2) The second problem is that CPEREP does not automatically position the user defined file before writing into it as it does when the file is tape 181. If the user defines the file to another tape drive, the solution to the problem is to issue appropriate CMS TAPE commands to position the tape before invoking CPEREP. If the user defines the file to a disk, the solution to the problem is for him to specify the DISP MOD option in his FILEDEF if he wants to add records at the end of an existing disk file.

Both RECFM and BLKSIZE must always be specified. The record format must be either V or VB.

ACCIN — This is the history file, normally it is a tape on tape drive 182. This file is used if either MERGE=Y or HIST=Y is specified explicitly or implicitly. (HIST=Y is implied when certain reports are requested). If either MERGE=Y or HIST=Y is specified, tape 182 is rewound before any reading is done.

By issuing his own FILEDEF for ACCIN prior to invoking CPEREP, the user can override CPEREP's FILEDEF. Thus, he can read history data from another tape drive or from a disk file. However, the rewinding of tape 182 is independent of the user's FILEDEF and this causes two problems: (1) Regardless of the user's FILEDEF, CPEREP attempts to rewind tape 182 as long as there is a tape 182 attached to the virtual machine (and provided that MERGE=Y or HIST=Y). If

182 is attached and ready, it is rewound; if attached but nct ready, the operator is notified and CPEREP will wait for him to make it ready. The solution to this problem is to DETACH tape 182 before running CPEREP. (2) The second problem is that CPEREP does not automatically rewind the user defined file (if it is another tape drive) before reading from it as it would for tape 182. The solution to this problem is for the user to issue a CMS TAPE command to rewind the tape prior to invoking CPEREP.

The record format must be either V or VB. If the history file is on a tape, both RECFM and BLKSIZE must be specified.

CPEREP Applications

The following examples show CPEREP used in applications that can benefit an installaticn's operation.

Example 1:

When the operator receives the message telling him that the error recording area is 90 percent full, he may have to act quickly to dump the error recording area to a CPEREP accumulation tape if he is to avoid losing data. This dumping is referred to as the "emergency offload" (although CPEREP does not support the OS/VS offload program (IFCOFFLD) to perform this function). In this situation, to save time and avoid mistakes, the operator may want to avoid entering the necessary control parameters frcm the terminal. Instead, he can have CPEREP read a file of control parameters that, in effect, provide an immediate offload facility. The file should contain the following three parameters which can all be put in the file on a single 80 byte record:

```
SYSUM=Y  ACC=Y  ZERO=Y
```

If the file happens to be named OFFLOAD EREPCTL, then the operator could achieve an "emergency off load" by typing the following line after he has attached a virtual tape 181 to his virtual machine:

```
CPEREP   OFFLOAD   EREPCTL
```

The captured error records on the accumulator (ACC) tape can then be processed later at a more convenient time.

Example 2:

In installaticns having more than one system installed, with devices shared between systems, it is desirable to run reports covering the entire installation rather than individual reports on each system. If the error log records from each system are accumulated on separate accumulation tapes, then reports covering the overall installation cannot be produced until the separate tapes are combined in some way. Under OS/VS, this is easy since the tapes can be concatenated into a single input file using the OS/VS JCL. But VM/370's CMS has no corresponding facility for concatenation, so a less direct method of combining the data is used. One such method is to use the accumulation capability of CPEREP to copy input tapes one at a time to an accumulation output tape. For example, if you have five input tapes to be combined, you would run CPEREP five times with the following control parameters:

```
HIST=Y   PRINT=NO   ACC=Y
```

In this way a combined set of data is built up on a sixth (output) tape. Or you could make just four runs if you do not mind using one of the five input tapes as an output tape and adding the other four input tapes to it.

The order of the input tapes and their chronological order of creation make no difference to the accumulation process. When reports are generated from an accumulation tape (or from anywhere) the reports are effectively sorted in the desired sequence.

Example 3:

At installations having shared devices, channel switches, etc., the SHARE parameter is used for running most reports. And a large number of SHARE parameters might be required in a large installation. Furthermore, this set of SHARE parameters would be fairly stable, changing only when the installation's I/O configuration changes. In this case it is probably worthwhile to keep the required set of SHARE parameters in a file by itself. The SHARE parameters in the file would be followed by a TERMINAL control parameter so that the "Mixed Method" of entering operands could be used. The fileid of this file would be specified on the CPEREP command line. As a result, the SHARE parameters would be read from the file and then additional parameters would be prompted for at the terminal, allowing parameters requesting a particular report to be entered.

SET RECORD and SET MODE Facility

SET RECORD Facility

The CP SET command with the RECORD option is a valuable asset in the diagnosis of system hardware I/O problems on a System/370 controlled by VM/370. The SET RECORD can only be invoked by the Class F user.

By inserting the proper operands in this command, the error recording area receives records that were triggered by the following items

- Specific real I/O device address
- Specific limit count
- Specific sense byte data

The importance of the SET RECORD facility is readily apparent when one realizes that virtual machine I/O errors are not necessarily recorded on the system's error recording area. If SVC 76 is invoked, however, the chances of the loss of error records is lessened. CP records errors associated with its own operations; that is, spooling, paging, and CMS operations and so forth. Errors detected during CP initialized recovery attempts are not recorded by the SET RECORD option. It does not normally record I/O outboard errors associated with virtual machine operations unless it is specifically requested by a virtual machine invoking the SVC 76 instruction.

Outboard I/O errors from dedicated virtual machine devices are reflected to the virtual machine that initiated the SIO action. It is that virtual machine's responsibility to initiate recovery. This may entail, beside retry routines, error recording on another dedicated device of that virtual system. It is therefore conceivable that for multiple virtual machines on one VM/370 system, there could be multiple error recording or LOGREC areas. To the CE at the central site and to users of the virtual system, this could present many problems.

To circumvent the apparent problems, the CE can invoke the SET RECORD command. The SET RECORD command format and operands are fully described in the _VM/370 Operator's Guide_. This command allows the CE to monitor and record any specific unit check condition on any specified device. If the malfunction is sporadic in nature and there are large time lapses between failures, the SET RECORD command can be invoked and not disturbed for however long it takes to capture the quantity of errors desired for the device specified. If SET RECORD OFF is not entered, intensive recording is automatically terminated after 10 errors are recorded in the VM/370 error recording area for that device. SET RECORD values are not retained by system checkpoint activity, so if the VM/370 system operation is suspended and then loaded again, the SET RECORD command must also be reinvoked if monitoring of a specific device is to continue.

The SET RECORD function is available for one I/O device at a time. To specify a different device, invoke the SET RECORD command again with the desired new operands. CP overlays the first SET RECORD request with the second request so that the first SET RECORD request is obliterated. There is no way to initiate this method of error recording on multiple I/O devices.

The SET RECORD command contains a LIMIT operand. The LIMIT operand is the threshold value that indicates when recording is to take place.

Sense byte data consists of a selected sense byte bit or the logic output of the "and" or "or" condition of two selected sense byte bits.

Examples of the format for employing Intensive Recording mode follow:

```
S   REC ON raddr LIMIT nn BYTE nn BIT n AND BYTE nn BIT n
SET REC ON raddr LIMIT nn BYTE nn BIT n OR  BYTE nn BIT n
SET RECORD ON LIMIT nn BYTE nn BIT n
```

__Sample of SET RECORD command usage__:

```
s rec on 127 limit 05 byte 00 bit 4 and byte 03 bit 3

    --or--

set rec on 314 limit 01 byte 00 bit 7 or byte 01 bit 7
```

The first sample shows that when the real device addressed as 127 has accumulated five errors as a result of the "and" condition of bits 4 and 3 of sense bytes 00 and 03, respectively, the errors are recorded.

The second sample is similar but when this device, whose real address is 314, enccunters a bit 7 active either in byte 00 or 01, the errors are recorded.

To turn off all intensive recording, make the following entry. This nullifies previously issued SET RECORD option.

__Example__

```
SET RECORD OFF
```

SET MODE Facility

The function of the recovery facility mode switching routine is to allow installation support personnel to change the mode that CPU retry and ECC recording are operating in. This routine receives control when a user with class F privileges issues some form of the SET MODE command. A check is initially made to determine whether or not this is VM/370 running under VM/370. If it is, then the request is ignored and control is returned to the calling routine. The SET MODE command is described in the __VM/370 Operator's Guide__.

The SET MODE command has five operands which are described as follows:

The MAIN operand applies to processor storage bit failures that are detected and corrected by hardware logic. The SET MODE MAIN command is treated as invalid if issued on a 3031, 3032, or 3033 processor.

The RETRY cperand pertains to processor instruction failures that are CPU detected and corrected by recycling the failing instruction through the system logic again.

The QUIET operand causes the specified facility (MAIN or RETRY) to be placed in quiet mode, in order tc preclude the recording of errors.

The RECORD operand causes the count of soft errors to be reset to zero and the specified facility to be placed in record mode; this is the mode in which CPU retry and/or ECC errors are recorded.

The CPUID operand is an optional selection operand effective only fcr the attached processor mode of operation of VM/370. The CPUID operand allows the user to apply the previously specified operands to either the attached prccessor or the main processor. If CPUID is not specified cn the command line, then the applicable MAIN, RETRY, QUIET, and RECORD operands apply tc both processors.

The errcr recording of instructions that are retry-corrected cr ECC-corrected storage errors is determined by the setting of contrcl register 14 bit 4.

ON = RECORD MODE
OFF = QUIET MODE

The initial setting is a function of processcr design (that is, the system reset can either initialize soft recording or not); afterwards, soft recording can be invoked only by the SET MODE command. Suspensicn of soft reccrding can be achieved by arriving at the threshold count cr by invoking the SET MODE QUIET option. Note that the status of record mode is retained by VM/370 through "warm" and "cold" start procedures (system abend conditions). For more details on soft recording, refer to the topic "Recovery Modes" in Section 2.

CP's TRACE Command

The CP TRACE facility of VM/370 is a powerful tool that can assist the CE in problem diagnosis. By the use of this command, a printout of designated program activity can be obtained. This command belongs to the G privilege class and can be employed by the general user as an aid in program fault analysis.

The command is flexible to the extent that a program trace can be obtained for a particular machine operation or a mix of system machine operations comprising some or all of the following:

- SVC interrupts
- I/O interrupts
- Program interrupts
- External interrupts
- Privileged, Branch, or All instructions
- Channel instructions and related activity
- CSWs

The format and operands of the TRACE command are described in the VM/370 CP Command Reference for General Users.

Certain functions provided by TRACE operands are obviously useful to the CE. For example, SIO or CSW with the I/O interrupt operand; both indicate the real device address with which I/O operation was involved.

In using the TRACE command, output data is printed on the CE's virtual machine console if the PRINTER option is not invoked. The CE's terminal (the default output device) is specified by the BOTH operand or by invoking the TERMINAL operand. Thus, in the course of using TRACE, the printer output device is altered. The PRINTER operand refers to the virtual high speed printer. The file for the PRINTER containing the TRACE activity is relayed to the real spooling printer after the CLOSE command is invoked to close or signify the end of that file.

TRACE activity, optioned to the printer directly or indirectly by invoking the SPOOL CONSOLE command, is transmitted to a remote printer by utilizing the facilities of RSCS. Remote spooling procedures are described in the VM/370 RSCS User's Guide.

In operation, after invoking the TRACE command, the TRACE operation halts the program being traced after executing the first encountered condition specified by the TRACE operands. To initiate the program again and resume TRACE activity, the CE must issue the BEGIN command.

Before resuming TRACE execution, the virtual machine user can alter the previously imposed trace facilities. This procedure is described in the following text.

Assume a program is loaded in the virtual CPU. The virtual system then enters console function mode prior to program execution. The TRACE command function is now used with the ALL option and the BEGIN command is invoked. The ALL option allows instruction tracing among other things. Therefore, the virtual system after startup again enters console function mode after the printout of the first executed instruction. Assume now, that it has been decided not to record all facilities of the TRACE command, and that SVC, I/O, and program interruption tracings are to be eliminated. These interrupt conditions are now entered with the TRACE command and the OFF option. BEGIN is again issued, and the subsequent TRACE table no longer contains these interrupt entries.

The TRACE command then has the flexibility of accepting multiple cr single additions or deletions of operands.

After the next printout at the terminal, execution of the program is again halted in console function mode. An examination discloses that the trace facilities are satisfactory, the TRACE command is then invoked with the RUN option. Now, the program, after executing another BEGIN, runs to the completion, printing out trace data without any BEGIN intervention. If, however, the program is looping, or if the user wants to suspend tracing activity, he signals CP by means of an attenticn interrupt, then enters:

```
trace end
```

Examples of invoking TRACE are:

```
trace svc
trace all
trace svc program i/o both run
tr program off
tr end
tr ccw printer
```

To summarize, the TRACE command allows tracing SVC, I/O, program and external interrupt conditions as well as SIO, privileged instructions, CCW, branch instructions, all instructions or all of the above.

Trace facilities can either be turned on or off. Trace printout can be optioned tc the user's terminal or the spool virtual printer or both. Using the facilities of RSCS, trace output can be spocled to a remote printer.

The TRACE command executed on the user's terminal defaults to the NORUN condition (stops after each trace print line) unless the RUN option is specified.

For a printout of a trace operation where the virtual printer was used as the output device, the CLOSE PRINTER command must be executed.

Notes:

1. A branch to the next sequential address or to the same address is not identified in the trace table.

2. Erroneous branch I/O, or instruction-tracing results can be obtained when TRACE encounters instructions that examine or modify the next two successive bytes of the following instruction.

3. I/O operations for virtual channel-to-channel adapters, with both ends connected to the same virtual machine, cannot be traced.

Figure 34 shows trace data invoked by applying the CP TRACE command with the fcllcwing opticns:

```
trace sic ccw i/o csw printer
```

The PRINTER operand directs the trace data file to print out on the system's spcoling printer.

```
I/O 001A96 SIO   9C002000  CONS 0009  CC 1
*** 001AEE I/O   0009 ==> 001AB2  CSW 0800
I/O 001A96 SIO   90002000  DASD 0191  CC 0  DASD 0331  CAW 00003560
CCW 003560 07003314 40000006  07AA38 0707AA80 40100006
 "   SEEK  00000000 000004     SEEK  0000017F C000
CCW 003568 29003310 60000004  07AA40 29056310 60800004
CCW 003570 08003568 00000000  07AA48 0807AA40 2910000C
CCW 003578 060036E0 20000050  07AA50 060566E0 20800050
*** 001AEE I/O   0009 ==> 001AB2  CSW 0400
CSW V 0191 00003570 0E000004  R 0331 0007AA48 0E000004
*** 001AEE I/O   0191 ==> 001AB2  CSW 0E00

                 .          .          .          .
                 .          .          .          .
                 .          .          .          .
```

Figure 34. Segment of a Trace Printout of a Program's I/O Operation

See the TRACE command and the complete listing of the printout message formats available with this command in the _VM/370 CP Command Reference for General Users_.

Note: If the virtual machine assist feature is enabled on your virtual machine, CP turns it off while tracing SVC and program interruptions (SVC, PRIV, BRANCH, INSTRUCT, or ALL). After the tracing is terminated with the TRACE END command line, CP turns the assist feature on again.

If the virtual machine is running virtual=real (V=R) with NOTRANS ON, CP forces CCW translation while tracing SIOs or CCWs. After tracing is terminated with TRACE END, CCW translation is bypassed again.

NETWORK TRACE

VM/370 provides a means of capturing the basic transmission unit (BTU) header and data information pertaining to a particular 3704/3705 line resource. This is accomplished by invoking the NETWORK TRACE command. NETWORK TRACE is effective only if the 3704/3705 communications controller is loaded with either the network control program (NCP) or the partitioned emulation program (PEP). NETWORK TRACE is not effective for 3704/3705 devices loaded with the 270x emulation program (EP). This CP command can only be invoked by the privilege class F user.

For information concerning the header and other related information concerning the 3704 and 3705 operations, consult the publication _IBM 3704 and 3705 Communications Control Network Control Program Generation and Utilities Guide and Reference Manual_ (for OS/VS TCAM users), Order No. GC30-3007.

For information on how this command is used, see the NETWORK command description in the _VM/370 Operator's Guide_.

RSCS Logging

The remote spooling communications subsystem (RSCS) has the ability to log all I/O activity on a particular teleprocessing line. Normally, such logging is not needed but, if a problem exists that requires tracing I/O on a line, logging can be turned on. The RSCS virtual machine operator turns it on and off by issuing the privilege class G command, CMD, with the LOG or NOLOG operand.

To start the logging operation, the RSCS operator issues CMD, then enters the 1 to 8 character link identifier of the remote station associated with the link, followed by the keyword, LOG. LOG starts the logging of I/O activity on the line and NOLOG stops the logging operation. The format and operands of CMD are described in the *VM/370 Remote Spooling Communications Subsystem (RSCS) User's Guide*.

The output of the logging is a printer spool file containing a one-line record for each I/O transaction on the line; for example, each time a teleprocessing buffer is written into or read out of.

When logging is turned off (NOLOG) the output is printed. The distribution code on the printer output is the linkid for which logging was being done.

The contents of the log record in order of occurrence from left to right are as follows:

21 bytes		The first 21 bytes of the log record are the first 21 bytes of the teleprocessing buffer, including BSC bytes, MULTI-LEAVING[1] bytes (for SML only), and enough initial data bytes to fill the field.
7 bytes	Read I/O	Last seven bytes of the CSW.
	SML Write I/O	First seven bytes of SML buffer (the buffer header used internally by SML but not transmitted).
	NPT Write I/O	Not applicable.
3 bytes		RSCS I/O synch lock for this I/O operation.
1 byte		These bytes are the sense bytes (if any).
8 byte		CCW associated with the I/O operation

The fields of the record are separated by blanks. Figure 35 shows the read and write log records for SML.

[1] Trademark of IBM

SAMPLES OF READ AND WRITE RECORDS FOR SML

```
1070                                        0779C80C00018E 800000  00  0207100720000190
1070                                        0779C80C00018E 000000  00  0107100760000002
1002808FCF9094000026                        0779C80C000186 800000  00  0207100720000190
1002818FCFA0940000                          0779C80C000186 8C0000  00  0107100760000009
1002818FCF9491C140009483C140009483C1400094  0779C80C00003C 800000  00  0207100720000190
1070                                        0779C80C00003C 800000  00  0107119F60000002
                                            0779C80E000190 800000  01  0207119F20000190
1002828FCF9483C8C6C9D3C57A40C4E787C4C5E7C5  0779C80E00005C 800000  02  0207119F20000190
323D                                        0779C80E00005C 000000  00  0107119F60000002
1002828FCF9483E4C4C5E2E37A40C8D6E2E3D3C9D5  0779C80C00000C 800000  00  0207119F20000190
1070                                        0779C80C00000C 000000  00  0107100760000002
1002838FCF9481CC50D5E4D4C2C5D9407E4050F100  0779C80C000008 800000  00  0207100720000190
1070                                        0779C80C000008 000000  00  0107119F60000002
1002848FCF9481FF5C5C5C40C3C1E4E2C5E240E3C8  0779C80C000003 800000  00  0207119F20000190
1070                                        0779C80C000003 000000  00  0107100760000002
1002858FCF9481C7C3D740D84007C6009481E350E3  0779C80C0000E7 800000  00  0207100720000190
1070                                        0779C80C0000E7 000000  00  0107119F60000002

                                              SML INTERNAL SYNCH SENSE        CCW
                                                 BUFFER    LOCK  BYTE
21 BSC, MULTILEAVING AND DATA BYTES                - OR -

              TP BUFFER                     ADDR STATUS COUNT
                                                 BYTES

                                                  CSW
```

Figure 35. Read and Write Log Records for SML.

System abend (abnormal ending) conditions can be prompted by real System/370 system operator intervention involving PSW restart. System abend conditicns can also be caused by program SVC 0 operation. This may happen when CP is in a program predicament which it cannot correct and therefore cannot validly continue processing. SVC 0 may also occur when the CP system recognizes a catastrophic situation that was prompted by a hardware malfunction.

When such situations occur, SVC 0 invokes a system dump. The dump operation prompted by the main processor (or attached processor, if applicable) captures the system registers and defined storage areas and may or may nct contain a trace table with the sequence of events that occurred just before the condition that caused the abnormal ending. This trace table data appears in dump output if the CP MONITOR command with the STOP operand was not invoked before the dump operation. Consult the _VM/370 System Programmer's Guide_ for details of the CP MONITOR command and CP's internal trace facility. The selection of such options can expedite system recovery.

Note: The internal trace facility should not be confused with the CP TRACE command functions.

Facilities also exist within CP to allow the automatic spooling of abend dump files onto DASD devices (if so desired) by a CP SET command option.

During the interval between the malfunction and the resumption of system activity, logout or error recording may also have taken place. The system dump file (previously spooled to a DASD device) can then be processed and formatted by the VMFDUMP command. This command extracts data pertinent to the type of abend and creates a problem report. It also prompts the user for additional information that describes the problem. The VMFDUMP command is described fully in the _VM/370 Interactive Problem Control System (IPCS) User's Guide_ and the _VM/370 Operator's Guide_.

The extent of system abend and VMFDUMP utilization is controlled by the system operator and cannot be invoked by the CE.

Data concerning hardware status, sense, and I/O operation is in the RDEVBLOK, IOBLOK, and IOERBLOK control blocks.

The RDEVBLOK and IOBLOK illustrations are given in Figures 9 and 10 in section 3 under "I/O Error Recovery--Detailed Description."

The information in these blocks, in conjunction with program support personnel or customer program personnel, may assist the CE in defining the cause of the system fault or aid in reconstructing the sequence that prompted the system fault. Basically, the full formatted dump produces the results discussed below.

1. The header contains the time and date of the abend as well as an abend code and the processor identity that initiated the dump operation.

2. This is followed by PSWs, CAW, CSW, the time-of-day clock, the clock comparator, the prefix register, the processor and interval timer values of the processor that caused the abend.

3. This step applies to attached processor operations only: Next the PSA (prefix storage area) of the main processor is printed followed by the PSA values for the attached processor if the system was in attached processor mode when the abend occurred.

4. Following this is data extracted from CP's symbol table (DMKSYM), which contains the storage location of selected entry points for the CP system.

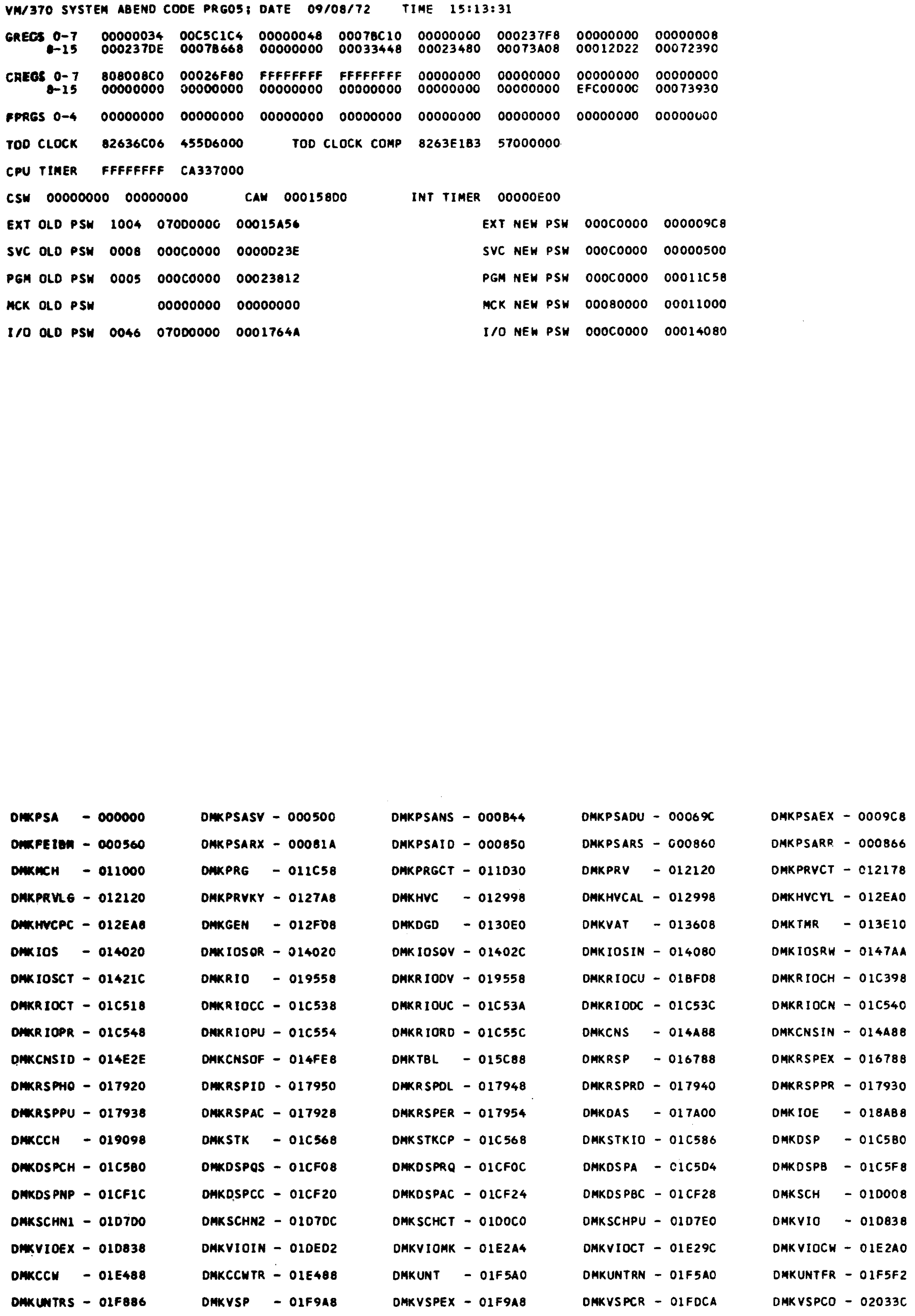
5. The tabulations that follow the symbol table printout are pages that are applicable to the real system hardware. These blocks represent every channel, every control unit, and every device that is represented as available to VM/370 operations. These blocks are designated as RCHBLOK, RCUBLOK, and RDEVBLOK, respectively. Those devices that are actively involved with system operations at the occurrence of system abnormal ending are indicated by an adjacent display of an active IOBLOK.

6. These blocks are followed by statistics applicable to the spool files that are applied to the spooling devices (system reader, printer, and punch). These blocks are designated as spooled file blocks (SFBLOK). If no spooling activity exists, then the VMFDUMP output indicates this (as indicated in the following VMFDUMP sample).

7. The spooled file data is followed by the CORTABLE. This table indicates the real address of the four doubleword entries that contain pointers to the SWPTABLE, the PAGTABLE, the previous entry in queue, and the next entry in queue. Also contained in this block are flags to indicate whether the page is on the flush list, the free list, or is shared or unavailable. The CORTABLE printout also indicates the user identity and the page assignment at the time of the abnormal ending.

8. After the CORTABLE, there is a progression of data blocks that are related to each logged on user. They are listed in the following order: the virtual machine blocks (VMBLOCK), virtual channel blocks (VCHBLOK), virtual control unit blocks (VCUBLOK), virtual device blocks (VDEVBLOK), and virtual console control blocks (VCONCTL). This is followed by Segment tables, Page tables and Swap tables (SEGTABLE, PAGTABLE, SWPTABLE), respectively that are applicable to the associated user's virtual machine activity.

Figure 36 illustrates the output of a formatted VMFDUMP operation (uniprocessor mode).



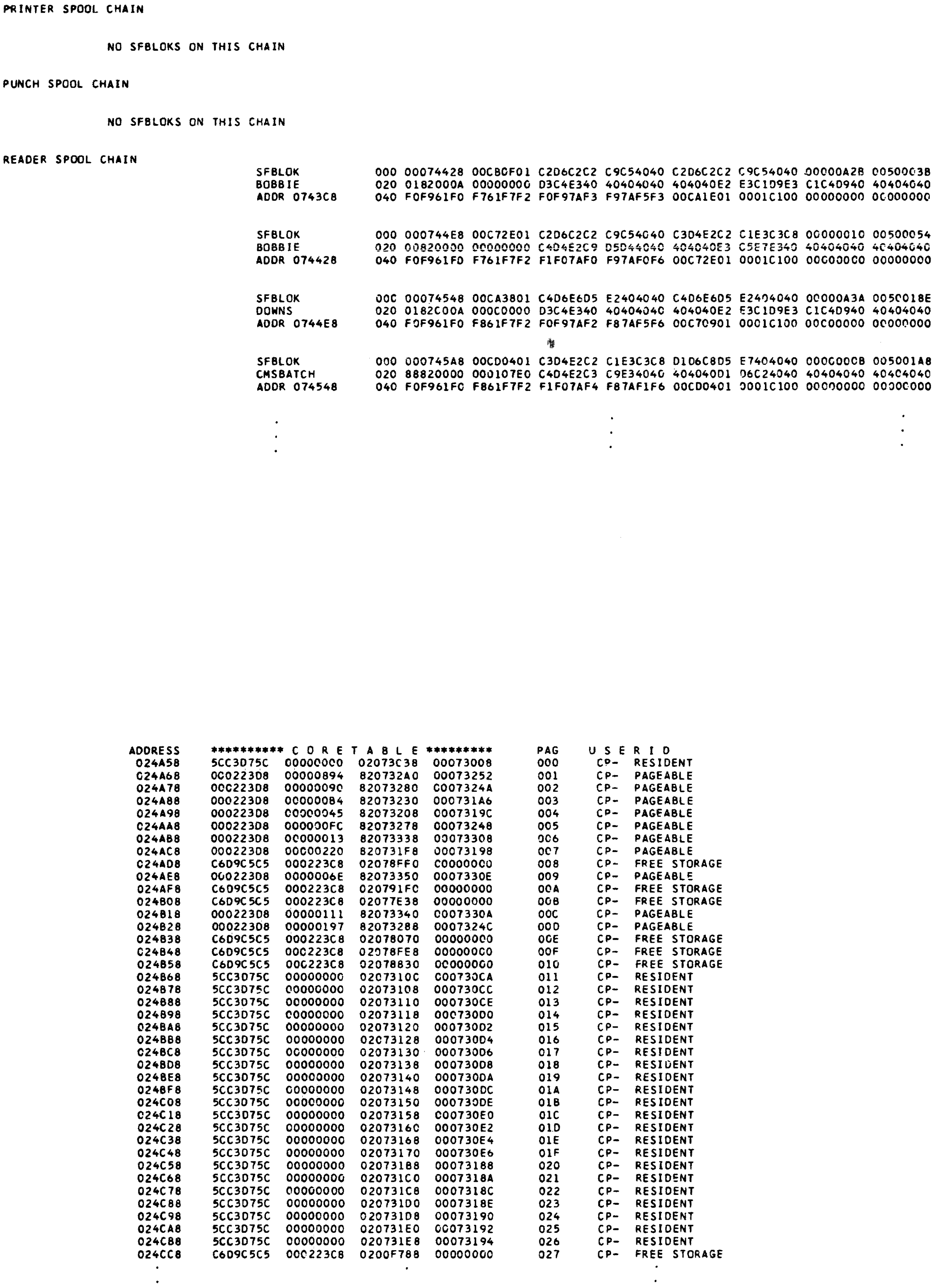
```
VM/370 SYSTEM ABEND CODE PRG05; DATE  09/08/72    TIME  15:13:31

GREGS 0-7    00000034  00C5C1C4  00000048  00078C10  00000000  000237F8  00000000  00000008
     8-15    000237DE  00078668  00000000  00033448  00023480  00073A08  00012D22  00072390

CREGS 0-7    808008C0  00026F80  FFFFFFFF  FFFFFFFF  00000000  00000000  00000000  00000000
     8-15    00000000  00000000  00000000  00000000  00000000  00000000  EFC0000C  00073930

FPRGS 0-4    00000000  00000000  00000000  00000000  00000000  00000000  00000000  00000000

TOD CLOCK    82636C06  455D6000      TOD CLOCK COMP  8263E1B3  57000000

CPU TIMER    FFFFFFFF  CA337000

CSW  00000000  00000000        CAW  000158D0        INT TIMER  00000E00

EXT OLD PSW  1004  070D0000  00015A56                         EXT NEW PSW  000C0000  000009C8

SVC OLD PSW  0008  000C0000  0000D23E                         SVC NEW PSW  000C0000  00000500

PGM OLD PSW  0005  000C0000  00023812                         PGM NEW PSW  000C0000  00011C58

MCK OLD PSW        00000000  00000000                         MCK NEW PSW  00080000  00011000

I/O OLD PSW  0046  070D0000  0001764A                         I/O NEW PSW  000C0000  00014080
```

```
DMKPSA   - 000000    DMKPSASV - 000500    DMKPSANS - 000B44    DMKPSADU - 00069C    DMKPSAEX - 0009C8

DMKPEIBM - 000560    DMKPSARX - 00081A    DMKPSAID - 000850    DMKPSARS - C00860    DMKPSARR - 000866

DMKMCH   - 011000    DMKPRG   - 011C58    DMKPRGCT - 011D30    DMKPRV   - 012120    DMKPRVCT - C12178

DMKPRVLG - 012120    DMKPRVKY - 0127A8    DMKHVC   - 012998    DMKHVCAL - 012998    DMKHVCYL - 012EA0

DMKHVCPC - 012EA8    DMKGEN   - 012F08    DMKDGD   - 0130E0    DMKVAT   - 013608    DMKTMR   - 013E10

DMKIOS   - 014020    DMKIOSQR - 014020    DMKIOSQV - 01402C    DMKIOSIN - 014080    DMKIOSRW - 0147AA

DMKIOSCT - 01421C    DMKRIO   - 019558    DMKRIODV - 019558    DMKRIOCU - 01BFD8    DMKRIOCH - 01C398

DMKRIOCT - 01C518    DMKRIOCC - 01C538    DMKRIOUC - 01C53A    DMKRIODC - 01C53C    DMKRIOCN - 01C540

DMKRIOPR - 01C548    DMKRIOPU - 01C554    DMKRIORD - 01C55C    DMKCNS   - 014A88    DMKCNSIN - 014A88

DMKCNSID - 014E2E    DMKCNSOF - 014FE8    DMKTBL   - 015C88    DMKRSP   - 016788    DMKRSPEX - 016788

DMKRSPHQ - 017920    DMKRSPID - 017950    DMKRSPDL - 017948    DMKRSPRD - 017940    DMKRSPPR - 017930

DMKRSPPU - 017938    DMKRSPAC - 017928    DMKRSPER - 017954    DMKDAS   - 017A00    DMKIOE   - 018AB8

DMKCCH   - 019098    DMKSTK   - 01C568    DMKSTKCP - 01C568    DMKSTKIO - 01C586    DMKDSP   - 01C5B0

DMKDSPCH - 01C5B0    DMKDSPQS - 01CF08    DMKDSPRQ - 01CF0C    DMKDSPA  - C1C5D4    DMKDSPB  - 01C5F8

DMKDSPNP - 01CF1C    DMKDSPCC - 01CF20    DMKDSPAC - 01CF24    DMKDSPBC - 01CF28    DMKSCH   - 01D008

DMKSCHN1 - 01D7D0    DMKSCHN2 - 01D7DC    DMKSCHCT - 01D0C0    DMKSCHPU - 01D7E0    DMKVIO   - 01D838

DMKVIOEX - 01D838    DMKVIOIN - 01DED2    DMKVIOMK - 01E2A4    DMKVIOCT - 01E29C    DMKVIOCW - 01E2A0

DMKCCW   - 01E488    DMKCCWTR - 01E488    DMKUNT   - 01F5A0    DMKUNTRN - 01F5A0    DMKUNTFR - 01F5F2

DMKUNTRS - 01F886    DMKVSP   - 01F9A8    DMKVSPEX - 01F9A8    DMKVSPCR - 01FDCA    DMKVSPCO - 02033C
```

Figure 36. Formatted VMFDUMP (Part 1 of 6)

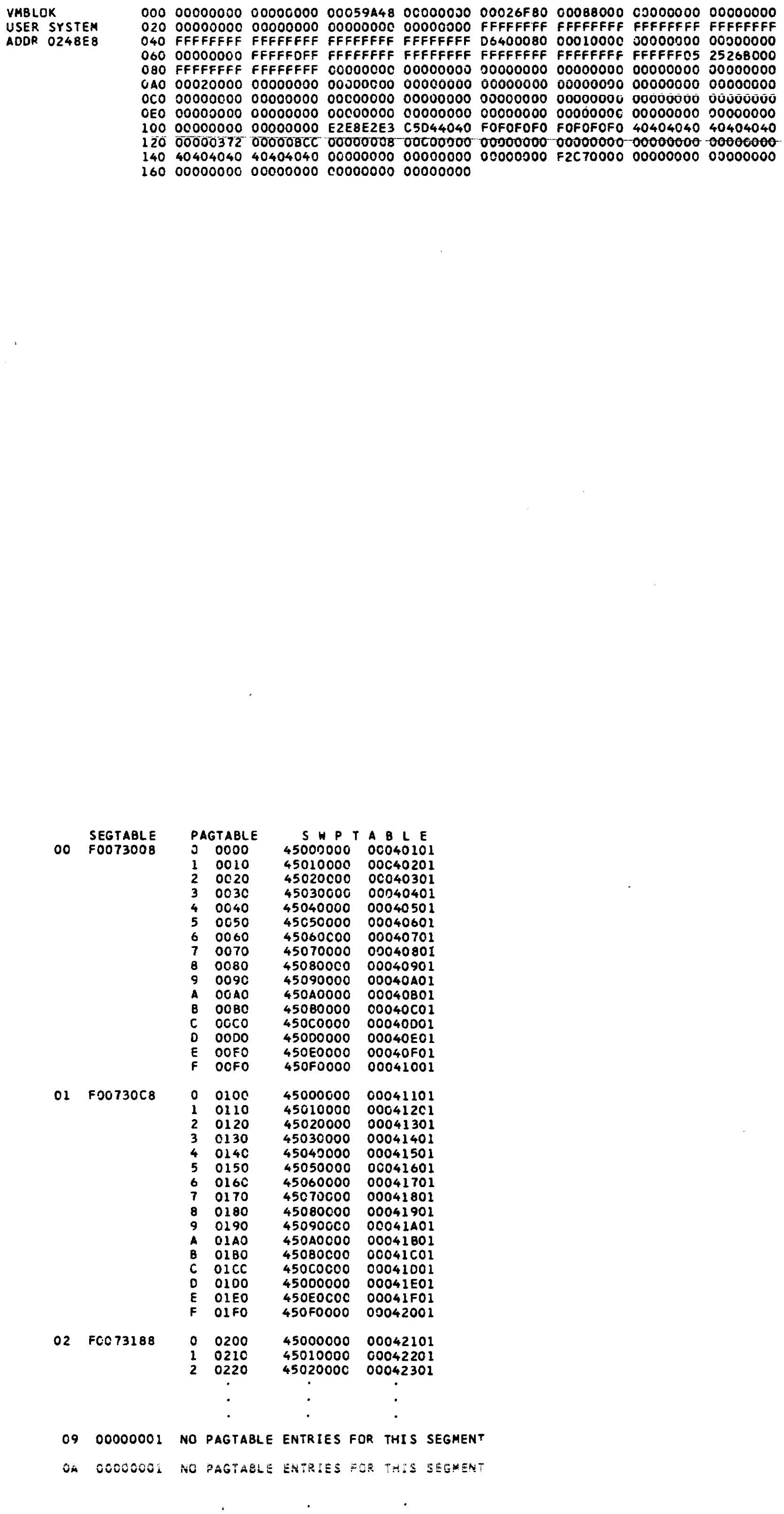
```
RCHBLOK       000 00000000 00000000 0001C398 0001C398 00000000 00000000 00000000 00000000
CHAN 0XX      020 FFFF0000 FFFF0040 0080FFFF 00C0FFFF 0100FFFF 0140FFFF 0180FFFF 01C0FFFF
ADDR 01C398   040 0200FFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF

        RCUBLOK       000 00080000 00000000 0001BFD8 0001BFD8 0001C398 00000000 00000000 00000000
        UNIT 00X      020 FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF 005000A0 C0F00140
        ADDR 01BFD8

                RDEVBLOK      000 000C0000 00022082 000195A8 000195A8 0001BFD8 00000000 00000000 00000000
                DEV  00C      020 00000002 00000000 000248E8 00000000 40404040 40400000 C0000000 00000000
                ADDR 0195A8   040 00000000 00000000 00000000 00000000

                RDEVBLOK      000 000D0000 00021082 000195F8 000195F8 0001BFD8 00000000 00000000 C3C10000
                DEV  00D      020 0000001D 00000000 000248E8 000C0000 40404040 40400000 00000000 00000000
                ADDR 0195F8   040 00000000 00000000 00000000 00000000

                RDEVBLOK      000 000E0000 00031041 00019648 00019648 0001BFD8 00000C00 00000000 C1404040
                DEV  00E      020 000005CA 00000000 C00248E8 00000000 40404C40 40400000 00000000 00000000
                ADDR 019648   040 00000000 00000001 000C0000 00000000

                RDEVBLOK      000 000F0000 00031041 00019698 00019698 0001BFD8 C0000000 00000000 E2404040
                DEV  00F      020 00000077 00000000 000248E8 00000000 40404040 40400000 00000000 00000000
                ADDR 019698   040 00000C00 0000C001 00000000 00000C00

        RCUBLOK       000 00180000 00000000 0001C018 0001C018 0001C398 00000000 00000000 00000000
        UNIT 01X      020 FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFC000
        ADDR 01C018

                RDEVBLOK      000 000F0000 00008000 00019558 00019558 0001C018 00000000 00000000 00000000
                DEV  01F      020 000001C2 00000000 00025260 00090000 40404040 40400000 784A7C7F 82000000
                ADDR 019558   040 00000000 00C00000 C0000000 0000000C

        RCUBLOK       000 00200000 00000000 0001C058 0001C058 0001C398 00000000 00000000 0000000C
        UNIT 02X      020 019001E0 02300280 02D00320 037003C0 04100460 04B00500 055005A0 05F00640
        ADDR 01C058

                RDEVBLOK      000 00000000 80888020 000196E8 000196E8 0001C058 00000000 C000000C 00000000
                DEV  020      020 00000002 00074230 000248E8 00000000 40404040 40400C00 784A7C7F 82000000
                ADDR 0196E8   040 00000000 00001004 00000000 0000000C

                        ACTIVE IOBLOK 000 00208000 00074230 0001CF0C 0001CFCC 00000000 00000000 000248E8 00014E2E
                        ADDR 074230   C20 00015850 00000000 00062278 0C000001 00000000 00000000 00000000 00000000

                   .                                        .                                        .
                   .                                        .                                        .
                   .                                        .                                        .

RCHBLOK       000 02000000 00400000 0001C458 0001C458 00000000 00000000 0000000C 00000000
CHAN 2XX      C20 FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF 0280FFFF FFFFFFFF FFFFFFFF
ADDR 01C458   040 FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF 02C0FFFF FFFFFFFF FFFFFFFF

        RCUBLOK       000 00500000 20000000 0001C258 0001C258 0001C458 00000000 00C00000 00000000
        UNIT 25X      020 22602280 23002350 23A023F0 24402490 FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF
        ADDR 01C258

                RDEVBLOK      000 00000000 20000410 0001B7B8 0001B7B8 0001C258 00000000 00000000 00000000
                DEV  250      020 00000000 00000000 000248E8 00000000 40404040 4040C000 00000000 00000000
                ADDR 01B7B8   040 00000000 00000000 00000000 00000000

                  .                                         .                      .
                  .                                         .                      .
                  .                                         .                      .

        RCUBLOK       000 00D00000 00000000 0001C298 0001C298 0001C458 00000000 00C00000 00000000
        UNIT 2DX      020 24E02530 2580FFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF
        ADDR 01C298

                RDEVBLOK      000 00000000 00700402 0001BA38 0001BA38 0001C298 00000000 00000000 00000000
                DEV  2D0      020 0000EA27 00000000 000740B0 00000000 C3D7C4D9 04F10000 000795A0 00000000
                ADDR 01BA38   040 0001BA38 00000000 00000000 00000000

                                RECBLOK -PAGE 000 0007B890 00291318 FFFFE0FF FFFFFFFF
                                ADDR 0795A0

                                RECBLOK -PAGE 000 00008318 00421718 FFDFFFFF FFFFFFFF
                                ADDR 07B890
```

Figure 36. Formatted VMFDUMP (Part 2 of 6)



```
PRINTER SPOOL CHAIN

              NO SFBLOKS ON THIS CHAIN

PUNCH SPOOL CHAIN

              NO SFBLOKS ON THIS CHAIN

READER SPOOL CHAIN
                                 SFBLOK        000 00074428 00CB0F01 C2D6C2C2 C9C54040 C2D6C2C2 C9C54040 00000A2B 0050003B
                                 BOBBIE        020 0182000A 00000000 D3C4E340 40404040 404040E2 E3C1D9E3 C1C4D940 40404040
                                 ADDR 0743C8   040 F0F961F0 F761F7F2 F0F97AF3 F97AF5F3 00CA1E01 0001C100 00000000 00000000

                                 SFBLOK        000 000744E8 00C72E01 C2D6C2C2 C9C54040 C3D4E2C2 C1E3C3C8 00000010 00500054
                                 BOBBIE        020 00820000 00000000 C4D4E2C9 D5D44040 404040E3 C5E7E340 40404040 40404040
                                 ADDR 074428   040 F0F961F0 F761F7F2 F1F07AF0 F97AF0F6 00C72E01 0001C100 00000000 00000000

                                 SFBLOK        00C 00074548 00CA3801 C4D6E6D5 E2404040 C4D6E6D5 E2404040 00000A3A 0050018E
                                 DOWNS         020 0182C00A 000C0000 D3C4E340 40404040 404040E2 E3C1D9E3 C1C4D940 40404040
                                 ADDR 0744E8   040 F0F961F0 F861F7F2 F0F97AF2 F87AF5F6 00C70901 0001C100 00C00000 00000000

                                 SFBLOK        000 000745A8 00CD0401 C3D4E2C2 C1E3C3C8 D1D6C8D5 E7404040 0000000B 005001A8
                                 CMSBATCH      020 88820000 000107E0 C4D4E2C3 C9E34040 404040D1 D6C24040 40404040 40404040
                                 ADDR 074548   040 F0F961F0 F861F7F2 F1F07AF4 F87AF1F6 00CD0401 0001C100 00000000 00000000
                                   .                                          .                                      .
                                   .                                          .                                      .
                                   .                                          .                                      .




                    ADDRESS     *********** C O R E T A B L E *********     PAG     U S E R I D
                     024A58     5CC3D75C  00000000  02073C38  00073008      000       CP-   RESIDENT
                     C24A68     0C0223D8  00000894  820732A0  00073252      001       CP-   PAGEABLE
                     024A78     00C223D8  00000090  82073280  C007324A      002       CP-   PAGEABLE
                     024A88     000223D8  00000084  82073230  000731A6      003       CP-   PAGEABLE
                     024A98     000223D8  00000045  82073208  0007319C      004       CP-   PAGEABLE
                     C24AA8     000223D8  000000FC  82073278  00073248      005       CP-   PAGEABLE
                     024AB8     000223D8  00000013  82073338  00073308      006       CP-   PAGEABLE
                     024AC8     000223D8  00000220  820731F8  00073198      007       CP-   PAGEABLE
                     C24AD8     C6D9C5C5  000223C8  02078FF0  C0000000      008       CP-   FREE STORAGE
                     024AE8     00022308  0000006E  82073350  0007330E      009       CP-   PAGEABLE
                     024AF8     C6D9C5C5  000223C8  020791FC  00000000      00A       CP-   FREE STORAGE
                     024B08     C6D9C5C5  000223C8  02077E38  00000000      00B       CP-   FREE STORAGE
                     024B18     000223D8  00000111  82073340  C007330A      00C       CP-   PAGEABLE
                     024B28     000223D8  00000197  82073288  0007324C      00D       CP-   PAGEABLE
                     024B38     C6D9C5C5  000223C8  02078070  00000000      00E       CP-   FREE STORAGE
                     C24B48     C6D9C5C5  000223C8  02078FE8  00000000      00F       CP-   FREE STORAGE
                     024B58     C6D9C5C5  000223C8  02078830  00000000      010       CP-   FREE STORAGE
                     024B68     5CC3D75C  00000000  0207310C  C00730CA      011       CP-   RESIDENT
                     024B78     5CC3D75C  00000000  02073108  000730CC      012       CP-   RESIDENT
                     024B88     5CC3D75C  00000000  02073110  000730CE      013       CP-   RESIDENT
                     024B98     5CC3D75C  C0000000  02073118  000730D0      014       CP-   RESIDENT
                     024BA8     5CC3D75C  00000000  02073120  000730D2      015       CP-   RESIDENT
                     024BB8     5CC3D75C  00000000  02073128  000730D4      016       CP-   RESIDENT
                     C24BC8     5CC3D75C  00000000  02073130  000730D6      017       CP-   RESIDENT
                     024BD8     5CC3D75C  00000000  02073138  000730D8      018       CP-   RESIDENT
                     024BE8     5CC3D75C  00000000  02073140  000730DA      019       CP-   RESIDENT
                     024BF8     5CC3D75C  C0000000  02073148  000730DC      01A       CP-   RESIDENT
                     024C08     5CC3D75C  00000000  02073150  000730DE      01B       CP-   RESIDENT
                     C24C18     5CC3D75C  00000000  02073158  C00730E0      01C       CP-   RESIDENT
                     024C28     5CC3D75C  00000000  02073160  000730E2      01D       CP-   RESIDENT
                     024C38     5CC3D75C  00000000  02073168  000730E4      01E       CP-   RESIDENT
                     024C48     5CC3D75C  00000000  02073170  000730E6      01F       CP-   RESIDENT
                     024C58     5CC3D75C  00000000  02073188  00073188      020       CP-   RESIDENT
                     024C68     5CC3D75C  00000000  020731C0  0007318A      021       CP-   RESIDENT
                     024C78     5CC3D75C  00000000  020731C8  0007318C      022       CP-   RESIDENT
                     024C88     5CC3D75C  00000000  020731D0  0007318E      023       CP-   RESIDENT
                     024C98     5CC3D75C  00000000  020731D8  00073190      024       CP-   RESIDENT
                     024CA8     5CC3D75C  00000000  020731E0  00073192      025       CP-   RESIDENT
                     024CB8     5CC3D75C  00000000  020731E8  00073194      026       CP-   RESIDENT
                     024CC8     C6D9C5C5  000223C8  0200F788  00000000      027       CP-   FREE STORAGE
                        .                             .                                           .
                        .                                                                         .
```

Figure 36. Formatted VMFDUMP (Part 3 of 6)



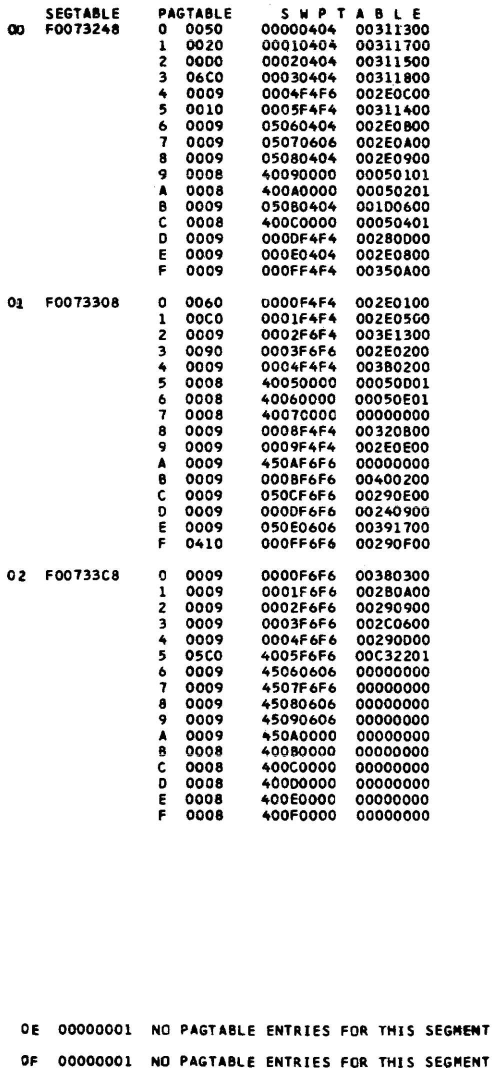
```
VMBLOK          000 00000000 00000000 00059A48 00000000 00026F80 00088000 00000000 00000000
USER SYSTEM     020 00000000 00000000 00000000 00000000 FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF
ADDR 0248E8     040 FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF D6400080 00010000 00000000 00000000
                060 00000000 FFFFFDFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFF05 25268000
                080 FFFFFFFF FFFFFFFF 00000000 00000000 00000000 00000000 00000000 00000000
                0A0 00020000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
                0C0 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
                0E0 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
                100 00000000 00000000 E2E8E2E3 C5D44040 F0F0F0F0 F0F0F0F0 40404040 40404040
                120 00000372 000008CC 00000008 00C00000 00000000 00000000 00000000 00000000
                140 40404040 40404040 00000000 00000000 00000000 F2C70000 00000000 00000000
                160 00000000 00000000 00000000 00000000
```

```
      SEGTABLE      PAGTABLE       S W P T A B L E
00    F0073008      0   0000       45000000   00040101
                    1   0010       45010000   00040201
                    2   0020       45020000   00040301
                    3   0030       45030000   00040401
                    4   0040       45040000   00040501
                    5   0050       45050000   00040601
                    6   0060       45060000   00040701
                    7   0070       45070000   00040801
                    8   0080       45080000   00040901
                    9   0090       45090000   00040A01
                    A   00A0       450A0000   00040B01
                    B   00B0       450B0000   00040C01
                    C   00C0       450C0000   00040D01
                    D   00D0       450D0000   00040E01
                    E   00F0       450E0000   00040F01
                    F   00F0       450F0000   00041001

01    F00730C8      0   0100       45000000   00041101
                    1   0110       45010000   00041201
                    2   0120       45020000   00041301
                    3   0130       45030000   00041401
                    4   0140       45040000   00041501
                    5   0150       45050000   00041601
                    6   0160       45060000   00041701
                    7   0170       45070000   00041801
                    8   0180       45080000   00041901
                    9   0190       45090000   00041A01
                    A   01A0       450A0000   00041B01
                    B   01B0       450B0000   00041C01
                    C   01CC       450C0000   00041D01
                    D   01D0       450D0000   00041E01
                    E   01E0       450E0000   00041F01
                    F   01F0       450F0000   00042001

02    F0073188      0   0200       45000000   00042101
                    1   0210       45010000   00042201
                    2   0220       45020000   00042301
                          .            .          .
                          .            .          .
                          .            .          .

 09   00000001  NO PAGTABLE ENTRIES FOR THIS SEGMENT

 0A   00000001  NO PAGTABLE ENTRIES FOR THIS SEGMENT

                          .            .          .
```

Figure 36. Formatted VMFDUMP (Part 4 of 6)

```
VMBLOK         000 82636538 97A4F000 00C33448 000598E0 00059980 00080000 00059900 00033858
USER V145D     020 00033970 0001A318 00030007 000E0000 00000028 0050FFFF FFFFFFFF FFFFFFFF
ADDR 059A48    040 FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF 80000000 400000DC 004A4000 00000C00
               060 00000000 FFFFFF00 FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF A09D8000
               C80 FFFFFFFF FFFFFFFF 00000000 00000000 0C000000 00000032 00000000 00000000
               0A0 0002C000 00000000 000C0000 00000000 00000000 00000000 00000000 00000000
               CC0 0C00000C 0000000C 00000000 00000000 00000000 00000000 00000C00 00000000
               0E0 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
               100 00000000 00000000 E5F1F4F5 C4404040 F7F5F860 F6F7F6F4 F2F74040 40404040
               120 0000000F 00000000 00000028 00000000 0000000C 0C000000 00000000 000000C0
               140 E2D3C5C5 D7404040 826364D0 0C000000 0000001F 000C0000 0000A240 C0000000
               160 000000C0 00000000 00000C00 00000000


VCHBLOK        000 00000000 00000000 C00000F0 FFFF0028 FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF
CHAN 0XX       020 FFFFFFFF FFFF0050
ADDR 0599D0


          VCUBLOK        000 00000000 C0000000 FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF
          UNIT 00X       020 00380070 00A8FFFF
          ADDR 033858


                    VDEVBLOK       000 000C0000 20820000 00000000 00000000 00000000 00000000 C0000000 00000000
                    DEV  00C       020 C1000C00 00000008 00000000 00059A48 00000000 00000C00
                    ADDR 0339A8


                    VDEVBLOK       C00 000D0000 10820000 00000000 00000000 00000000 000000C0 00000000 00000000
                    DEV  00D       020 C100000C 00010000 00000000 00059A48 00000000 00000000
                    ADDR 0339E0


                    VDEVBLOK       000 000E0000 10410000 0000000C 000C0000 000C0000 00000C00 00000000 00000000
                    DEV  00E       020 C1000000 00010000 00000000 00059A48 00000000 00000000
                    ADDR 033A18


          VCUBLOK        000 00100000 00000000 FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF
          UNIT 01X       020 FFFFFFFF FFFF0000
          ADDR 033948


                    VDEVBLOK       000 000F00C0 80000000 00000000 00000000 000C0000 00000000 00008 8F0 00000000
                    DEV  01F       020 C0000000 00000000 00000000 00059A48 00000000 00000000
                    ADDR 033970


                                        VCONCTL        000 00000000 0000000C 00000C00 00000000 00000000 00000000
                                        ADDR 0088F0


          VCUBLOK        000 00300000 00000000 00E00118 FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF
          UNIT 03X       020 FFFFFFFF FFFFFFFF
          ADDR 033880




                    VDEVBLOK       000 00000000 80100000 00000000 0000000C 00000000 0C000000 0C00C0C0 00000000
                    DEV  030       020 0C000C00 00000000 00000000 00059A48 00000000 00000000
                    ADDR 033A50


                    VDEVBLOK       000 00010000 80100000 00000000 00000000 00000000 00000000 00000000 00000000
                    DEV  031       020 00000000 00000000 000C0000 00059A48 00000000 00000000
                    ADDR 033A88


          VCUBLOK        000 00F00000 C0000080 FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF
          UNIT 0FX       020 FFFFFFFF FFFF0150
          ADDR 0338A8


                    VDEVBLOK       000 000F0000 20400000 00000000 0000000C 000000C0 00000000 00000000 00000000
                    DEV  0FF       020 00000000 00000008 00000000 00059A48 C000000C 00000000
                    ADDR 033AC0


VCHBLOK        000 C1000000 C0000080 FFFFFFFF FFFF0078 FFFFFFFF FFFFFFFF FFFF00C8 FFFFFFFF
CHAN 1XX       C20 FFFFFFFF FFFFFFFF
ADDR 0599F8


          VCUBLOK        000 00300000 C0000000 018801C0 FFFFFFFF FFFFFFFF FFFF01C0 FFFFFFFF FFFFFFFF
          UNIT 13X       020 FFFFFFFF FFFFFFFF
          ADDR 0338D0


                    VDEVBLOK       000 00000000 0003A000 00000048 40C4E2C3 D6D5D5C5 C3E340C8 C5C1D3C4 40404040
                    DEV  130       020 E4E2C5D9 E2407E40 F0F2F961 F7F24040 40404040 40404040
                    ADDR 033AF8
```

Figure 36. Formatted VMFDUMP (Part 5 of 6)



```
    SEGTABLE      PAGTABLE      S W P T A B L E
00  F0073248      0  0050      00000404  00311300
                  1  0020      00010404  00311700
                  2  0000      00020404  00311500
                  3  06C0      00030404  00311800
                  4  0009      0004F4F6  002E0C00
                  5  0010      0005F4F4  00311400
                  6  0009      05060404  002E0B00
                  7  0009      05070606  002E0A00
                  8  0009      05080404  002E0900
                  9  0008      40090000  00050101
                  A  0008      400A0000  00050201
                  B  0009      050B0404  001D0600
                  C  0008      400C0000  00050401
                  D  0009      000DF4F4  00280D00
                  E  0009      000E0404  002E0800
                  F  0009      000FF4F4  00350A00

01  F0073308      0  0060      0000F4F4  002E0100
                  1  00C0      0001F4F4  002E0500
                  2  0009      0002F6F4  003E1300
                  3  0090      0003F6F6  002E0200
                  4  0009      0004F4F4  003B0200
                  5  0008      40050000  00050D01
                  6  0008      40060000  00050E01
                  7  0008      40070000  00000000
                  8  0009      0008F4F4  00320B00
                  9  0009      0009F4F4  002E0E00
                  A  0009      450AF6F6  00000000
                  B  0009      000BF6F6  00400200
                  C  0009      050CF6F6  00290E00
                  D  0009      000DF6F6  00240900
                  E  0009      050E0606  00391700
                  F  0410      000FF6F6  00290F00

02  F00733C8      0  0009      0000F6F6  00380300
                  1  0009      0001F6F6  002B0A00
                  2  0009      0002F6F6  00290900
                  3  0009      0003F6F6  002C0600
                  4  0009      0004F6F6  00290D00
                  5  05C0      4005F6F6  00C32201
                  6  0009      45060606  00000000
                  7  0009      4507F6F6  00000000
                  8  0009      45080606  00000000
                  9  0009      45090606  00000000
                  A  0009      450A0000  00000000
                  B  0008      400B0000  00000000
                  C  0008      400C0000  00000000
                  D  0008      400D0000  00000000
                  E  0008      400E0000  00000000
                  F  0008      400F0000  00000000


  0E  00000001  NO PAGTABLE ENTRIES FOR THIS SEGMENT

  0F  00000001  NO PAGTABLE ENTRIES FOR THIS SEGMENT
```

Figure 36. Formatted VMFDUMP (Part 6 of 6)

Index

¢ (cent symbol), line edit use 23

! (attention symbol) 26
  sample usage 31

(pound symbol), line edit use 23

@ (At sign), line edit use 23

" (double quotes), line edit use 23

A
abnormal termination (abend) (*see also* VMFDUMP)
  dump 106
access, virtual machine requirements 11
address
  alteration
    console 24,28
areas, error recording (*see* error recording cylinders)
assistance, system operator 13
At sign (@), line edit use 23
ATTACH, command, usage 24
attached processor
  operation, results of uncorrectable errors 69
  summary of machine check handler action 68
  system damage 65
attention
  signaling, sample usage 26
  symbol (!)
    sample usage 26,31

C
CCH (channel check handler)
  additional functions 72
  error messages 74
  function 66
  initialization 72
  overview 60
  reaction to errors 74
  summary 71,73
CE's
  privilege class 14
  /system operator, relationship 13
  use of IPCS 13
  virtual machine
    capabilities/limitations 3,5
    protective features 3
    typical configuration 14
cent symbol (¢), line edit use 23
channel check
  error record layout 56
  handling by SVC 76 41
  reflection to virtual machine 60
  system action 72
channel check handler (*see* CCH (channel check handler))
checks, terminal facility 22
class, privilege 16
CMD command, RSCS usage 103
CMS (Conversational Monitor System)
  prerequisite for CPEREP 16
  warning, file destruction 17
codes
  line transmission 19
  wait state 74
conditions, terminal communication line 22
console functions, systems, CP command equivalency 3
control block linkage
  environmental data recording 51
  fatal error 50
  I/O operation 44
  I/O retry 46
  SDR recording 47
  structure for sense byte analysis 45
  2305 environmental data recording 50
  3330/3340/3350 environmental data recording 51
control units, line 18
correspondence (line transmission code) 19
CP commands, equivalency to system console functions 3
CP nucleus, storage errors 62
CPEREP (*see also* EREP)
  ACCDEV 95
  ACCIN 95
  applications 96
  brief description of use 16,64
  CMS the environment for 16
  command entry
    file entry method 92
    mixed entry method 92
    prompting method 91

  - command format 83-84
  - consolidation of error recording from different systems 97
  - DIRECTWK 95
  - duplication of VS EREP's IFCOFFLD (offload) function 96
  - edit facilities 16,64
  - EREPPT 94
  - FILEDEFs 94
  - invoked, console entry methods 91
  - operand entry, rules 91
  - operands, brief descriptions 85-89
  - OS/VS EREP overview 80
  - OS/VS EREP relationship 79
  - publication requirements for use 79
  - screening of operands 80
  - SERLOG 95
  - shared I/O configuration changes 97
  - SYSIN 94
  - terminal session, anotated console listing 92
  - TOURIST 95
  - type of error records recorded 81
  - use with MSS error records 16
  - use with VS1/VS2 Subsystem Data Analyzer (SDA) Program 16
  - user classes 82
  - vs OS/EREP record formats 80

- CP-initiated I/O operation, error recovery 63
- CP-owned volumes, linking to for test purposes 24
- cylinders (areas), error recording 48

D

- damage to system, recovery attempts 64
- DASD (Direct Access Storage Device)
  - environmental data recording, sense data 50
  - error recording conditions 53
- DASD device, testing 24
- data, security 12
- DEFINE, command (CP), usage 24
- destruction, file 12
- devices, supported, line equipment 18
- diagnostic tests (*see* OLTSEP (Online Test Standalone Executive Program))
- differences/exceptions, OS/VS EREP vs VM/370 EREP 82
- double error recording 64
- dump
  - device specified by SET command 106
  - system 106

E

- ECHO command
  - sample printout 34
  - used for terminal checkout 34
- editing
  - error records, CPEREP 64
  - input line 23
- EREP (*see also* CPEREP)
  - CPEREP equivalency 16,64
  - CPEREP relationship 79
  - data set requirements 93
  - reports, operand requirements 91
- error checking and correction (ECC) 70
- error handling overview 37
- error messages, to operator 44
- error recording
  - conditions, specific devices 53
  - CP modules used 43
  - cylinders (areas)
    - editing error records 64
    - full condition 48
    - virtual machine owned 38
  - dedicated devices 39
  - differences 39
  - edit facilities 16,64
  - functions 48
  - outboard recordings 48
  - record format 53
  - relationship of I/O configuration to DMKRIC and MSC Table Create Program 39
  - SVC 76 63
  - types of errors 38
  - virtual vs real addressing 39
  - virtual vs real device type 39
- error recovery
  - CP-initiated I/O operations 63
  - features, introduction 60
  - from soft machine checks 62
  - functional 67
  - I/O, detailed description 44
  - levels 67
  - machine check, hard 61
  - modes 70
  - operator-initiated restart 68
  - procedures 63
  - processor errors 61
  - processor retry 37,67
  - protection key errors 62
  - storage error 61
  - system 67
  - system repair 68
  - user termination 67
  - virtual machine initiated I/O operations 63

errors
   channel check
      handling by SVC 76 41
      reflection to virtual machine 60
      system action 72
   correction code (see error checking and
    correction (ECC))
   I/O, discussion 37
   machine check, system action 67
   messages, CCH, a referral 74
   record fields, source of data 57
   record formats
      channel checks 56
      header layout 58
      machine checks 56
      nonstandard 55
      unit check (short form) 55
   record layout, unit check 55
   record modifications, SVC 76 42
   reflection 37
   types recorded 38
EXTERNAL, command, example of use 28

F
F privilege class 16
failure (see also errors)
   line 22
   logon 23
fatal error, control block linkage 50
FILEDEFs for CPEREP 94
files
   destruction 17
   protection 12
   security 12
formats, error record (see record,
 formats)
frames, SRF (Service Record File),
 description 48
FRIEND
   OLTS 28
   sample printout 28

G
G privilege class 14

H
hardware maintenance
   virtual machine, overview 1
   VM/370 essential requirements 9
hardware problem analysis
   from a queued system task,
    advantages/disadvantages 2
   from the dedicated real system,
    advantages/disadvantages 1
   from the virtual machine,
    advantages/disadvantages 3
header
   layout, error record 58
   record, error, source of data 59

I
inboard error recording (see Recovery
 Management Support (RMS))
input, line editing 23
intensive recording mode 14,52,98
I/C
   devices, specifying, error recording 98
   environmental data recording 50
      2305 control block linkage 51
      3330/3340/3350 control block linkage
       51
   error recording
      permanent error 50
      structure for sense byte analysis 45
   errors (see also hardware problem
    analysis)
      control blocks linkage I/O retry 46
      DASD error conditions 53
      discussion 37
      intensive recording 52
      maintenance from a virtual machine,
       statistical evaluation 4
      message to operator 44
      reccvery, detailed description 44
      SDR recording 49
   operations
      control block linkage 44
      CP 37
      virtual machine 37
   testing 23,28,30
      minidisk 24
      terminals 25
IPCS (Interactive Problem Control System),
 CE usage 13

L
line
   devices supported by VM/370 18
   editing 23
   terminal facility check 22
   transmission
      codes 18
      tables 18
line control units 18
line delete, logical edit symbol 23
LINK command, use for testing CP-owned volumes 24
log records, SML 104
logging errors, determination for error recording 38
logical line delete 23
logical line end, edit symbol 23
logon
   a prerequisite for testing 21
   correspondence versus EBCD/PTTC codes 20
   failure 22
      messages 23
   successful 22
logout, storage assignment 75

M
machine check
   error record layout 56
   hard, recovery 61
   soft, error recovery 62
machine check handler (see MCH (machine check handler))
malfunction (see errors)
MCH (machine check handler)
   description 66
   function 66
   overview 60,66
   reaction to error 67
   summary 68
   with attached processor application 68
MESSAGE command
   sample printout 33
   use as an aid to logon 22
   used for terminal checkout 32
messages
   error, CCH, a referral 74
   to operator 44
minidisk testing 24
mode
   intensive recording 14
   system recovery (see recovery, mode)
modifications, error record, SVC 76 42
MONITOR command, trace data 106
MSG command (CP) (see MESSAGE command)

N
NETWORK command, TRACE operand, brief description 103
nonstandard record layout, error record layout 55
notional conventions, brief description 82

O
OLTS (Online Test Sections) 25
   example of printout 25
   invoking 22,24
   maintaining 9
   test runs from the virtual machine 4
   testing the virtual console 27
   virtual machine vs standalone system environment, test results analysis 4
OLTSEP (Online Test Standalone Executive Program)
   initialization 11,22
   maintaining 9
   OLTS (see OLTS (Online Test Sections))
OLTSEP-RETAIN, invoking 30
OLTS/FRIEND 28
   sample printout 28
   testing, operator assistance 28
Online Test Sections (see OLTS (Online Test Sections)
Online Test Standalone Executive Program (see OLTSEP (Online Test Standalone Executive Program))
operands
   CPEREP
      brief description 85-89
      required for reports 91
   CPEREP format 83-84
operating system, recognition by SVC 76 41
operator (see system operator)
OS/VS EREP (see EREP)

P
parameters, passing 41
permanent, I/O error, error recording 50
pound symbol (#), line edit use 23
prerequisites, OLTS 21
printout
   sample of ECHO 34
   sample of FRIEND 28
   sample of MESSAGE usage 33
   sample of OLTS 25
   sample of RETAIN . 31
privilege classes, CE's 16
problem analysis (see hardware problem analysis)
   from the virtual machine 3

- rocessor errors, VM/370 recovery 61
- rocessor retry 37,67
- rotection key errors, error recovery 62

- UERY, command (CP), example of use 28
- uiet recording mode 71

- eal machine vs virtual machine, hardware maintenance 1
- ecording, intensive mode 52
- ecording modes (see also SET command (CP) RECORD operand)
- ecording of error records, type recorded, VM/370 vs OS/VS2 81
- ecords, format, error recording 53
- ecovery (see error recovery)
- Recovery Management Support (RMS)
  - damage assessment 68
  - objectives 66
  - summary of functions 66
  - uncorrectable errors, machine check 69
  - VM/370 support 66
- relationship, CE/system operator 13
- requirements, for testing virtual machine 11
- reset, intensive recording 99
- restart
  - after system damage 64
  - operator-initiated 68
- RETAIN
  - procedures, invoking 30
  - sample printout 31
- retry
  - processor 37,67
  - via SET MODE command 99
- RMS (see Recovery Management Support)
- RSCS (Remote Spooling Communications Subsystem), tracing the line 103
- rules, CPEREP operand entry 91

S

- SDR (Statistical Data Recorder)
  - record layout 55
  - VM/370 usage 47
- SDR recording, initiated by SHUTDOWN 49
- security, data (file) 12
- sense data, DASD environmental recording 50
- Service Record File (see SRF)
- SET command (CP)
  - MODE operand
    - description 99
    - threshhold count 100
    - usage 62,70
    - use 99
  - RECORD operand
    - description 98
    - examples 99
    - usage 52
- SML (Spool MULTI-LEAVING), log records 104
- soft errors
  - count control 70
  - explanation 68
  - limiting 70
  - recording at system initialization 70
- SRF (Service Record File)
  - access to 48
  - frames, description 48
- starter system 21
- storage, assignments, logout 75
- storage errors
  - CP nucleus 62
  - system recovery 61
- STORE, command, example of use 28
- supported systems, SVC 76 40
- SVC 76
  - description 40
  - error record modifications 42
    - type DDR 43
    - type MDR 43
    - type MIH 43
    - type OBR 42
    - type program abend 42
  - error recording 63
    - requirements 38
  - functions 40
  - handling of channel errors 41
  - operating system recognition 41
  - parameter passing 41
  - systems support 40
- system
  - configuration
    - minimum 21
    - starter 21
  - dump 106
  - repair 68
  - termination 67
- system console functions, CP command equivalency 3
- system damage
  - attached processor recovery 65
  - attached processors, affinity reset 65
  - system restart facilities 64

- system operator
  - aid from 13
  - assistance 13
  - /CE relationship 13
- SYS1.LOGREC (*see* error recording)

T

- terminals
  - address alterations 24,28
  - check
    - via ECHO 34
    - via MESSAGE 32
  - supported by VM/370 18
  - transmission codes 18
- termination, system operation (*see* fatal errors)
- test(s)
  - diagnostics (*see* OLTSEP (Online Test Standalone Executive Program))
  - line transmission code 19
  - MESSAGE command 23
  - minidisk 24
  - system check, basic 22
- testing, from a virtual machine 11
- threshhold count, SET MODE 100
- trace
  - NETWORK command, brief description 103
  - RSCS line 103
- TRACE command (CP) (*see also* NETWORK command TRACE operand)
  - described 101
  - invoking, examples 101
  - output 101
  - printout segment 102
- trace table, MONITOR command 106

U

- uncorrectable errors
  - machine check
    - attached processor action 69
    - uniprocessor action 69
  - system action 69
- unit check
  - (*see also* error handling)
  - error record layout 55
  - error record layout (short form) 55
  - error recordings for 2305, 3330, 3340, 3350 53
- user
  - terminals 18
  - termination, error recovery 67
- user class required, CPEREP 82

V

- virtual console, testing 26
- virtual machine
  - a tool for I/O problem analysis, statistical evaluation 4
  - as troubleshooting aid 1
  - CE's
    - capabilities/limitations 3,5
    - protective features 3
  - error recording 39
  - error recording cylinders (areas) 38
  - I/O error recovery 63
  - results of uncorrectable errors 69
  - the CE's 14
- Virtual Machine Facility/370 (VM/370)
  - error recording, differences 39
  - online message, 3704/3705 22
  - recovery features
    - channel check handler 60
    - machine check handler 60
  - support, RMS 66
  - system, recovery 67
- VMFDUMP command
  - description 106
  - sample of output 108

W

- wait state codes 74

3

- 3277, OLTS testing 26
- 3704/3705, VM/370 online message 22

READER'S COMMENT FORM

Title: IBM Virtual Machine Facility/370: OLTSEP and Error Recording Guide

Order No. GC20-1809-7

Please check or fill in the items; adding explanations/comments in the space provided.

Which of the following terms best describes your job?

□ Customer Engineer
□ Engineer
□ Instructor
□ Manager
□ Mathematician
□ Operator
□ Programmer
□ Sales Representative
□ Student/Trainee
□ Systems Analyst
□ Systems Engineer
□ Other (explain below)

How did you use this publication?

□ Introductory text □ Reference manual □ Student/ □ Instructor text
□ Other (explain) ______________________________

Did you find the material easy to read and understand? □ Yes □ No (explain below)

Did you find the material organized for convenient use? □ Yes □ No (explain below)

Specific criticisms (explain below)

Clarifications on pages ______________________________
Additions on pages ______________________________
Deletions on pages ______________________________
Errors on pages ______________________________

Explanations and other comments:

Thank you for your cooperation. No postage necessary if mailed in the U.S.A.

GC20-1809-7

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 40 ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE:

International Business Machines Corporation
Department D58, Building 706-2
PO Box 390
Poughkeepsie, New York 12602

Attn: VM/370 Publications

Fold and tape   Please Do Not Staple   Fold and tape

IBM VM/370 OLTSEP & Error Recording Guide   Printed in U.S.A.   GC20-1809-7

International Business Machines Corporation
Data Processing Division
1133 Westchester Avenue, White Plains, N.Y. 10604

IBM World Trade Americas/Far East Corporation
Town of Mount Pleasant, Route 9, North Tarrytown, N.Y., U.S.A. 10591

IBM World Trade Europe/Middle East/Africa Corporation
360 Hamilton Avenue, White Plains, N.Y., U.S.A. 10601

GC20-1809-7

IBM VM/370 OLTSEP & Error Recording Guide

Printed in U.S.A. GC20-1809-7

International Business Machines Corporation
Data Processing Division
1133 Westchester Avenue, White Plains, N.Y. 10604

IBM World Trade Americas/Far East Corporation
Town of Mount Pleasant, Route 9, North Tarrytown, N.Y., U.S.A. 10591

IBM World Trade Europe/Middle East/Africa Corporation
360 Hamilton Avenue, White Plains, N.Y., U.S.A. 10601