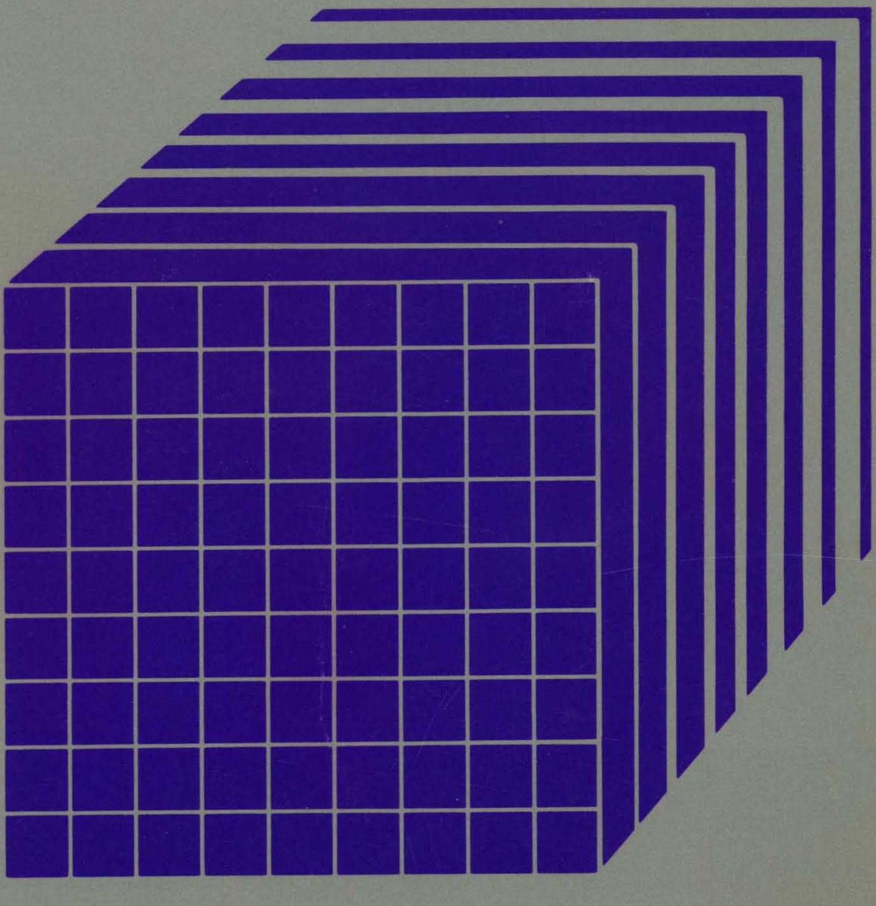
IBM

# Virtual Machine/ System Product

# OLTSEP and Error Recording Guide

Release 4

SC19-6205-3



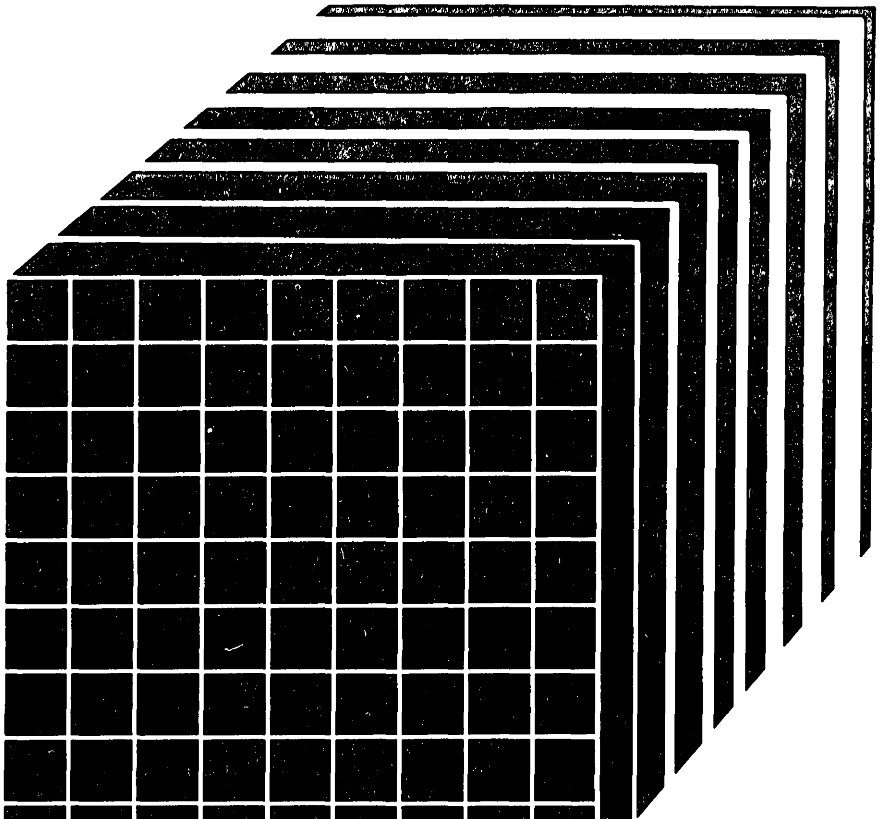
IBM

# Virtual Machine/ System Product

# OLTSEP and Error Recording Guide

Release 4

SC19-6205-3



**Notice:** The term VM/SP, as used in this publication, refers to VM/SP when used in conjunction with VM/370 Release 6.

## Fourth Edition (December 1984)

This edition, SC19-6205-3, is a major revision of SC19-6205-2, and applies to Release 4 of Virtual Machine/System Product (VM/SP), program number 5664-167, and to all subsequent releases and modifications until otherwise indicated in new editions or Technical Newsletters. Changes are made periodically to the information herein; before using this publication in connection with the operation of IBM systems, consult the latest *IBM System/370 and 4300 Processors Bibliography*, GC20-0001, for the editions that are applicable and current.

**Summary of Changes**

For a detailed list of changes, see page iii.

Changes or additions to the text and illustrations are indicated by a vertical line to the left of the change.

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

Publications are not stocked at the address given below. Requests for 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, Programming Publications, Dept. G60, P.O. Box 6, Endicott, NY, U.S.A. 13760. IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you.

# Summary of Changes

To obtain editions of this publication that pertain to earlier releases of VM/SP, you must order using the pseudo-number assigned to the respective edition. For:

Release 3, order ST00-1354
Release 2, order SQ19-6205
Release 1, order ST19-6205

**Summary of Changes**
**for SC19-6205-3**
**for VM/SP Release 4**

**New**

The base VM/IPCS component includes function equivalent to that within the VM/IPCS Extension licensed program.

The IBM 3480 Magnetic Tape Subsystem is supported.

The SAVESYS, VMSAVE, and IPL functions allow a page image copy of up to 16 Meg of virtual storage to be saved and restored.

Users may define up to 32 privilege classes.

**Changed**

The information about VM/SP CPEREP and EREP is now contained in *EREP User's Guide and Reference*, GC28-1378.

Minor technical and editorial changes have been made throughout this publication.

**Summary of Changes**
**for SC19-6205-2**
**for VM/SP Release 3**

**PER Command**

*New:* The CP PER command is added with VM/SP Release 3. This command is used to monitor certain events as they occur during program execution in the user's virtual machine.

*New:* Appendix A is added to provide an example for using EREP with VM/SP.

### EREP Version 2, Release 2

*Changed:* The "VM/SP CPEREP And EREP" portion of Section 4 is updated to reflect any changes to the EREP information as a result of the changes within EREP Version 2, Release 2.

### Miscellaneous:

*Changed:* Minor technical and editorial changes have been made throughout this publication.

## Summary of Changes
## for SC19-6205-1
## as Updated by SN24-5738

### Missing Interrupt Handler

*New:* The missing interrupt detector has been extended so that CP not only detects missing interrupt conditions, but also attempts to correct them. CP informs the system operator whether or not corrective action was successful.

To give the user optimum system availability, the missing interrupt handler allows you to vary the time interval allowed for I/O completion for the supported devices.

## Summary of Changes
## for SC19-6205-0
## as Updated by SC19-6205
## for VM/SP Release 2

### VM/SP Support for IBM 3375 Direct Access Storage Device

*New:* VM/SP support for the 3375 is added. VM/SP supports the IBM 3375 as a spooling, paging, and system residence device. T-disk, mini-disk, and dedicated support is also provided.

# Preface

This publication is intended for the IBM customer engineer (CE) or any system support personnel familiar with OLTS testing procedures. Note that wherever CE is used, system support personnel is implied. It is also assumed that the CE is knowledgeable about VM/SP and virtual machine concepts as outlined in the *VM/SP Introduction*. The CE should know the VM/SP logon process described in *VM/SP Terminal Reference.*

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 System/370 and the VM/SP is also discussed.

Section 2 discusses the requirements for testing I/O devices from a virtual machine environment that 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/SP 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/SP facilities that allow more detailed information to be obtained for problem analysis and repair. These include:

- CPEREP and OS/VS EREP
- Intensive Recording Mode
- Trace Option
- Program Event Recording
- IPCSDUMP/PRTDUMP.

Appendix A contains an example EXEC program for using EREP with VM. The EXEC can be used to:

- "Emergency offload" of the Error Recording Area (ERA) onto a tape.
- Clear/reset the ERA.
- Generate some reports from the EREP History File.

## Prerequisite Publications

*Virtual Machine/System Product:*

*Introduction*, GC19-6200

*Operator's Guide*, GC19-6206

If the IBM 3767 Terminal is used, *IBM 3767 Operator's Guide*, GA18-2000, is also prerequisite.

The base VM/SP IPCS component now includes function equivalent to that within the VM/IPCS Extension licensed program. Details of this component are found in the *VM/SP Interactive Problem Control System Guide*, SC24-5260.

## Corequisite Publications

*Virtual Machine/System Product:*

*CMS Command and Macro Reference*, GC19-6209

*CP Command Reference for General Users*, SC19-6211

*Running Guest Operating Systems*, SC19-6212

*System Messages and Codes*, SC19-6204

*Data Areas and Control Block Logic Volume 1 (CP)*, LY24-5220

*Data Areas and Control Block Logic Volume 2 (CMS)*, LY24-5221

*EREP User's Guide and Reference*, GC28-1378

## Related Publications

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

*Virtual Machine/System Product:*

*Planning Guide and Reference*, SC19-6201

*Installation Guide*, SC24-5237

*CMS User's Guide*, SC19-6210

*Operator's Guide*, SC19-6202

*System Product Editor User's Guide*, SC24-5220

*System Product Editor Command and Macro Reference*, SC24-5221

*EXEC 2 Reference*, SC24-5219

*Remote Spooling Communication System Networking Operation and Use*, GC24-5058

*3704, 3705 NCP/VS Version 2 Logic*, SY30-3007.

## Supplemental Publications

*Virtual Machine/System Product:*

*Quick Reference*, SX20-4400

*Commands (General User) Reference Summary*, SX20-4401

*Commands (Other than General User) Reference Summary*, SX20-4402

*SP Editor Command Language Reference Summary*, SX24-5122

*EXEC 2 Language Reference Summary*, SX24-5124

*System Product Interpreter Language Reference Summary*, SX24-5126

*Note:* If all you want all the supplemental publications, use Order No. SBOF3820.

## Referenced Publications

*IBM System/370 Principles of Operation*, GA22-7000

*OS/VS1 SYS1.LOGREC Error Recording*, GC28-0668

*OS/VS2 System Programming Ligrary: SYS1.LOGREC Error Recording For MVS*, GC28-0677

*OS/VS Mass Storage Control Table Create*, GC35-0013

## Terminology

In this publication, the terms used are as follows:

- "2741" applies to both the IBM 2741 and the IBM 3767 Communication Terminals unless otherwise specified.
- "3081" refers to the IBM 3081 processor.
- "3262" refers to the IBM 3262 Printer.
- "3270" encompasses the IBM 3275, 3276, 3277, 3278, and 3279 Display Stations. Note that a specific device type is used only when a distinction is required between device types.
- "3330 series" is used to mean the IBM 3330 Disk Storage, Models 1, 2, and 11; the IBM 3333 Disk Storage and Control, Models 1 and 11; and the IBM 3350 Direct Access Storage operating in 3330/3333 Models 1 and 11 compatibility mode.

- "3370" refers to the 3370 Direct Access Storage Device, Models A1, A2, B1, and B2.

- "3375" refers to the 3375 Direct Access Storage Device.

- "3380" refers to the IBM 3380 Storage Facility.

  If the 3380 attached to the 3880 Controller Model 3 with Speed Matching Buffer (Feature #6550) is part of your installation, CP will permit execution of the Extended Count Key Data Channel programs. Additional details can be found in the *VM/SP Planning Guide and Reference* and *VM/SP Installation Guide.*

- "3480" refers to the IBM 3480 Magnetic Tape Subsystem.

- "3880" refers to the IBM 3880 Storage Control Unit.

- "FB-512" refers to those IBM DASD units that implement fixed block (512-byte blocks) architecture, which includes the IBM 3310 and 3370 devices.

- "System/370" applies to 4300 series processors and the 303x series processor, unless otherwise noted.

- "Block" is used to describe DASD space on FB-512 devices. Note that FB-512 devices comprise the IBM 3310 and IBM 3370 Direct Access Devices employing fixed-block mode.

- "Cylinder" is used to describe DASD space on count-key-data devices supported by the VM/SP System Control Program.

- "DASD space" is used when there is no need to differentiate between count-key-data devices and FB-512 devices.

- "VSE" refers to the combination of the DOS/VSE system control program and the VSE/Advanced Functions program product. In certain cases, the term DOS is still used as a generic term; for example, disk packs initialized for use with VSE or any predecessor DOS or DOS/VS system may be referred to as DOS disks. Note that the DOS-like simulation environment provided under the CMS component of VM/SP continues to be referred to as CMS/DOS.

- Display terminal usage information also applies to the IBM 3036, 3138, 3148, and 3158 Display Consoles when used in display mode, unless otherwise noted.

- Printer information pertaining to the IBM 3284 and IBM 3286 Printers also pertains to the IBM 3287, 3288, and 3289 Printers, unless otherwise noted.

- Discussion about attached processor configurations is also applicable to multiprocessor configurations unless otherwise noted.

*Notes:*

1. *External interrupt reflection may cause OLTSEP Release 4.0, 4.1, or 5.0 execution problems. How to circumvent these problems is discussed under "Invoking OLTS" on page 2-12.*

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 to use the OLTTEP/OLLT/OLTT diagnostic package. If these test facilities are to be invoked, they must be used with VS with TCAM in a standalone System/370.*

3. *The privilege classes (A-G) mentioned in this book refer to the IBM-defined classes. If the installation overrides any of the IBM-defined privilege classes, consult the installation's administrator for any new authorizations and restrictions.*

# The VM/SP Library

## Evaluation

| | |
|---|---|
| GENERAL INFORMATION | GC20-1838 |
| INTRODUCTION | GC19-6200 |

## Index

| | |
|---|---|
| LIBRARY GUIDE, GLOSSARY, AND MASTER INDEX | GC19-6207 |

## Planning

| | |
|---|---|
| PLANNING GUIDE AND REFERENCE | SC19-6201 |
| RUNNING GUEST OPERATING SYSTEMS | GC19-6212 |
| DISTRIBUTED DATA PROCESSING GUIDE | SC24-5241 |
| RELEASE 4 GUIDE | SC24-5248 |

## Installation

| | |
|---|---|
| INSTALLATION GUIDE | SC24-5237 |
| SYSTEM DEFINITION FILES | SC24-5256 |

## Administration

| | |
|---|---|
| SYSTEM PROGRAMMER'S GUIDE | SC19-6203 |
| GCS GUIDE | SC24-5249 |
| GCS MACRO REFERENCE | SC24-5250 |

## Operation

| | |
|---|---|
| OPERATOR'S GUIDE | SC19-6202 |

## Applications

| | |
|---|---|
| APPLICATION DEVELOPMENT GUIDE | SC24-5247 |

## End Use

| | |
|---|---|
| TERMINAL REFERENCE | GC19-6206 |
| CMS PRIMER | SC24-5236 |
| CMS PRIMER FOR LINE-ORIENTED TERMINALS | SC24-5242 |
| CMS USER'S GUIDE | SC19-6210 |
| CMS COMMAND AND MACRO REFERENCE | SC19-6209 |
| SP EDITOR USER'S GUIDE | SC24-5220 |
| SP EDITOR COMMAND AND MACRO REFERENCE | SC24-5221 |
| CP COMMAND REFERENCE | SC19-6211 |
| SP INTERPRETER USER'S GUIDE | SC24-5238 |
| SP INTERPRETER REFERENCE | SC24-5239 |
| EXEC 2 REFERENCE | SC24-5219 |

## Reference Summaries

To order all the Reference Summaries, use order number SBOF 3221

| | |
|---|---|
| QUICK REFERENCE | SX20-4400 |
| COMMANDS (GENERAL USER) | SX20-4401 |
| COMMANDS (Other than General User) | SX20-4402 |
| SP EDITOR COMMAND REFERENCE SUMMARY | SX24-5122 |
| SP INTERPRETER REFERENCE SUMMARY | SX24-5126 |
| IPCS REFERENCE SUMMARY | SX24-5138 |
| EXEC 2 REFERENCE SUMMARY | SX24-5124 |

## Diagnosis

- SYSTEM MESSAGES AND CODES SC19-6204
- SYSTEM MESSAGES CROSS-REFERENCE SC24-5284
- OLTSEP AND ERROR RECORDING GUIDE SC19-6205
- IPCS GUIDE SC24-5260
- SERVICE ROUTINES PROGRAM LOGIC LY20-0890
- PROBLEM DETERMINATION VOL. 1 (CP) LY20-0892
- DATA AREAS AND CONTROL BLOCKS VOL. 1 (CP) LY24-5220
- PROBLEM DETERMINATION VOL. 2 (CMS) LY20-0893
- DATA AREAS AND CONTROL BLOCKS VOL. 2 (CMS) LY24-5221

## Auxiliary Service Support

Device Support Facilities
5748-XX9

- DEVICE SUPPORT FACILITIES USER'S GUIDE AND REFERENCE GC35-0033

Environmental Recording
Editing and Printing
(EREP)

- EREP USER'S GUIDE AND REFERENCE GC28-1378

## Auxiliary Communication Support

RSCS Networking
Version 2
5664-188

- RSCS NETWORKING VERSION 2 GENERAL INFORMATION GH24-5055
- RSCS NETWORKING VERSION 2 PLANNING AND INSTALLATION SH24-5057
- RSCS NETWORKING VERSION 2 OPERATION AND USE SH24-5058
- RSCS NETWORKING VERSION 2 DIAGNOSIS REFERENCE LY24-5228
- RSCS NETWORKING VERSION 2 REFERENCE SUMMARY SX24-5135

Advanced
Communication
Function
For VTAM
(ACF/VTAM)
5664-280

- VTAM GENERAL INFORMATION FOR VM/SP GC30-3246
- VTAM INSTALLATION AND RESOURCE DEFINITION SC23-0111
- VTAM MESSAGES AND CODES FOR VM/SP SC30-3275
- VTAM DATA AREAS FOR VM/SP SC30-3249
- VTAM DIAGNOSIS GUIDE SC23-0116
- VTAM DIAGNOSIS REFERENCE SC23-0117

VM/Pass-
Through
Facility
5748-RC1

- VM/PASS-THROUGH FACILITY GENERAL INFORMATION GC24-5206
- VM/PASS-THROUGH FACILITY GUIDE AND REFERENCE SC24-5208
- VM/PASS-THROUGH FACILITY LOGIC LY24-5208

# Contents

# Figures

# Section 1. Hardware Maintenance--Real Machine System/370 Versus VM/SP

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; therefore, the system must be used to affect 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 for 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 and/or storage malfunctions or suspected device malfunctions, when established local and offline troubleshooting techniques fail, is best achieved from an environment that is totally and exclusively under the control of the Customer Engineer. Total control of a system and its resources excludes the use of 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 unique or radical, can be undertaken because there is no risk of destroying the customer's programs and data sets. However, this ideal method of 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 on weekends or on 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:

- 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 on 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: OLTEP) 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 priority in relation to running 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, no provision is made 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 by 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/SP Terminal Reference*) and solicited a minimum amount of assistance from 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 is 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/SP. Protective features of the VM/SP 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 CE's virtual machine have negligible effect on other System/370 operations. As an alternative to having the power of a real System/370 at the CE's 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 and device applications and processes that are not supported by VM/SP; these are detailed in the appendix about "VM/SP Restrictions" in the *VM/SP Planning Guide and Reference.*

The facilities provided by VM/SP and the virtual machines it supports are described in the *VM/SP 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 and/or 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.

Some of the functions that can be simulated for the virtual machine by use of commands are given in Figure 1-1 on page 1-4.

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 given in an appendix in the *VM/SP CP Command Reference for General Users.*

| Command | Function |
|---|---|
| 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 |

**Figure 1-1. Keyed in "Console Function" Commands and Their Functions**

Commands that are available to the general user, as well as the CE, are described in the *VM/SP CP Command Reference 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/SP 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/SP'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 OLTS that violated VM/SP architecture (see the appendix about "VM/SP Restrictions" in the *VM/SP Planning Guide and Reference*) 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 OLTS from a virtual machine.

The tabulated results were also indicative of the fluid nature of 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 |
| 2305 | 2 |
| 2318 | 1 |
| 2319 | A0 |
| 2400 | 5 |
| 2540 | 1 |
| 2703 | 1 |
| 2821 | 1 |
| 2835 | 2 |
| 2803 | 2 |
| 3145 | 1 |
| 3215 | 1 |
| 3330 | 1 |
| 3803 | 1 |
| 3830 | 1 |

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

*Note:* *OLTS tabulations as a result of RETAIN/370 and the 2955 interface are identical to the results obtained by the site CE invoking the tests (see "OLTSEP-RETAIN/370" on page 2-18).*

None of the tests executed in the VM/SP virtual machine environment resulted in a hang, reset, or loop condition of the virtual machine, nor was there any perceivable effect on the operations and security of VM/SP and other associated virtual machines.

## Points for the CE to Consider about Virtual Machine Use

As stated previously, the *VM/SP 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 *VM/SP Running Guest Operating Systems.*

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 appendix about "VM/SP Restrictions" in the *VM/SP Planning Guide and Reference* should be consulted to see whether or not the malfunction or suspected malfunction is a violation of VM/SP architecture. Certain OLTS diagnostics violate VM/SP 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, a minidisk.

- All system errors and I/O errors are not written out to the VM/SP error recording area; consult Section 3, "Error Handling" on page 3-1 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/SP itself does not generate EOD and IPL records. No error recordings of these types are accepted for the VM/SP system or any other virtual systems. Certain other error types are also not processed.

- Most CCWs and CCW chains are subject to VM/SP control program modifications so that VM/SP can 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/SP 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/SP Introduction.*

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

- 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 thorough analysis of a machine's malfunction.

- In attempts to service components of a 3850 Mass Storage System (MSS), 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 ensure a quiesced mass storage system, it is recommended that the CE test programs 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/SP virtual machine operations and MSS errors are reflected to VM/SP'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 hour 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 sequence of output files when the need is urgent.

- The I/O configuration of the virtual machine should be such that each virtual channel maps to a real channel of a single type and model. This requirement is explained in detail in the appendix about "VM/SP Restrictions" in the *VM/SP Planning Guide and Reference*. 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.

- Hardware and software problems in VM/SP can cause the abnormal termination (abend) of a virtual machine. These terminations (such as forced logoff) cause the register and storage contents of the virtual machine to be lost, thus rendering problem analysis ineffective. With the use of the VMSAVE and SAVESYS functions, you can save a page image copy of up to a 16 Meg virtual machine (including its register contents, PSW, and storage keys) on preallocated DASD space, thereby making system analysis and system recovery possible. For details refer to "Virtual Storage Preservation" in *VM/SP System Programmer's Guide* and "Saved System DASD Requirements" in *VM/SP Planning Guide and Reference.* The saving of virtual machines on a VMSAVE area can be adversely affected by malfunctions of the checkpoint, spooling, and abend dump modules as well as channel check handler (CCH) and machine check handler (MCH) modules.

# Section 2. VM/SP Maintenance Essentials

This section contains information about the following:

- 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
- Invoking OLTS
- OLTS-FRIEND
- OLTSEP-RETAIN/370
- Basic Terminal Check via the MESSAGE Command
- Basic Terminal Check via the ECHO Command.

VM/SP 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/SP 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/SP 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/SP 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 functioning.

4. The communication path to and from the CE's terminal is functioning and is in an enabled state.

5. The CE user virtual machine identification (userid) and password are known to the CE.

6. The device(s) to be tested must be available for the CE's exclusive use.

Note: *When even one 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/SP Planning Guide and Reference or the VM/SP Operator's Guide.*

Hardware maintenance encompasses the following major areas of a system complex: the main processing unit (and the attached processor, if applicable) and the 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/SP.

- The I/O equipment, however, can be maintained by using online test 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 not conform to the OS or DOS labeling techniques, nor do VM/SP 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/SP supervision could contain these protection features. Therefore, the CE must proceed cautiously, since tape and disk files encountered on a VM/SP 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/SP. The management of these facilities belongs, in part, to the system operator. The CE's virtual machine is also part of the system operator'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 of the system operator's responsibility to other users' virtual machines. The system operator, because of schedules and system workload, may not be able to 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' operations affect the throughput time of the CE's diagnostic operations.

The relationship between the system operator and the CE may enhance I/O maintenance. This can be done by having the system operator exercise system options within his control. Suppose, for example, a problem exists and the tag lines are suspect. Oscilloscope 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 of 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 of other user I/O activity. Note also that if the system operator has access to the problem report file system of the Interactive Problem Control System (IPCS) virtual machine, an initial and instant analysis for the current problem by comparison to a base of previously reported customer VM/SP problems can help determine whether the malfunction occurred in the hardware or was a software problem.

## The CE's Virtual Machine

Hardware I/O maintenance can be accomplished by having the CE operate his own virtual machines from a terminal device while permitting other VM/SP users to continue operating their own virtual machines. The CE's virtual machines are unique in that his CP command privilege class F allows him to clear the error recording area and to set intensive mode recording.

The virtual machines, accessed through a VM/SP terminal device, provide 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 programs of his choice without affecting other users.

In most instances, the CE needs no dedicated processor 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/SP CP Command Reference for General Users.*

Sample directories for typical virtual machines for the CE's use are defined in the sample directories provided with the product tapes. The sample directory listings can be found in the *VM/SP System Definition Files.* These sample directories may need to be modified to make them compatible with the installation. The sample directory for EREP is defined in the sample directory listing as USER EREP. The sample directory for OLTSEP is defined in the sample directory listing as USER OLTSEP.

## Command Privilege Class for the CE

The CE's virtual machine is similar to other virtual machines running under VM/SP. 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 IBM-defined privilege classes. (The installation has the option of expanding the privilege classes to 32.) 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/SP 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/SP 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/SP, 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 *RSCS Operation and Use.*

Note: *The RSCS Networking Version 2 program product (5664-188) is recommended for use with VM/SP.*

The use of CPEREP is also important in relation to its use with the 3850 Mass Storage System (MSS). Errors accumulated on the VM/SP error recording area must be placed in the CPEREP accumulator output tape for additional processing. Detailed instructions for using CPEREP are contained in *EREP User's Guide and Reference.*

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 "Using the CP SET Command Facilities" on page 4-3. 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 is also necessary for access to the CE area on FB-512 devices. The size and location of this area is described in the particular device reference publication.

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.

*Note:* *If the installation has redefined the privilege classes, the CE may be functioning under a new privilege class. You, the CE, should ensure that the installation authorized you to issue the diagnostic commands previously reserved for privilege class F.*

This book illustrates the use of only those VM/SP 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/SP CP Command Reference for General Users*, the *VM/SP Operator's Guide*, and the *VM/SP CMS Command and Macro Reference.*

*Note:* *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-cited 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 erasing or by overlaying. For a discussion of the CMS file management system, see *VM/SP CMS User's Guide.***

## Console Terminal Communications Considerations

A console terminal is used as a communications device between the user and the processor. Those devices supported by VM/SP can also be used as virtual machine console terminals. Some of those devices, however, need specific hardware features to facilitate VM/SP usage. For a complete list of devices supported by VM/SP and used as console terminals, refer to *VM/SP Planning Guide and Reference.*

VM/SP 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(s) |
|---|---|
| 2701 | 3271 Model 2 |
| 2702 | 3272 Model 2 |
| 2703 | 3274 Model 1B |
| 3704 | 3274 Models 1C, 1D, and 51C |
| 3705 | 3276 Models 2, 3, and 4 |
| 3725 | 3276 Models 2, 3, and 4 |

The *VM/SP Planning Guide and Reference* contains a list of the features necessary for each device to operate in the VM/SP environment.

VM/SP 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/SP supported devices listed in *VM/SP Planning Guide and Reference.* One exception is the 2741 unit, which uses either PTTC/EBCD or Correspondence code.

The supported terminal devices can be categorized as belonging to IBM Telegraph Terminal Control Unit Model 1/2.

For a list of the features and RPQs necessary for VM/SP usage of these terminals and consoles, consult the *VM/SP Planning Guide and Reference.*

VM/SP 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, 3380). 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/SP, 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 that device and the deciphering of the LOGON (or DIAL) command.

Determination of whether the device on the enabled line is a 1052 or a 2741 is handled by the initial communication sequence between VM/SP and the terminal; this is illustrated in Figure 2-1 on page 2-7. VM/SP handles the 3767 terminal and the 2741 terminal identically.

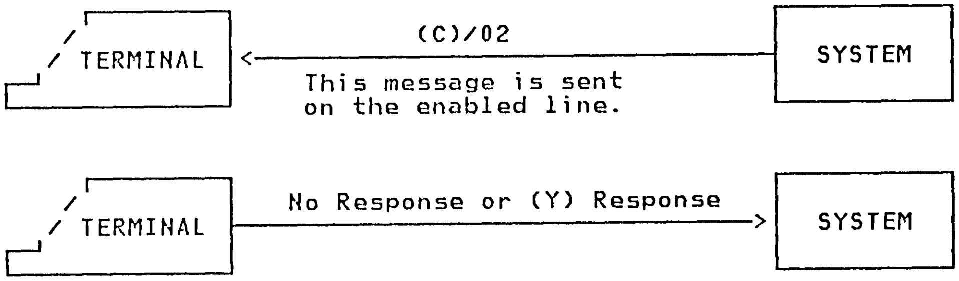


**Figure 2-1. VM/SP Terminal 1051 or 2741 Determination Procedure**

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/SP CP Command Reference for General Users*. One of these CP commands is the LOGON command shown in Figure 2-2.

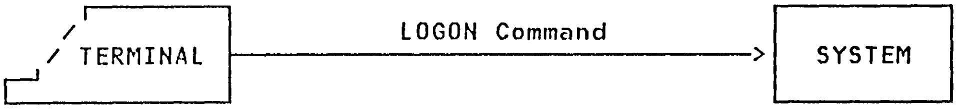


**Figure 2-2. Determining the Line Transmission Code for the 2741**

Code determination is done by the program examination of the LOGON 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.

The differences between the two codes when the LOGON command is entered from the terminal is shown in Figure 2-3.

| | | 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 2-3. 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 a storage area that contains a specific number of doublewords that describe the characteristics, and other data applicable to a designated device. 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/SP control program areas. In VM/SP, the 1052 and 3215 are architecturally and functionally equivalent. Therefore, the 1052 and 3215, along with 3210 and 2150, all equate to the same hexadecimal equivalent. This causes the output of the OBR summary records to reflect the device type as a 1052 rather than as a real device type.

# Conditions for Invoking Tests

## Processor Reliability

VM/SP, and the system on which it runs, must achieve a basic reliability for CE diagnostic use. This is achieved by hardware configurations that meet VM/SP criteria for system generation. Refer to *VM/SP Planning Guide and Reference* for a complete list of devices, model numbers, and features supported by VM/SP. The *VM/SP Installation Guide* gives sample hardware configurations that comprise the minimum system requirements for running VM/SP 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/SP 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 (AP) and multiprocessor (MP) operations, a serious malfunction on a processor could cause the system to convert to uniprocessor (UP) mode operation on the remaining processor.*

Basic VM/SP 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 or 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:* *Any user 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 logged onto his virtual machine.*

## Successful Logon

Successful logon indicates terminal and communication path reliability, CE virtual machine accessibility, and compliance with VM/SP logon requirements. Logon is successful if the terminal responses progress as far as the LOGON AT message.

**Example:**

```
logon erep
ENTER PASSWORD
password (you must supply; is not visible)

LOGON AT hh:mm:ss zzz day-of-week mm/dd/yy


VM/SP CMS - mm/dd/yy hh:mm
.
.
.
```

*Note:* *VM/SP will always type a mask for all users of typewriter-like terminals, as follows:*

```
ENTER PASSWORD:
XXXXXXXX
```

and will not advance the paper until you have typed in your password. If you make an error when typing your password, you must begin again with logging on.

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

- A 2741 terminal is connected to VM/SP via a 3704/3705 line in NCP (network control program) mode.

  *Note: VM/SP does not support NCP (Network Control Program) mode. VM/SP also does not support the MTA (Multiple Terminal Access) feature of NCP. However, if the VTAM Service Machine (VSM) component of SNA (Systems Network Architecture) is in place, the appropriate network control program will be in effect.*

- The user's terminal is connected to a 3704/3705 line that is in NCP mode and has the Multiple Terminal Access (MTA) feature.

  *Note: VM/SP does not support NCP (Network Control Program) mode. VM/SP also does not support the MTA (Multiple Terminal Access) feature of NCP. However, if the VTAM Service Machine (VSM) component of SNA (Systems Network Architecture) is in place, the appropriate network control program will be in effect.*

- 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/SP terminal entry rules have been violated.
- VM/SP program not operational.
- Successful logon would exceed current system operating parameters; therefore, it is not allowed.

*Note:* Additional information about terminal use and messages and responses that may be received can be found in the "VM/SP Log-on Procedures" section of the *VM/SP Terminal Reference.*

### Line and Terminal Facility Check

Data path failure or VM/SP 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/SP is indeed operational and if the terminal in question is online and enabled to the system. If the system operator response is affirmative, local testing and communication line checks should be initiated.

Terminal messages will be received if the VM/SP terminal logon procedures are not followed. Use the *VM/SP Terminal Reference* 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.

Several different messages or prompts may be received, depending on the conditions detected during logon. These messages should be self explanatory. (Additional information about terminal messages and responses that may be received can be found in the "VM/SP Log-on Procedures" section of the *VM/SP Terminal Reference.*) You may 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 operator message-text

msg op message-text

m op message-text
```

If a message response from the system operator is not forthcoming or the message cannot be entered via terminal equipment, then other media must be used to establish communication with the system operator.

If an acknowledgement 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 but the response was a facsimile of one of the following:

```
MAXIMUM USERS EXCEEDED
USERID MISSING OR INVALID
userid NOT IN CP DIRECTORY
PASSWORD INCORRECT
ALREADY LOGGED ON type raddr
```

CE action should be to relay this data via the MESSAGE command back to the system operator. The system operator can then defer maintenance to a later time period, or can arrange an environment so that CE LOGON 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/SP's four input line edit functions; they are described in the *VM/SP Terminal Reference.* The following table gives a brief explanation:

| Character | Function | Meaning |
|---|---|---|
| @ | Logical Character Delete | Preceding character is deleted. |
| # | Logical Line End | End of logical input line. Use of this symbol allows multiple entries on one line. |
| ¢ | Logical Line Delete | Previously entered data line is deleted. |
| " | Logical Escape | Escape character must precede any of the line edit function characters in order for them 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 to 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 on logon, the initial program load (IPL) operation, and the virtual logoff process (ending the terminal session) are described in *VM/SP Terminal Reference.*

## 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 personally, or, if the CE is at a remote location, communicating via messages as follows:

```
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 for 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 units) 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 factor of 0 (that is, the DASD unit is described in the system with its minimum and maximum cylinder or block 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 command.

The CE is now ready to load his virtual machine with OLTSEP. This is done by issuing the IPL command to the addressed device. Upon completion of the operation, OLTSEP responds to the CE's terminal as though he were using the real system console (3215, etc.). Figure 2-4 on page 2-14 shows a sample of the complete logon operation, OLTS testing, and logoff operation. The sample session shown in Figure 2-4 on page 2-14 would suffice for diagnostics run from a display terminal. The major difference is that the exclamation point is not indicated on 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 for specific types of devices being tested. Unexpected results will occur when certain OLTS programs are executed. This is because invalid CCW command codes were processed by VM/SP. Terminal control devices (2701, 2702, 2703) often appear to 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 OLTS selection (DEV/TEST/OPT) defines the same terminal that is serving as the virtual system console, refer to "Invoking OLTS to Virtual Machine Console Terminals" on page 2-14.*

## Invoking OLTS to Virtual Machine Console Terminals

```
logon erep
ENTER PASSWORD:
password (CE enters password, which remains unseen)
LOGMSG - hh:mm:ss[1]  mm/dd/yy[1]
LOGON AT hh:mm:ss zzz day-of-week mm/dd/yy[1]
msg operator attach oltsep tape on 382 as 382[2]
TAPE 382 ATTACHED
msg operator attach dasd 333 as 333[2]
DASD 333 ATTACHED
CP
i 382[3] [4]
04 SEP188D ENTER DATE IN FOLLOWING FORMAT 'MM/DD/YY'
r 04,'mm/dd/yy'[5]
04 SEP330D ENTER TIME IN THE FORMAT 'HH.MM.SS'
r 04,'hh.mm.ss'[6]
SEP392I OLT LOAD ADDRESS IS 020000 HEX.
```

[1] The current time and date is displayed or printed. The zzz represents the time zone in which you are located.

[2] 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.

[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/SP 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
```

[5] Enter today's date.

[6] Enter the current time; use periods between the hours, minutes, and seconds.

**Figure 2-4 (Part 1 of 2). A Typical CE Terminal Session Using OLTSEP-OLTS**

```
SEP102I OLTS RUNNING
SEP107I OPTIONS ARE NTL, NEL, NPP, FE, NMI, EP, CP, PR,
 SI, NRE
01 SEP105D ENTER DEV/TEST/OPT/[7]
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
SEP158I  S T3830B   UNIT 0333
press the PA1 key[8]
CP
log
LOGOFF AT hh:mm:ss[9] ON mm/dd/yy[9]
```

[7] Description of OLTSEP test options are disclosed in the CE document *IBM Maintenance Program: OLTSEP Operator's Guide*, D99SEPD.

[8] Observe that in this example, a long string of OLTS were requested to run on unit 0333. Pressing the PA1 key 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 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.

[9] The current time and date is displayed or printed.

**Figure 2-4 (Part 2 of 2). A Typical CE Terminal Session Using OLTSEP-OLTS**

Situations can occur where the CE may wish to initiate OLTSEP and run OLTS on the same device. In such cases, spurious results are likely to be generated. This happens because the data and control path to the device are being used by two independent programs. 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. For instance, assume a CE wishes to run diagnostic tests on his virtual console, a 3277. The CE logs onto the VM/SP system, loads OLTSEP, and directs the OLTS to be run on the same terminal. OLTS expects a non-formatted screen. The display screen has previously been formatted by VM/SP to be compatible for its own use. Thus, displayed results are dissimilar to expected OLTS 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/SP service the device in a similar manner. The terminal serving as the virtual console and as the test device falls into this category. Use Figure 2-4 on page 2-14 and substitute values.

## OLTS-FRIEND

A sample of an OLTS-FRIEND operation invoked from a virtual machine environment is shown in Figure 2-5. 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 from a System/370 environment.

```
logon erep
ENTER PASSWORD:
password (CE enters password, which remains unseen)
LOGMSG - hh:mm:dd[1] mm/dd/yy[1]
LOGON AT hh:mm:ss[1] zzz day-of-week mm/dd/yy

TAPE 381 ATTACHED
DASD 131 ATTACHED
CP
ipl 381
DISABLED WAIT STATE. CP ENTERED; REQUEST, PLEASE[2]
CP
query lines
```

[1] The current time and date is displayed or printed. The zzz is the time zone in which you are located.

[2] OLTSEP expects a console address of 01F to 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 OLTSEP attempted to send a message to the console address specified by storage location 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, he placed the address in the proper OLTSEP program location. Resumption of OLTSEP operation is invoked by using the CP EXTERNAL command (the virtual machine's external interrupt).

**Figure 2-5 (Part 1 of 2). A CE Terminal Session Invoking OLTS-FRIEND**

```
CONS 009 ON DEV 04B[3]
st b48 00000009
STORE COMPLETE
ext
04 SEP188D ENTER DATE IN FOLLOWING FORMAT 'MM/DD/YY'
r 04[4],'mm/dd/yy'[4]
04 SEP330D ENTER TIME IN THE FORMAT 'HH.MM.SS'
r 04,'hh.mm.ss'[5]
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/
press the PA1 key[6]
CP
logoff
LOGOFF AT hh:mm:ss[7] ON mm/dd/yy[7]
```

[3] 009 indicates the virtual console address and 04B represents the true line address to which the terminal is connected.

[4] Enter today's date.

[5] Enter the current time; use periods between the hours, minutes, and seconds.

[6] Pressing the PA1 key allows the CE to enter the CP environment to perform some virtual machine function; and, at the same time, the previously engaged operation is temporarily suspended. In this instance, the CE chose to log off the system.

[7] The current time and date is displayed or printed.

**Figure 2-5 (Part 2 of 2). A CE Terminal Session Invoking OLTS-FRIEND**

# OLTSEP-RETAIN/370

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

1. Establish line communication to RETAIN/370 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    <----(Initial RETAIN/370 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/370 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/370 site personnel or the on-site CE after the initial test on the specified device was initiated by the on-site CE. Note that "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/370 operation utilizing the OLTS tests from a virtual environment is subject to the same restrictions as are other programs run in other VM/SP virtual machines. See the appendix about "VM/SP Restrictions" in VM/SP Planning Guide and Reference.*

A result of an OLTSEP-RETAIN/370 operation is shown in Figure 2-6 on page 2-19.

```
logon erep
ENTER PASSWORD:
service (CE supplies; is not visible)
LOGON AT hh:mm:ss[1] zzz day-of-week mm/dd/yy
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
DASD 137 ATTACHED
LINE 080 ATTACHED

i 137
04 SEP188D ENTER DATE (AND TIME)-'MM/DD/YY,HH/MM/SS'
r 04'mm/dd/yy,hh/mm/ss'[2]
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'[3]
SEP163I * RETAIN/370 READY
01 SEP105D ENTER DEV/TEST/OPT/
r 01,'380/2400a/nfe,re/'[4]
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/
r 01,'/2400A-D//'[5]
```

[1] Current time and date are displayed or typed. The zzz represents the time zone in which you are located.

[2] You must enter today's date and the current time; use diagonals as shown in the prompt line.

[3] Initial RETAIN/370 request. Note that the letters "re" mean the RETAIN/370 option.

[4] The CE who is on site initiates this response.

[5] The RETAIN/370 site initiates this response.

**Figure 2-6 (Part 1 of 2). Terminal Session Showing Use of RETAIN/370**

```
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?'[6]
01 SEP105D ENTER DEV/TEST/OPT/
R 03,'YES THANKS TERMINATE OPERATIONS'[7]
01 SEP105D ENTER DEV/TEST/OPTION/
press the PA1 key[8]
CP
log
LOGOFF AT hh:mm:ss[9] ON mm/dd/yy[9]
```

[6] Message sent by CE who is on site.

[7] Reply sent from the RETAIN/370 site.

[8] Pressing the PA1 key allows the CE to enter CP environment. At this time, the CE chose to log off the system although some other virtual machine functions could have been performed.

[9] Current time and date are displayed or typed.

**Figure 2-6 (Part 2 of 2). 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/SP is operational and the related interface to the terminal is enabled. Note that VM/SP does not support 3704/3705/3275 lines in NCP mode nor the MTA feature. 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/SP terminal and line condition are:

- The VM/SP program must be operational

- The teleprocessing line must be enabled or the related 3704/3705/3725 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/SP CP Command Reference for General Users*. Essentially, the user enters 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 the user is sending a message to himself, he should 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/SP Terminal Reference* and that the terminal is placed in COMMUNICATE mode.

***Example:***

```
   □<---------(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/SP normally translates lowercase alphabetic data to uppercase in responding to terminal MESSAGE commands.

## Basic Terminal Check via the ECHO Command

Assume that the CE can successfully logon to his assigned virtual machine. Assume also that his terminal is failing because of a local or 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 to 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 ECO command is exclusive to the privilege class G user.

Information on the format and use of the ECHO command is detailed in the *VM/SP CP Command Reference for General Users.* In summary, to use the ECHO command, you must be logged onto your virtual machine and you must be in the CP environment. With these conditions satisfied, you 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 for each line entered.

Figure 2-7 is a terminal session using the CP ECHO command.

```
logon erep
ENTER PASSWORD:
password (CE supplies; it remains unseen)
LOGON AT hh:mm:ss day-of-week mm/dd/yy[1]

echo 3[2]
ECHO ENTERED ; TO TERMINATE TEST TYPE END
ENTER LINE
NOW is THE time□
NOW is THE time }
NOW is THE time } ECHO command responses
NOW is THE time }
ENTER LINE
end[4]
```

[1] Current time and date are displayed or printed.

[2] CE elects three responses.

[3] CE enters text for test. The square bullet (□) represents pressing the return key.

[4] CE must type the word "end" when the test finishes inasmuch as a subsequent command entry otherwise would be treated as ECHO text.

**Figure 2-7. A CE Terminal Session Invoking ECHO Command**

# Section 3. Error Handling

This section contains information about the following:

- Overview of I/O Error Handling
- Record Modification for VM/SP Error Recording
- I/O Error Recording and Error Recording Area
- Recovery Management Support
- Machine Check Handler
- Channel Check Handler
- Missing Interrupt Handler.

# Overview of I/O Error Handling

In multiprocessor mode, both processors have I/O capability. However, in the event of I/O errors, the I/O recovery methods are essentially the same as for uniprocessor systems.

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

## CP-Related Request Error Handling

I/O error recovery is attempted for CP-initiated I/O operations to CP-supported devices.

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. The cycle continues until either the error is corrected or it is classified as unrecoverable. The I/O error recovery routine, which is the controlling factor, may indicate that:

- The activity is to be retried.
- The error has been corrected.
- The error is unrecoverable.

## Virtual-User-Related Request Error Handling

I/O error recovery is attempted by CP for virtual user I/O operations through VM/SP's Diagnose Interface, which mainly is an interface for CMS.

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 procedure lets the errors appear as they would have 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, service level(s), and other data, none of which are part of CP.

The I/O error recovery routine, which is the controlling factor, may indicate that:

- The activity is to be retried
- The error has been corrected
- The error is unrecoverable.

### Error Handling for CP

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 unrecoverable when the maximum allowable number has been reached. When an I/O error routine indicates that an error condition is unrecoverable, 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/SP system residence pack.

- VM/SP spooling errors

- VM/SP 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/SP 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 is 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/SP does not record the error on its error recording area; the SVC 76 interruption is reflected 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/SP (ECPS:VM/370) on systems supported by VM/SP.

## Error Recording -- VM/SP 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-handling and address-handling schemes used by that operating system can be used to create a factual error record.

With VM/SP, these operating systems operate under the control of the Control Program (CP) component of VM/SP. 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/SP's control, the following are virtual, not real:

- The device
- The data address
- 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.

Other devices, whether or not supported by VM/SP, can be dedicated to an operating system, in which case VM/SP does not translate data addresses or device types. Device address mapping, however, may still be done.

In a 3850 Mass Storage Systems (MSS) application, the 3330V (3330 virtual volume) associated with a given CPUID is specified as input to the OS/VS Mass Storage Control Table Create Program. This program is described in *OS/VS Mass Storage (MSS) Installation Planning and Table Create*, GC35-0028. 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/SP 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/SP'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/SP 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 the CE 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/SP takes care of this problem.

## SVC 76

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/SP 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/SP (running in a virtual machine environment)
OS/360 (Release 21)
VS1
VS2 Release 1 (with single address space)
VS2 Release 2 (with multiple address spaces)
DOS/VSE.

***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/SP traps it. VM/SP 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/SP will:

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

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

3. Inform the VM/SP 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-operation instruction insofar 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/SP gathers the data for the error recording record.

If any of the above-mentioned operating systems are running standalone and 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 error-recording data set.

In either case, as far as job processing is concerned, SVC 76 and I/O error recording are 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/SP error recording area immediately and informs the VM/SP 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/SP examines the contents of general registers 0 and 1 to determine if valid conditions exist for handling the error recording data. Figure 3-1 shows the result of the comparison.

| System | General Register | Operation and/or Meaning |
|---|---|---|
| OS<br>(Release 21 or above )<br>VS1<br>VS2<br>VM/370 BSEPP<br>VM/370 SEPP<br>VM/SP<br>(in a virtual environment) | 0 | Two's complement of the error record length |
| | 1 | Address of the error record |
| DOS/VSE | 0 | Address of the error record minus 8 |
| | 1 | **Bytes** **Content**<br>0 Bit zero must be a 1<br>1-3 CCB address (DOS control block for I/O) |

**Figure 3-1. Contents of General Registers for Various Systems**

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

## Record Modification for VM/SP Error Recording

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

- Type 30, 36, OBR (Outboard Recorder)

  *Common Fields*:

  > *Primary and Alternate CUA* are replaced with the real device address corresponding to the virtual device address.
  >
  > *CPUID* (processor model number) is replaced with the real machine model number.
  >
  > *JOBID* is replaced with the virtual user ID.

  *Device Dependent Fields*:

  For dedicated DASD units no modification is required. For nondedicated DASD units, 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 if the DASD unit is a count-key-data device; or adjusts the physical block number if the device is a fixed block device.

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

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

*2305, 3330, 3340, 3350, 3375, 3380, and FB-512* 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 not 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* (processor 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* (processor model number) is replaced with the real machine model number.

  *JOBID* is replaced with the virtual user's ID.

  *Primary CUA or "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* (processor 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 units, no modification is required. For non-dedicated DASD units, 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* (processor 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, DMKIOF, and DMKIOJ.

***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/SP error recording area. In addition, a message is sent to the VM/SP primary system operator informing him of the real unit address of the device and the userid that is performing the I/O. 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/SP error routines are invoked for recording counter and buffer overflow statistics for various devices, for recording demounts, and for recording general statistical data in the VM/SP error recording area.

## I/O Error Recovery--Detailed Description

I/O error recovery is attempted for CP-initiated I/O operations to 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 of a virtual machine event, the I/O interrupt is reflected to the virtual machine. For 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. In multiprocessor applications, although the I/O operations can occur on both processors, all I/O error recovery procedures are essentially the same as those of uniprocessor methods.

During an I/O operation, the control block linkage shown in Figure 3-2 is in effect.

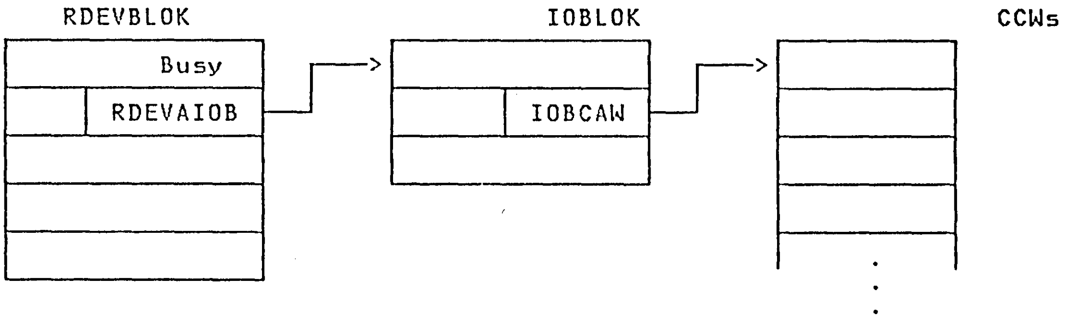


**Figure 3-2. I/O Operation Control Block Linkage**

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 3-3.

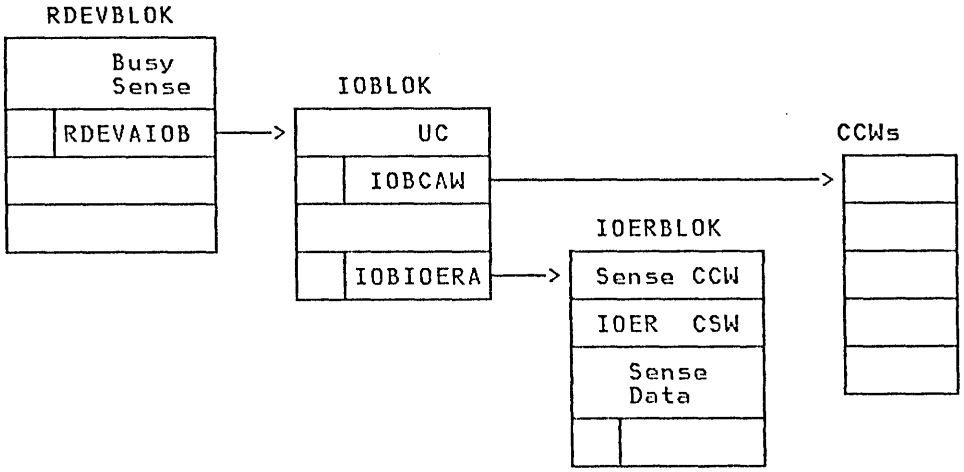


**Figure 3-3. I/O Error Recovery Control Block Structure for Sense—Byte Analysis**

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 whether or not 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 3-4.

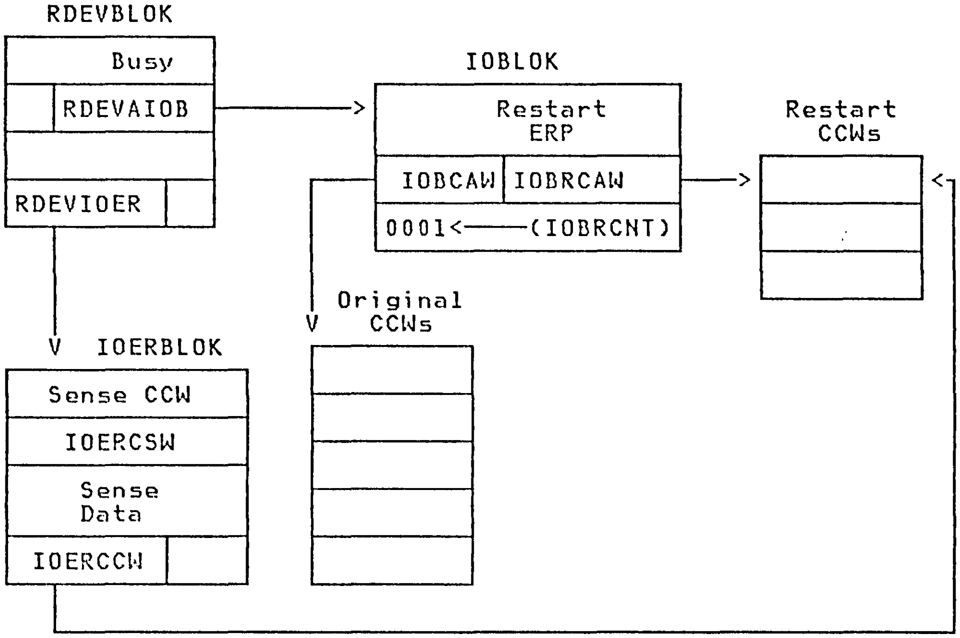


**Figure 3-4. Control Block Linkage for Retry**

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 3-5 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 permanent error flag 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 it does, 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/SP 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.

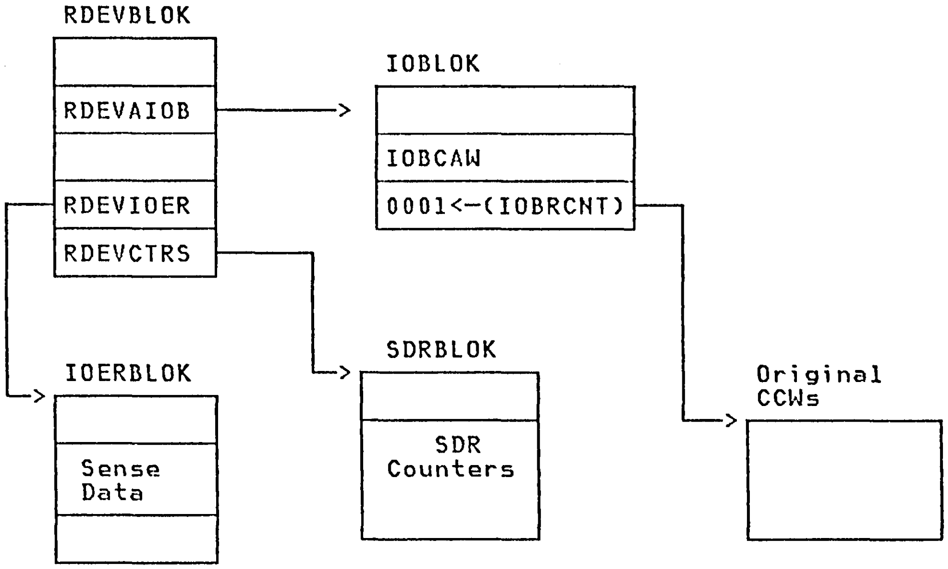


**Figure 3-5. Control Block Relationship for SDR Counter Update**

# I/O Error Recording and Error Recording Area

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

## Error Recording Routines

The error recording routines of VM/SP do not actually perform I/O operations. Instead, the I/O error routines treat the error recording area allocated on the VM/SP system residence pack as a logical extension of VM/SP storage. These extensions of VM/SP storage are in the form of logical pages that can be read and written out of by the paging supervisor of VM/SP. 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.

## Error Recording Areas

On count-key-data devices, 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/SP generation procedures. On FB-512 devices the error recording area is any number of adjacent pages assigned on the system residence volume. The starting page number and the number of pages are specified in the VM/SP generation procedures. In any case, when the error recording area is 90 percent full, and again when 100 percent full, the I/O error routines instruct the VM/SP system operator to invoke the CPEREP command[1] 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.

Because of the support provided for the 303x processors, CPEREP processing is not dependent on the content or engineering change (EC) level of the processor logouts to format machine check and channel check records. Instead, the 7443 Service Record File (SRF) device provides format and content information contained in frames on diskette to format MCH and CCH records. In a 303x attached processor and multiprocessor environment, each processor has its own SRF device. Customer engineering maintains the SRF frames (records containing text and scan buffer codes to format MCH and CCH records) on each SRF device. CPEREP makes use of these frames to interpret and format inboard errors for hardcopy output.

[1] Detailed instructions for using CPEREP are contained in *EREP User's Guide and Reference.*

At initialization, the VM/SP system control program recognizes the presence of multiple SRF devices in certain 303x attached processor and multiprocessor environments. CP accesses the SRF device(s) at initialization, retrieves the frames, and records them at the beginning of the error recording area.[2] When multiple SRF devices exist in a 303x AP or MP environment, the header portion of each SRF frame record written to the error recording area identifies the processor by processor number and model number. The interrupt handler routine identifies which MCH and CCH records pertain to which processor. In this way, CPEREP uses SRF frames to format MCH and CCH records for printed reports by matching the inboard error records to their respective frames.

Each time an engineering change (EC) requiring a new diskette is installed in any 303x processor environment, the privilege class F user must issue the CPEREP CLEARF command. This command clears and reformats the error recording area by accessing the format information in the SRF frames on the newly installed diskette.

In 303x processor environment, system generation procedures provide support for the SRF device(s) so that CPEREP can properly format machine check and channel check records created by each processor. A channel path must also exist between the main processor and the SRF of the attached processor in a 303x attached processor environment. Establishing this channel path allows CP to read frames from each of the SRF devices to the error recording area. For the requirements needed to generate support for the SRF device(s), see *VM/SP Planning Guide and Reference.*

The SRF device is accessed by VM/SP to read frame data (1) during VM/SP system initialization if the error recording cylinders have not been previously formatted; and (2) as a result of running CPEREP with the CLEARF operand. To ensure that the VM/SP control program has access to the SRF device after initialization, 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 (C1) on the service support console.

3. Note that 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 a 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

[2] This sequence occurs if, and only if, at initiation (1.a) the SYSERR cylinders are not formatted for CPEREP, (1.b) there are defined and operational paths to both SRFs, and (2) if -- at CLEARF operation time -- operational paths exist to both SRFs. For additional information, see *VM/SP Planning Guide and Reference.*

a virtual machine when it initializes its error log in order that frame data may be read from the SRF. The VM/SP 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.

In single processor mode, the SRF device of the VM/SP processor must be attached to the MVS V=R virtual machine before MVS runs IFCDIP00 to initialize SYS1.LOGREC.

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

- Outboard Recording
  - Statistical data recording of errors related to VM/SP
  - 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/SP 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, the I/O error recording routine formats a short OBR record for any devices that have SDR counters associated with them. (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/SP-initiated I/O events are recorded by the I/O error recording routines of VM/SP. When a device-dependent error recovery procedure determines that an I/O event cannot be successfully recovered, the permanent error 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 3-6. The I/O error recording routines format the error and record it on the error recording area.

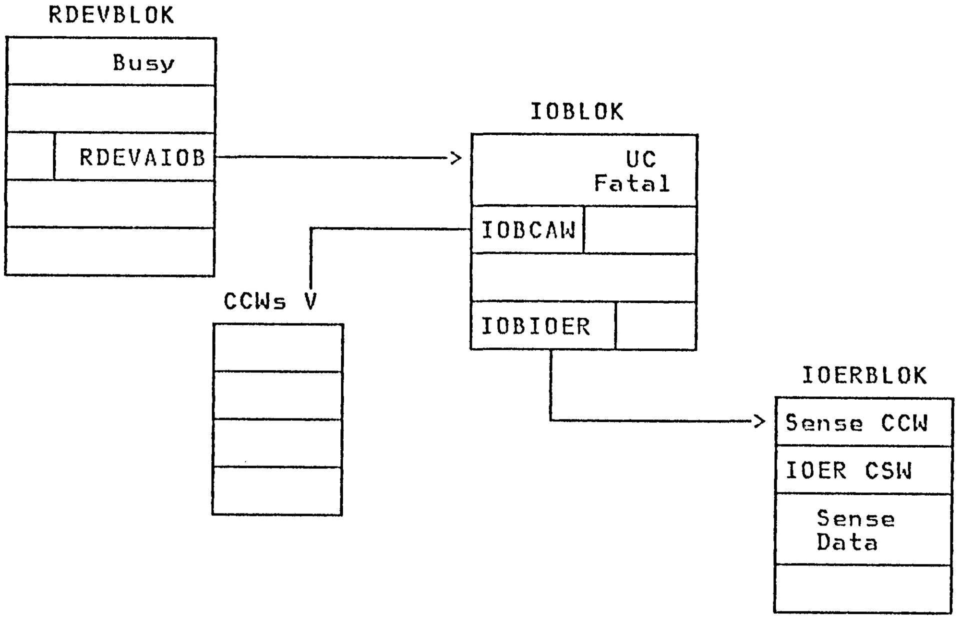


**Figure 3-6. Control Block Linkage--Unrecoverable Error Condition**

## Environmental Data Recording

When the I/O supervisor receives a unit check interruption from a 3330, 3340, 3350, 3375, 3380, or 2305, the count-key-data error recovery procedure is invoked. If the unit check is from an FB-512 device, that error recovery procedure is used. In any case, 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, 3375, 3380, FB-512 | 2 | 3 | Environmental data |

The manner in which the error recovery procedure passes the data to the I/O error recording routine is different for the 2305 than it is for the 3330, 3340, 3350, 3375, 3380, and FB-512, as shown in Figure 3-7 and Figure 3-8 on page 3-18, respectively.

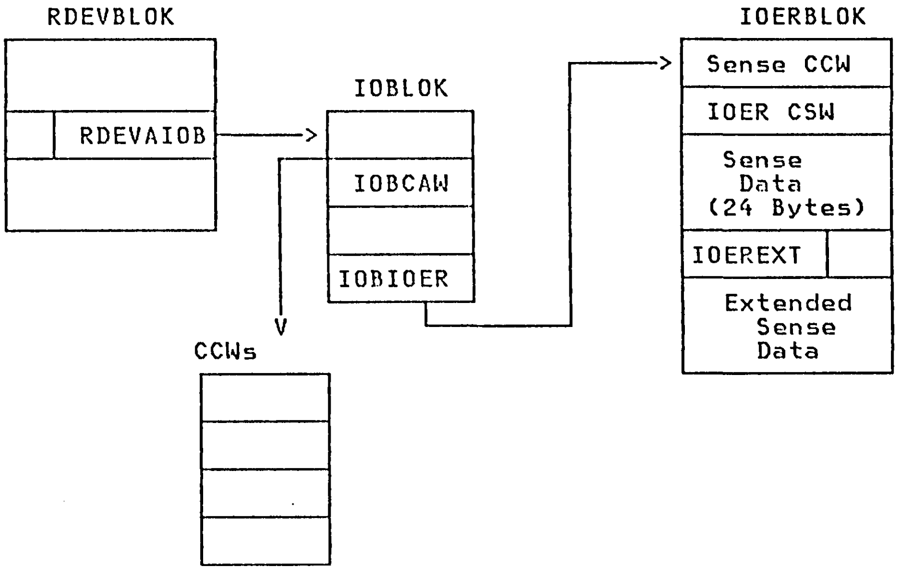


**Figure 3-7. 2305 Control Block Structure--Environmental Data Recording**

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

A different control block linkage exists on the 3330, 3340, 3350, 3375, 3380, and FB-512 environmental data recordings because of the amount of data. The DASD error recovery procedure builds multiple IOERBLOKs and chains them together to pass the data to the I/O error recording routines.

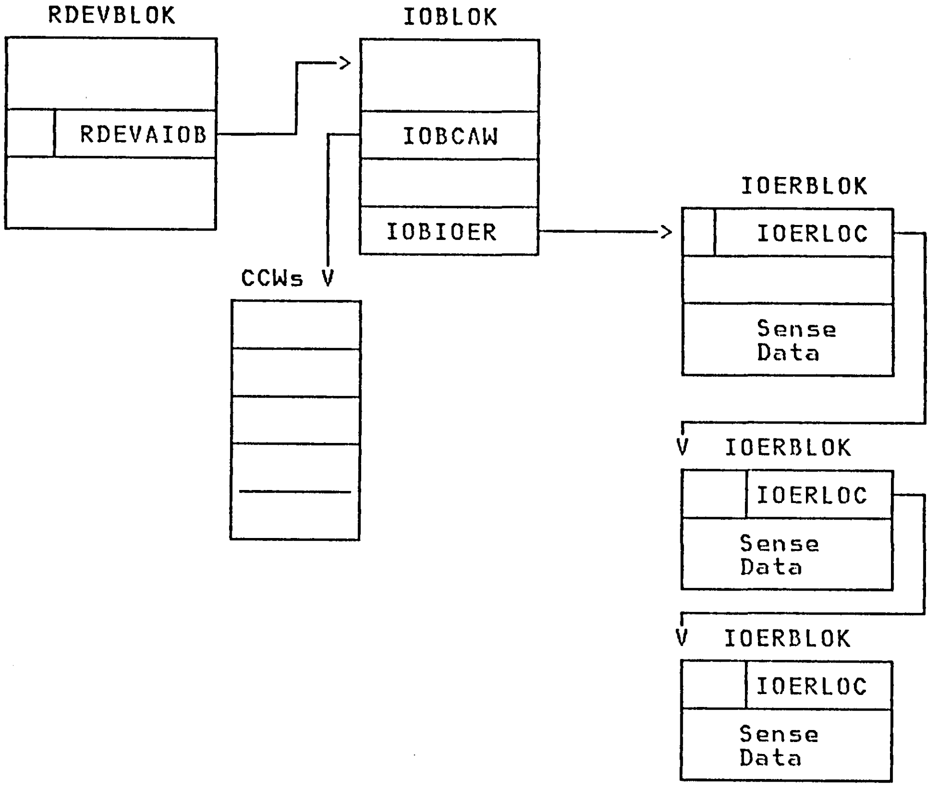


**Figure 3-8. 3330, 3340, 3350, 3375, 3380, and FB-512 Control Block Structure--Environmental Data Recording**

The IOERLOC pointer in the IOERBLOK points to the next IOERBLOK on 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 proper 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 match previously defined sense data values. The SET RECORD command starts intensive mode. The specified device must be in use by the virtual machine issuing the SET RECORD command.

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/SP 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[3] to retrieve the records created while this option was active.

No messages appear to inform either the VM/SP 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 one 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 class F user invokes SVC 76 to pass a record to CP to record, CP will honor the request.*

## VM/SP I/O Error Recordings

Unit check error (outboard) records (OBR) are written for all users, except those with privilege class F, when any of the following conditions exist. However, records will be kept if the privilege class F user specifies intensive care for a particular device. Figure 3-9 lists the device and reason for the OBR being written.

| Device(s) | Reason OBR Was Written |
|---|---|
| All VM/SP supported units | An unrecoverable (permanent) I/O error has occurred. It was initiated as a VM/SP I/O task -- a CP request. |
| All units with SDR counters | Counter overflow -- setting was exceeded. |
| All units with SDR counters | The SHUTDOWN command and/or the NETWORK SHUTDOWN command were issued. |
| All units with SDR counters | The VARY OFFLINE command was issued. |
| 2305, 3330, 3340, 3350, 3375, 3380, FB-512 | Equipment check. |
| 2305, 3330, 3340, 3350, 3375, 3380, FB-512 | Busout check. |
| 2305, 3330, 3340, 3350, 3375, 3380, FB-512 | The MDR record on the BUFFER UNLOAD command (X'A4' or X'24') is being directed to a nondedicated DASD by a virtual machine. |
| 3340 | Seek check. |

**Figure 3-9 (Part 1 of 2). Devices for which OBRs are written**

[3] Detailed instructions for using CPEREP are contained in *EREP User's Guide and Reference.*

| Device(s) | Reason OBR Was Written |
|---|---|
| 2305, 3340, 3350, 3375, 3380, FB-512 | Data check. |
| 2305, 3340, 3350, 3375, 3380, FB-512 | Overrun. |

**Figure 3-9 (Part 2 of 2). Devices for which OBRs are written**

## Error Recording Record Layout

Error recordings vary in length and format depending on the malfunction or the device encountered. Data that relates to machine check, unit check (OBR), channel check, missing interrupt, or nonstandard (MDR) conditions is contained in the appropriate error record. The error records are described and mapped in *EREP User's Guide and Reference.*

Figure 3-10 on page 3-21 identifies the source for the data that is contained in the header record. Figure 3-11 on page 3-23 identifies the source for the data that is contained in each error record.

For additional information on error record layout as used by the CP component of VM/SP, refer to *VM/SP Data Areas and Control Block Logic Volume 1 (CP).* For information on the printout format of supported error record types, refer to the *EREP User's Guide and Reference.*

| Record Header Field | Source of Data |
|---|---|
| Class/Source (1 byte) | From calling routines or type of entry |
| System Release Level (1 byte) | System description module |
| Switches (Dependent/Independent -- 4 bytes) | |
| Byte 0 | |
| Bit 0 Multiple Record Recording | NA |
| 1 NS Machine | Always 0 (using NS Clock Binary) |
| 2 Record Truncated | PSW |
| 3 Reserved for IBM Use | - |
| 4 Time Macro Used (HHMMSS) | Always 1 |
| 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 (OBR) | |
| Bit 0 SDR dump (EOD) | RECORDER |
| 1 Temporary error | IOBLOK |
| 2 Short record | RECORDER |
| 3 MP system | NA |
| 4 Processor B | NA |
| 5 Volume dismount | NA |
| 6 SVC requested | NA |
| Byte 2 MISSING INTERRUPT | |
| Bit 0 Channel end interrupt pending | MIH |
| 1 Device end interrupt pending | MIH |
| NA = Not Applicable | |

**Figure 3-10 (Part 1 of 2). Header Record Table**

| Record Header Field | Source of Data |
|---|---|
| Byte 2 MISCELLANEOUS DATA RECORDER<br>(Nonstandard)<br>Record ID Code (in hexadecimal)<br>01 = 3330<br>02 = 2305 Model II<br>03 = 3270<br>04 = 3211<br>05 = 3705<br>08 = 2715<br>09 = 3340<br>0A = 3330 Model II<br>10 = 3211-type printers (except 3211 itself)<br>11 = 3350<br>12 = 2305 Model I<br>14 = 3380<br>16 = 3310<br>17 = 3370 Model A1 or B1<br>18 = 3375<br>1A = 3370 Model A2 or B2<br>40 = 8809<br>41 = 3480<br>FF = Reserved for IBM use | |
| Record Count (1 byte) | |
| Reserved for IBM use (1 byte) | |
| System Date and Time (8 bytes) | Recorder |
| CPU Identification (8 byte) | Store Processor ID |
| NA = Not Applicable | |

**Figure 3-10 (Part 2 of 2). Header Record Table**

| MC | UC | UC Short | CC | Non Std | MI | Data Recorded | Source |
|---|---|---|---|---|---|---|---|
| x | x |  | x |  | x | Job ID (USERID) | VMBLOK |
|  |  | x | x |  | x | Channel & Unit | RDEVBLOK |
| x | x |  |  |  |  | Mach Ck Old PSW | MCH Buffer |
| x | x |  |  |  |  | Mach Ck Independent Logout | MCH Buffer |
| x | x |  |  |  |  | Processor Hardware Logout | MCH Buffer |
| x | x |  |  |  |  | Damage Assessment | NA |
|  |  |  | x |  |  | Active I/O Units on Channel | CCH |
|  | x |  | x |  |  | Failing CCW | IOBLOK |
|  | x |  | x |  |  | CSW | IOERBLOK |
|  |  |  | x |  |  | Extended CSW | CCH |
|  | x | x | x |  |  | Physical Spindle or Channel & Unit | IOERBLOK<br>IOBLOK |
|  | x | x | x |  | x | Device Type | RDEVBLOK |
|  |  |  | x |  |  | Channel ID | RCHBLOK |
|  |  |  | x |  |  | Channel Logout | CCH |
|  | x |  |  |  | x | I/O Retries | IOBLOK |
|  | x |  |  |  | x | Volume ID | RDEVBLOK |
|  | x |  |  |  |  | Last Seek Address | IOBLOK<br>IOERBLOK |
|  | x |  |  |  |  | Actual Home Address | IOERBLOK |
|  | x |  |  |  |  | Sense Data | IOERBLOK |
|  |  |  |  |  | x | Time Interval | MIH |
|  |  |  | x |  |  | Multiprocessing | NA |
|  | x |  |  |  |  | Device-Dependent Data Count | IOERBLOK |
|  | x | x |  |  |  | Statistical Data Work Count | Recorder |
|  | x |  |  |  |  | Sense Byte Count | IOERBLOK |
|  |  |  |  | x | x | Device-Dependent |  |
|  | x | x |  |  |  | Statistical Data Record (SDR) Counters | SDRBLOK |

Legend:

CC = Channel Check
CCH = Channel Check Handler
MC = Machine Check
MCH = Machine Check Handler
MI = Missing Interrupt
MIH = Missing Interrupt Handler
NA = Not Applicable
Non Std =Nonstandard (MDR)
UC = Unit Check (OBR)

**Figure 3-11. Record Breakdown Table (Except Header)**

## VM/SP Recovery Features--Introduction

The primary objectives of VM/SP'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/SP.

## Machine Check Handler (MCH)

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. Concurrent 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.

- Isolate the failure to one or more channels and attempt to recover the failing channels by issuing CLRCH to the channels. If the channel(s) cannot be recovered, processing may continue with all paths through the failing channels marked offline.

- Place the system in a disabled wait state.

- If, while operating in AP mode, an unrecoverable malfunction occurred on the attached processor in problem program state, resume operations in uniprocessor mode.

- In VM/SP multiprocessor environments, processing may continue in uniprocessor mode if either processor malfunctions while in problem program state and recovery is not possible.

Virtual machines that have VMSAVE (Directory option or SET command operand) enabled normally save their register and storage contents in the event of certain abend situations. However, the following machine errors cause a disabled wait PSW to be loaded and may prevent saving the contents of a virtual machine.

- MCIC invalid
- PSW masks, key, program mask, or CC invalid
- Floating-point, control, or general registers invalid when CP was in control

- System damage
- Timer damage
- Processor clock damage
- Instruction processing damage when CP was in control
- Machine check recursion.

## 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/SP primary system operator. For the 4300 series processors, limited channel logout is still available, but no fixed or I/O extended logout area exists.

Virtual machines for which VMSAVE (Directory option or SET command operand) is enabled normally have their register and storage contents saved in the event of certain abend situations. However, catastrophic channel errors cause a disabled wait PSW to be loaded and may prevent saving the contents of a virtual machine.

## Missing Interrupt Handler (MIH)

The missing interrupt handler provides CP with an automatic means of monitoring system I/O activity for missing interrupt conditions. In order to minimize operator or system programmer intervention, the Control Program (CP) attempts to correct missing interrupt conditions. For every missing interrupt condition detected, a record is written to the system error recording area (LOGREC) in order to provide the operator or the system programmer with information to:

- take corrective action
- schedule maintenance for the device.

## Handling of Hard Machine Checks

If a permanent error (hard machine check) occurs on a 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/SP 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) the virtual machine.

If VM/SP 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 may continue in uniprocessor mode on the main processor.

In certain 303x attached processor environments, a Channel-set Switching facility may exist. This facility allows processing to continue on the attached processor in uniprocessor mode after the main processor enters a disabled wait state following a hard machine check or channel check that results in an uncorrectable error. Automatic processor recovery routines test for the Channel-set Switching facility. If the facility is present, CP switches all active channels on the main processor to the attached processor, and the processing continues on the attached processor in uniprocessor mode. The specific 303x attached processors that support Channel-set Switching are listed in the *VM/SP Planning Guide and Reference.*

Note: *The Channel-Set Switching facility is supported on the IBM 308X processor only when the system is generated for attached processor (AP) mode.*

If VM/SP is being run in multiprocessor mode and an unrecoverable error occurs on a processor that is executing in problem program state, system operation may continue in uniprocessor mode with the failing processor and its channels marked offline.

***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 corresponding page table entry is marked invalid if the error is intermittent or the page frame is marked 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/SP resets the virtual machine, clears its virtual storage by setting it 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/SP 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/SP or a virtual machine, the key is corrected and operation continues.

If the storage protect key error is uncorrectable 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/SP cause the VM/SP system to be terminated. An automatic restart reinitializes VM/SP.

***Extended Storage Key Protection:*** On a 308X processor complex or a 3033 processor equipped with the 3033 Extensions Feature (#6850), the control program (CP) can initialize storage protected by 4K keys rather than 2K keys. Because VM/SP now supports certain hardware instructions, the 4K storage keys and their associated frames can be set to zero at system initialization time.

CP also simulates the hardware instructions for virtual machine operating systems executing in extended control mode on either the 308X or the 3033 with the 3033 Extensions Feature.

For additional information about extended storage protection, see *VM/SP Planning Guide and Reference*, *VM/SP System Programmer's Guide*, and *VM/SP System Logic and Problem Determination Guide Volume 1 (CP).*

## 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 -- record mode or quiet mode.

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

A change from record 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 record mode, a machine check record containing information about the error is written in the error recording area. This record includes the data in the fixed logout area, the date, the time of day, and other pertinent information. In most cases, 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/SP or the problem program, depending upon which had control at the time the machine check occurred.

Multiple-bit ECC storage errors that occur on a 3031, 3032, 3033, or 308X processor are not recorded as soft errors, but rather as unrecoverable 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 or not the error is intermittent. The SET MODE MAIN command is treated as invalid on these processors.

## Error Recovery Procedures

VM/SP includes device-dependent error recovery procedures for all devices supported by VM/SP. Functionally, these procedures perform as their counterparts do in an OS or DOS system. VM/SP uses the standards used by OS or 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 error occurs that is related to a CP input/output operation, such as paging or spooling. If VM/SP cannot correct the error, VM/SP records the error and notifies the system resource operator of the error.

***Handling of Virtual Machine Input/Output Errors:*** VM/SP 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/SP to perform input/output operations, and VM/SP 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/SP provides uniform recording of errors encountered by operating systems running in virtual machines. VM/SP 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/SP (running in a virtual machine)
OS/VS1
OS/VS2 (including MVS/XA)
VSE.

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 from virtual to real device addresses and then records the error information in the VM/SP error recording area. 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/SP sends a message to the primary system operator.

If a virtual machine is using one of the above-cited 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 Record Editing and Printing (EREP) program is executed when the CMS command CPEREP is invoked.[4] The output produced by the command is determined by information contained in the VM/SP error recording area and/or SYS1.LOGREC data on tape and by the supplied CPEREP operands. The printed output from CPEREP under VM/SP has the same format as that generated by EREP running in an OS/VS machine.

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 from a 3031, 3032, or 3033 processor.

For additional information about CPEREP error record retrieval see "CPEREP Error Record Retrieval" on page 4-2.

## VM/SP 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.

[4] Detailed instructions for using CPEREP are contained in *EREP User's Guide and Reference.*

The service representative can execute online tests from a terminal as a user of the system; VM/SP 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/SP restrictions may not run correctly in a virtual machine environment.

## VM/SP Restart Facilities

When either MCH or CCH determines that an error has damaged the integrity of VM/SP, the system is placed in a disabled wait state. On a subsequent reloading of VM/SP, the system operator can elect to execute a warm start to allow completed spool files to be maintained. The system operator must then issue the CP ENABLE command to re-enable the terminal lines. 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/SP system failure, each user must reinitialize his virtual machine.

### Malfunction Handling

The same philosophy of malfunction handling is evident in attached processor and multiprocessor mode operations.

***Attached Processor Mode Malfunction:*** In attached processor mode, when error analysis determines that a nonrecoverable fault is associated with the attached processor while it was running in problem program state, the system continues to operate but in uniprocessor mode on the main processor.

Should the error occur on the main processor when the Channel Set Switching facility is present, then CP switches all active channels from the main processor and processing continues in uniprocessor mode.

***Multiprocessor Mode Malfunction:*** In multiprocessor mode, when an unrecoverable error occurs on either processor while that processor is running in problem program state, the system may be able to continue operating but in uniprocessor mode on the remaining processor. In addition, virtual machines associated with the failing processor (AFFINITY option set to the failing processor) are set for execution on the remaining processor. Those virtual machines are notified of the system action and their virtual machine consoles are placed in console function mode.

Note: *If the IBM 308X processor is initialized as a multiprocessor system and one of the processors fails, the failing processor and its channels are taken offline and CP does not perform channel set switching.*

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/SP 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 CP. 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
- interface control checks for user SIO-initiated channel programs
- 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 CP. RMS is part of the Control Program, and is provided on all processors supported by VM/SP and on their supported channels.

The primary objectives of RMS are:

- to reduce the number of system terminations that result from machine malfunctions
- to minimize the impact of such incidents when they occur (see Figure 3-12).

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 | One of the following:<br>• To record all machine checks and recover from hard machine checks<br>• To reset or terminate virtual machines<br>• To terminate System/370 operations<br>• If attached processor or multiprocessor mode, change to uniprocessor operations as needed. | DMKMCH[1]<br>DMKMCT[2]<br>DMKACR[2] |
| Channel Check Handler | One of the following:<br>• To record channel checks and effect proper recovery<br>• To terminate System/370 operations when necessary. | DMKCCH<br>DMKACR[2] |
| Missing Interrupt Handler | To record all missing interrupt conditions automatically so that CP, rather than the operator or system programmer, attempts to correct them. | DMKCFP<br>DMKCFQ<br>DMKCPI<br>DMKDID<br>DMKIOE<br>DMKIOS<br>DMKIOT |

**Figure 3-12. Summary of Recovery Management Support Functions**

[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 and multiprocessor mode termination situations, malfunction alerts, and automatic processor recovery are contained in the module DMKMCT.

## 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. Concurrent 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.

*Note: If you are using a 308X processor, remember that the 308X does not store:*

- *Machine-check fixed logout*
- *Machine-check extended logout*
- *Region code.*

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 whether MCH action should:

- 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.

- Isolate the failure to one or more channels and attempt to recover the failing channels by issuing CLRCH to the channels. If the channel(s) cannot be recovered, processing may continue with all paths through the failing channels marked offline.

- Place the system in a disabled wait state.

- Enter uniprocessor mode if the attached processor malfunctions while in problem program state and recovery is impossible. VM/SP attached processor operations enable such alternate processing.

- Enter uniprocessor mode if the main processor malfunctions while in problem program state and recovery is impossible. In certain 303x attached processor environments, the CP component of VM/SP -- when Channel-Set Switching facility is installed -- switches all active channels on the main processor to the attached processor.

- Enter uniprocessor mode if either processor of a multiprocessor environment 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
- system repair.

These levels of error recovery are discussed 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. If, however, (1) the retry or storage correction is unsuccessful, (2) it is impossible to retry the interrupted instruction, or (3) the storage failure cannot be repaired, RMS assesses the damage and does the following:

- If the fault is an SPF key failure, refresh the key where 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.

- If the malfunction cannot be repaired but is traceable to a particular virtual machine, terminate or reset the virtual machine.

- If system integrity is lost and nonrecoverable, terminate all SCP operations and post a wait state code.

- If the malfunction, in attached processor applications, 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.

- In multiprocessor application, if the processor malfunction occurs while the processor is running in problem program state and if recovery of the processor is not possible, cease operations on the failing processor and allow the system to continue operation in uniprocessor mode.

- If the error is a channel inoperative or I/O instruction/interruption timeout on a 3031, 3032, or 3033 processor, attempt recovery of the failing channel(s) by issuing CLRCH to each affected channel. If CLRCH does not restore a failing channel to an operational state and if the system can continue operation without that failing channel, mark all paths through the channel as being offline and continue system operation in the same mode as was in effect at the time the error occurred.

Any of the above conditions can produce one or both 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; these errors may or may not be recorded. See "Recovery Modes" on page 3-36 for a discussion of this.

The messages produced by the machine check handler (MCH) on supported VM/SP systems are described in *VM/SP System Messages and Codes.* Wait state codes 001 and 013, produced by the machine check handler routines, are also described in *VM/SP System Messages and Codes.*

The action that the MCH takes for a given situation is determined by:

- The error itself
- The operating environment of VM/SP
- Whether the system was performing
  - A CP function
  - A virtual machine function
  - No function at all (a loaded wait state condition when the error occurred).

Figure 3-13 on page 3-37 clarifies the action the system takes for the given situation.

## Recovery Modes

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/SP 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/SP supported processors, with the exception of the 155 II and 165 II, run disabled for ECC (error checking and correction) at system initialization. All processors, including the 4331, 4341, 3031 AP, and 3033 AP, run 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. In multiprocessor applications, SET MODE values can be set for either processor if desired. However, note that the SET MODE command can only be used by privilege class F users.

*Note: The SET MODE MAIN command is treated as invalid on the 3031, 3032, or 3033 processor (UP, AP, MP) as well as on the 3031 AP and on the 308X processor.*

| Error Condition | VM/SP Processing (CP) | | | |
|---|---|---|---|---|
| | Uniprocessor Operation | Attached Processor Operation | | Multi-processor Operation |
| | | Main | Attached | |
| Invalid machine check interrupt code | 1 | 1 | 1 | 1 |
| Invalid PSW data | 1 | 1 | 1 | 1 |
| Invalid program mask instruction address | 1 | 1 | 1 | 1 |
| Invalid logical storage registers | 1 | 1 | 1 | 1 |
| System damages | 1 | 1 | 1 | 1 |
| Time-of-day or processor clock errors | 1 | 1 | 1 | 1 |
| Processor clock errors | 1 | 1 | 1 | 1 |
| Channel check stop, channel | 1 | 1 | 1 | 1 |
| Multibit (unrecoverable) storage error | 1 | 1 | 1 | 1 |
| Multibit (intermittent) storage error | 1 | 1 | 1 | 1 |
| Storage Protect Key (unrecoverable) failure | 1 | 1 | 1 | 1 |
| Storage Protect Key (intermittent) failure | 2 | 2 | 2 | 2 |
| Malfunction alert | 5 | 1 | 1 | 1 |
| Channel group inoperative | 6 | 6 | 5 | 6 |

Legend:

1 = Load wait state PSW
2 = Refresh for retry operation
3 = Terminate the virtual machine
4 = Automatic processor recovery
5 = Not applicable
6 = Channel recovery

**Figure 3-13 (Part 1 of 2). Condition/Action Table for Uncorrectable Errors**

| Error Condition | Virtual Machine Processing | | | |
|---|---|---|---|---|
| | Uniprocessor Operation | Attached Processor Operation | | Multiprocessor Operation |
| | | Main | Attached | |
| Invalid machine check interrupt code | 1 | 1 | 1 | 1 |
| Invalid PSW data | 1 | 1 | 1 | 1 |
| Invalid program mask instruction address | 3 | 3 | 3 | 3 |
| Invalid logical storage registers | 3 | 3 | 3 | 3 |
| System damages | 3 | 3 | 3 | 3 |
| Time-of-day or processor clock errors | 1 | 1 | 3,4 | 3,4 |
| Processor clock errors | 3 | 3 | 3 | 3 |
| Channel check stop, channel | 1 | 1 | 1 | 1 |
| Multibit (unrecoverable) storage error | 3,2 | 3,2 | 3,2 | 3,2 |
| Multibit (intermittent) storage error | 3,2 | 3,2 | 3,2 | 3,2 |
| Storage Protect Key (unrecoverable) failure | 3 | 3 | 3 | 3 |
| Storage Protect Key (intermittent) failure | 2 | 2 | 2 | 2 |
| Malfunction alert | 5 | 1 | 3,4 | 3,4 |
| Channel group inoperative | 6 | 6 | 5 | 6 |

<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
6 = Channel recovery

**Figure 3-13 (Part 2 of 2). Condition/Action Table for Uncorrectable Errors**

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/SP 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/SP 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/SP 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 -- An Overview

There are several 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 recognized only on the 3031, 3032, and 3033 processors.

The channel check handler receives control from the I/O supervisor when any of the above-listed channel checks 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 to all ones both the ECSW and the logout areas.

- 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 upon 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. Figure 3-14 and Figure 3-15 show, in greater detail, under what circumstances the various channel checks are reflected to the virtual machine.

| Types | Nondedicated 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 3-14. Handling of Channel Check, Channel Control Check, and Interface Control Check**

| Types | Nondedicated Channel | Dedicated Channel |
|---|---|---|
| CP I/O | CP attempts recovery | CP attempts recovery |
| Virtual Machine SIO I/O | CP attempts recovery | 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 3-15. 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/SP-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.

Note: *When using the STIDP instruction, be aware that you will have a machine-type number stored for 308X instead of a model number. To get the model number, issue the STAP instruction. For more information on this difference, see IBM System/370 Principles of Operation.*

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

- Obtains storage for maximum I/O extended logout area for the VM/SP-supported system.

- Initializes logout and ECSW to all ones.

- Sets up the I/O extended logout 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/SP system's operator or customer engineer may obtain a copy of the record by using the CMS command CPEREP.[5] 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. If CCH determines that the error is an I/O interface inoperative error, CCH will call DMKACR to attempt to recover the failing channel. If the channel is successfully recovered or if system operation can continue with the channel being marked offline, CCH returns control to the I/O supervisor.

[5] Detailed instructions for using CPEREP are contained in *EREP User's Guide and Reference.*

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/SP Channel Check Handler action is summarized in Figure 3-16. Possible channel check action codes and their meanings are as follows:

| **Code** | **Meaning** |
|---|---|
| 1 | Schedule recording. |
| 2 | Schedule system termination with proper message (error data can be retrieved if SEREP is invoked). Note that when using a 308X or a 4300 processor, invoking SEREP will give you invalid results. |
| 3 | Error can be isolated to a device for retry. |
| 4 | Error can be isolated to a channel for retry. |

| Channel Address Valid | Retry Codes Valid | Channel Has Been Reset | Start I/O Time | Unit Address Valid | Action Code |
|---|---|---|---|---|---|
| No | | | | | 2 |
| Yes | No | | | | 2 |
| Yes | Yes | Yes | | | 1, 4 |
| Yes | Yes | No | Yes | | 1, 3 |
| Yes | Yes | No | No | No | 1 |
| Yes | Yes | No | No | Yes | 1, 3 |

**Figure 3-16. 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/SP System Messages and Codes.* 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/SP System Messages and Codes.*

## Missing Interrupt Handler -- An Overview

Virtual machine users can be locked out of the system or have system performance adversely reduced because of:

- Incomplete minidisk I/O
- Incomplete paging I/O.

When either of these conditions prevails, the missing interrupt handler (MIH) detects the particular condition and attempts corrective action. Thus, MIH eliminates or reduces the need for operator and/or system programmer intervention.

Missing interrupt handler is an integral part of the CP component and, as such, supports all hardware that is supported by VM/SP HPO with the exceptions cited below.

System I/O activity is monitored by MIH for any interrupts that are incomplete within a specified time interval. When MIH detects a missing interrupt, the control program (CP) attempts to correct the condition. When the corrective action attempt is completed, a record is made in the system error recording area (LOGREC) and a message is sent to alert the operator or the system programmer to take the corrective action manually or to schedule maintenance for the device where necessary.

Corrective action takes the form of simulating an error condition either to the CP I/O supervisor or to the virtual machine, whichever was the originator of the I/O operation.

To use MIH, DMKDID must be on your system and MIH must be set on. MIH can be set on by an option in your directory or by using the SET MIH command (a privilege class G command).

***Devices Monitored:*** Because interrupt timing varies widely among devices, CP monitoring has specifications for five different time intervals. This range of intervals permits flexibility in monitoring I/O activity according to an installation's own configuration and error rates where missing interrupts are a suspected cause.

The time intervals and the default time values follow:

| Devices Monitored | Default Time Values |
|---|---|
| Direct Access Storage | 15 seconds |
| Fixed Block Architecture | 15 seconds |
| Graphics Units<br>(except TYP1053 and TYP328X printers) | 30 seconds |
| Unit Record (input and Output)<br>(except TYP3800 and TYP3289E printers) | 1 minute |
| Tape Units | 10 minutes |
| Miscellaneous Devices | 12 minutes |

*Notes:*

1. *Missing interrupt handler does not support CLASTERM (terminal) devices, SNA devices, Pass-through virtual machine devices, or CLASSPEC devices.*

2. *Miscellaneous devices include:*

   - *MSS devices - includes those generated as CLASSPEC TYP3851 and as CLASDASD FEATURE=VIRTUAL or as CLASDASD FEATURE=SYSVIRT.*

   - *Graphic devices - TYP1053 and TYP328X printers.*

   - *Unit Record output devices - TYP3800 and TYP3289E printers.*

Each installation must make its own time interval settings if the default values are not compatible with its operations. In order to make the change, DMKSYS must be reassembled. In addition, a command can be used to provide for changed values for the duration of a particular initialization. Thus, when the system is reinitialized (via IPL), the DMKSYS default values would again be in effect. Note that in order to eliminate monitoring of any one or all groups of devices, any or all time values must be set to zero.

For the privilege class B user, there is the SET MITIME command with which to change the time interval settings. These values stay in effect until the system is reinitialized or another SET MITIME command is issued.

***Monitoring I/O Activity:*** When the missing interrupt handler module, DMKDID, receives control from a timer interrupt, all real device blocks are scanned. If the scan shows that the RDEVBUZY flag is on, which indicates that I/O activity is taking place or that the particular device is busy, then the RDEVMID flag is turned on. The RDEVMID flag indicates that the device is active for this time interval and that a device interrupt is pending. Both flags are reset by DMKIOT when the device causes an interrupt and if they are still on at the end of the next time interval, a missing interrupt condition exists.

Upon detecting a missing interrupt condition, a CPEXBLOK gives control to DMKDID at a later time to take further action. The DMKDID action consists of simulating an interface control check to DMKIOT. Where CP initiated the I/O, the failing device ERP is called to initiate I/O retry if possible. Where the virtual machine initiated the I/O, CP reflects the error to the virtual machine thus indicating that the operation is concluded. This initiates virtual machine retry operations. Before this action occurs, a ten-second timer is scheduled to return control to DMKDID. When DMKDID receives control, the RDEVMID bit is checked and if it is:

| RDEVMID<br>Bit Setting | Meaning |
|---|---|
| OFF (0) | Some I/O has been completed and a message sent to the operator or the system programmer to show that a missing interrupt was detected and cleared. |
| ON (1) | A message is sent to the operator or the system programmer that a missing interrupt was detected but not cleared. |

Note: *Whether the detected missing interrupt was cleared or not, a record is entered in the system error recording area (LOGREC).*

## Missing Interrupt Handler -- Summary

The missing interrupt handler (MIH) consists of resident routines in the CP nucleus. The resident modules are: DMKDID, DMKIOE, DMKIOS, DMKIOT, and DMKSYS. In addition, MIH has the following pageable modules: DMKCFJ, DMKCFP, DMKCFQ, DMKCFU, DMKCPI, and DMKCQS. A trace table entry, created for simulated interrupts, is generated by DMKDID and DMKACR. The trace table entry indicates that the interrupt is a simulated interface control check.

Note: *If the MIH Module, DMKDID, has been removed from the load list because your installation does not want or need any missing interrupt monitoring, do not use the privilege class B SET MITIME command. If you should issue the command under these circumstances, CP responds with an error message.*

***Other Error Messages and Wait State Codes:*** There are three critical phases of VM/SP CP operations where continuous system operation is vulnerable and may degenerate to wait state codes as a result of machine check or unrecoverable I/O error conditions. They are:

1. During VM/SP CP initialization
2. During system checkpoint activity
3. 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). For a description of these messages and wait state codes, see *VM/SP System Messages and 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 3-17 shows details of these areas. All numbers given are decimal. The 3031, 3032, 3033, and 308X 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.

*Note:* *Do not use the SEREP program on the 308X or 4300 processors because you will get invalid results.*

| | 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 | -- | -- |
| 135/138<br>135-3 | 256 | -- | 24<br>maximum | 64 | 176 | 186 |
| 145/148<br>145-3 | | 172 | 96<br>maximum | 64 | 176 | 186 |
| 155/158 | 155 & 158 channels do not log out | | | 64 | 176 | 186 |
| 165/168 | 165 & 168 channels do not log out | | | 64 | 176 | 186 |
| 4300-series[1] | No fixed or I/O extended logout areas | | | 64 | 176 | 186 |
| 3031<br>3032<br>3033 | -- | 172 | 640 | 64 | 176 | 186 |
| 308X<br>308X-series[1] | -- | 172 | 8 | 64 | 176 | 186 |

**Figure 3-17. VM/SP Fixed Storage and Logout Areas**

# Section 4. Additional CE Aids

This section contains information about the following:

- VM/SP CPEREP and EREP
- Using the CP SET Command Facility
  - SET RECORD Facility
  - SET MODE Facility
- TRACE Facility
  - CP TRACE Command
  - RSCS Logging
- Program Event Recording.

# VM/SP CPEREP and EREP

In order to use CPEREP, you must be in the CMS environment and have a user privilege class of C, E, or F to gain access to the records in the error recording area. When running CPEREP, you cannot include the operands on the command line, because many of them exceed the record length allowed for CMS commands. Instead, you enter the operands individually in response to prompting by the system, or you put all the operands for a single report in a separate file whose name you include on the CPEREP command line; or, you use a combination of both methods.

Detailed instructions for using CPEREP and EREP are contained in *EREP User's Guide and Reference.*

## CPEREP Error Record Retrieval

Because the VM/SP error recording area format differs from the SYS1.LOGREC data set format, the method of error record retrieval and erasure from DASD differs. To circumvent format incompatibilities, 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, calls on DMSREA to read records from the VM/SP 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.

Note: *CPEREP simulates EREP running under an OS/VS2 system, regardless of the operating system that generated the error records. Thus, the name of the error-recording data set is LOGREC and the messages are TOURIST data.*

Individual record formats in the OS/VS SYS1.LOGREC data set and the VM/SP error recording area are identical; however, VM/SP, through the medium of SVC 76, does not record on its error recording area all error record types. On the VM/SP system, errors passed to VM/SP for error recording (via SVC 76) that do not adhere to VM/SP standards are reflected back to the virtual machine to be recorded on its own error recording data set. The error record types recorded by VM/SP as opposed to the record types recorded by OS/VS and DOS/VSE operating systems are shown in *EREP User's Guide and Reference.*

Note: *Both CPEREP and EREP merely read the error records for reporting purposes. Neither is involved in writing error records, at occurrence time, to either the SYS1.LOGREC data set or the VM/SP error recording area.*

# Using the CP SET Command Facilities

## 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/SP. The SET RECORD facility 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 facility. 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, besides retry routines, error recording on another dedicated device of that virtual system. It is, therefore, conceivable that for multiple virtual machines on one VM/SP 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/SP 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/SP error recording area for that device. SET RECORD values are not retained by system checkpoint activity, so if the VM/SP 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 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 at 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, encounters 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 any previously issued SET RECORD option.

```
   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 in which processor RETRY and ECC recording are operating. This routine receives control when a user with class F privileges issues some form of the CP SET command with the MODE option. A check is initially made to determine whether or not this is VM/SP running under VM/SP. 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/SP Operator's Guide.*

The SET MODE command has five operands as follows:

**Operand** **Description**

MAIN This operand applies to processor storage bit failures that are detected and corrected by hardware logic. SET MODE MAIN is invalid for 303X and 308X processors.

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

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

RECORD This operand causes the count of soft errors to be reset to zero and the specified facility to be placed in RECORD mode; the mode in which processor RETRY and/or ECC errors are recorded.

CPUID This operand is effective only for the attached processor mode of VM/SP operation. It allows the user to apply the previously specified operands to either the main processor or the attached processor. If CPUID is not specified on the command line, then the applicable MAIN, RETRY, QUIET, and RECORD operands apply to both processors. Valid hexadecimal values for processor addresses are from 00 through 3F.

The error recording of instructions that are RETRY-corrected or ECC-corrected storage errors is determined by the setting of control register 14 bit 4.

```
ON  = RECORD MODE

OFF = QUIET MODE
```

The initial setting is a function of processor 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. Suspension of soft recording can be achieved by arriving at the threshold count or by invoking the SET MODE QUIET option. Note that the status of RECORD mode is retained by VM/SP through "warm" and "cold" start procedures (system abend conditions). For more details on soft recording, refer to "Recovery Modes" on page 3-36.

# TRACE Facilities

## TRACE Command

The CP TRACE facility of VM/SP is a very useful 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 privilege class G user 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 CP TRACE command are described in the *VM/SP 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 CP 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 *RSCS Operation and Use.*

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.

## Altering the CP TRACE Command Functions

Assume a program is loaded in the virtual processor. The virtual system then enters console function mode prior to program execution. The TRACE command function is now used with the ALL operand and the BEGIN command is invoked. The ALL operand 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 operand. 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 or 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 operand. 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, the user signals CP by means of an attention 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, PRIV, CCW, BRANCH, INSTRUCT, ALL, and CSW, or all of them.

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

The CP 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:*

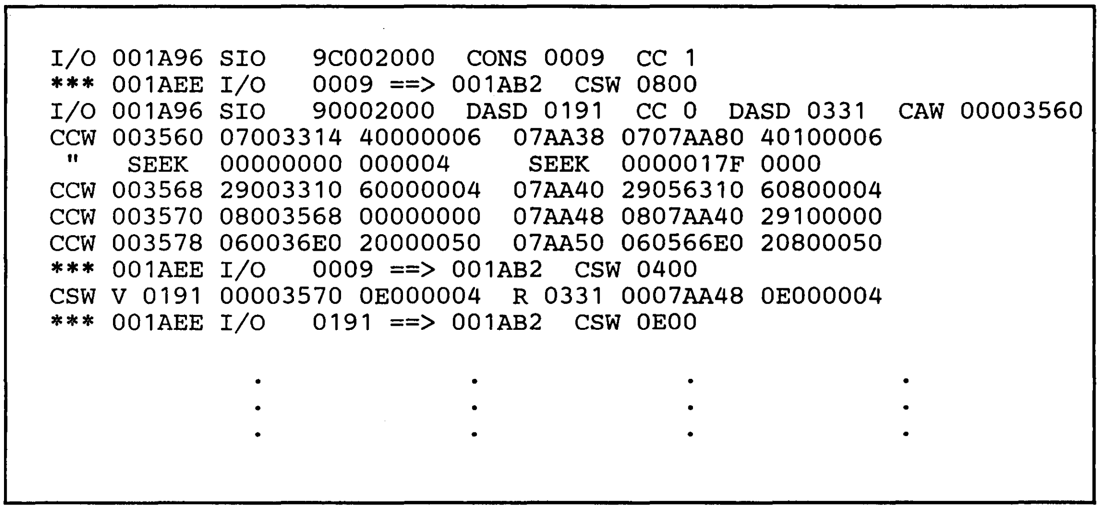
1. *A branch to the next sequential address or to the same address is not identified in the trace table.*

2. *Erroneous branch I/O, or instruction-tracing results, can be obtained when the CP TRACE command 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 4-1 shows trace data invoked by applying the CP TRACE command with the following options:

```
trace sio ccw i/o csw 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 0000
CCW 003568 29003310 60000004  07AA40 29056310 60800004
CCW 003570 08003568 00000000  07AA48 0807AA40 29100000
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 4-1. Segment of a CP TRACE Printout of a Program's I/O Operation**

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

See the TRACE command and the complete listing of the printout message formats available with this command in the *VM/SP 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.

## 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 *RSCS Operation and Use.*

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:

| **Total Bytes** | **Contents and/or Meaning** |
|---|---|
| 21 | The first 21 bytes of the log record are the first 21 bytes of the teleprocessing buffer, including BSC bytes, MULTI-LEAVING bytes (for SML only), and enough initial data bytes to fill the field. |
| 7 | For READ I/O, these are the last seven bytes of the CSW. For SML WRITE I/O, these are the first seven bytes of the SML buffer (the buffer header used internally by SML but not transmitted). For NPT WRITE I/O, these are not applicable. |
| 3 | RSCS I/O synch lock for this input/output operation. |
| 1 | This is the sense byte (if any). |
| 3 | CCW associated with the input/output operation. |

The fields of the record are separated by blanks. Figure 4-2 on page 4-10 shows the read and write log records for SML.

```
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 800000   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 -
         TELEPROCESSING BUFFER             ADDR STATUS COUNT
                                                BYTES
                                                 CSW
```

Figure 4-2. Read and Write Log Records for SML

# Program Event Recording

It is possible to monitor certain program events as they occur during program execution in the user's virtual machine by using the PER command. Trace output for the PER command is always produced after the monitored instruction executes.

## CP's PER Command

Options available with the PER command allow:

- Tracing of successful branch instructions.
- Tracing the execution of instructions that cause an alteration to general registers.
- Tracing the execution of instructions within the virtual machine.
- Tracing the execution of instructions within a virtual machine that alter storage.
- Directing the trace output to the terminal, the virtual printer, or both the terminal and virtual printer.
- Specifying a CP command or commands to be executed when a given event occurs.
- Limiting tracing for a given event type to instructions executed from within the specified range.
- Program execution to continue after trace output has completed, to stop after each trace output appears at the terminal, or to stop after a specified number of displays of trace output appear at the terminal.
- Suppressing the display of a specified number of events between displays.
- Counting execution of successful events.
- Replacing of the current traceset with a copy of the saved traceset.
- Saving a copy of the current traceset under the given name until tracing is ended or until the user logs off.
- Displaying the traceback table which contains the last six successful branch instructions on the terminal.

The format and operands of the CP PER command are described in the VM/SP *VM/SP CP Command Reference for General Users.*

The QUERY command with the PER option can be used to determine the events that are currently being traced.

## Additional Program Debugging Using PER

Two examples of using the PER command for program debugging follow. The first example uses the branch traceback table. The second example uses the PER COUNT command.

### Example 1 - Using the Branch Traceback Table

PER, used in conjunction with TRACE, can greatly reduce the difficulty of finding the cause of program interrupts. For example, if the problem is an operation exception (PROG 01), it may have been caused by a bad branch instruction.

The first step is to trace program interrupts using TRACE:

```
   trace prog
```

Run the failing program until the program interrupt occurs. When the program interrupt occurs, the address of the instruction causing the interrupt plus two is displayed. For example:

```
start
EXECUTION BEGINS...
***024602 PROG 0001 ==> 1E3D18
```

Next end TRACE and allow the program to finish. Reload the failing program and trace successful branches to the address of the bad instruction. For example:

```
per branch 24600
```

*Note:* The branch might be to an address before 24600. The branch might have encountered a valid op code. Therefore, it is sometimes necessary to use a larger branch into address. For example:

```
per branch 245F0-24600
```

When the branch to the bad instruction occurs, the branch instruction as well as the previous 5 successful branches are displayed. For example:

```
start
EXECUTION BEGINS...
==>020012 BR    07F1     024600 CC=0
TRACEBACK TABLE:
:1D1320 BR    07F3     1D125A
:1D1268 BR    07FE     1D1322
:1D1356 BNZ   4770E07C 1D139E
:1D13A2 BZ    4780E090 1D13B2
:1DFE98 BR    07FF     020000
```

*Note:* If control is transferred to the bad address by a LPSW or an interrupt (for example an SVC) PER BR does not trace this event. Therefore, it is a good idea to issue a TRACE PROG before starting the program. Then, if the program interrupt occurs before any PER output is produced, the PER TABLE command can be used to display the branch traceback table containing the last 6 successful branches. The last entry in the table is the last successful branch instruction executed before the program interrupt. While this is not necessarily the instruction causing the problem, hopefully it is near the failing instruction. It is now possible to restart the program using PER to trace the execution of instructions in the range beginning with this branch instruction, and ending at the program interrupt address.

### Example 2 - Using the PER COUNT Command

In this example, assume that there is a program loaded at location 20000 and that the program is 500 bytes (hexadecimal) in length.

Another method of finding the failing instruction is to use the PER COUNT command with TRACE. This method, as well as the use of the PER TABLE command, is well suited for problems other than just operation exceptions. If the program is abending with any sort of program exception, load the failing program, and issue the CP command:

```
trace prog
```

followed by:

```
per instruct range 20000.500
```

and then:

```
per count
```

Next start the failing program. No trace output from PER is produced while the COUNT option is in effect. When the program interrupt occurs, issue the QUERY PER command to display the current count.

```
query per
  1 INSTRUCT RANGE 020000-0204FF TERMINAL NORUN
    PER COUNT 2159
```

This means that 2159 instructions were executed before the instruction that caused the program interrupt. It is now possible to trace as many instructions leading up to the program interrupt as desired. To trace the last 15 instructions before the program interrupt, reload the failing program, and issue the following PER command:

```
per pass 2144
```

the response is:

```
PER COUNT 2159
PER COUNT ENDED
```

This command has two effects. First, it turns off the PER COUNT option, and second it applies the PASS option to the current traceset. The current traceset now contains:

```
1 INSTRUCT RANGE 020000-0204FF TERMINAL NORUN PASS 2144
```

Next start the failing program. The first 2144 instructions executed in the range 20000-204FF are not displayed. The 2145th instruction is displayed. When the instruction is displayed, issue:

```
per pass
```

This command resets the PASS option to the default (display every instruction). The current traceset now contains:

```
1 INSTRUCT RANGE 020000-0204FF TERMINAL NORUN
```

It is now possible to trace the last 15 instructions, and to use the DISPLAY command to display storage and register contents.

PER COUNT can also be used in conjunction with more specific trace elements to produce the desired results. For example, if a problem occurs as a result of the execution of an SVC 202 and the failing program issues many SVC 202s before failing, it may not be productive to use TRACE.

An alternative is to use PER to set up a traceset that traces only SVC 202s (op code X'0ACA') and to use PER COUNT to count the occurrences. First, load the failing program and then issue:

```
per instruct 0aca range 20000.500
per count
```

and start the program. When the failure occurs, issue a QUERY PER to check the count.

```
query per
  1 INSTRUCT 0ACA RANGE 020000-0204FF TERMINAL NORUN
    PER COUNT 623
```

The program can then be traced after using the PER PASS option as above to get close to the problem.

# IPCSDUMP/PRTDUMP

System abend (abnormal termination) conditions can be prompted by real System/370 system operator intervention involving PSW restart. System abend conditions can also be caused by program SVC 0 operation. This may happen when CP is in a program predicament that 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 not contain a trace table with the sequence of events that occurred just before the condition that caused the abend. 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/SP 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.*

### Automatic Spooling of Abend Dump Files

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

The system dump file (previously spooled to a DASD unit) can then be processed and formatted by the IPCSDUMP 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 PRTDUMP command formats and/or prints the symptom record on the first page with a disk dump file previously processed by IPCSDUMP. The IPCSDUMP command and the PRTDUMP command are described fully in the *VM/SP Interactive Problem Control System Guide.*

### Data Control Blocks that Define System Fault Cause

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

The RDEVBLOK, IOBLOK, and IOERBLOK relationships are illustrated in Figure 3-2 on page 3-10 and Figure 3-3 on page 3-10.

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 the interval timer values of the processor that caused the abend.

3. For 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/SP 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 abend are indicated by an adjacent display of an active IOBLOK.

   *Note:* If RDEVBLOK 200 is for an FB-512 device, the RECBLOK would be in a different format. For the actual format of an FB-512 RECBLOK, see *VM/SP Data Areas and Control Block Logic Volume 1 (CP).*

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, the PRTDUMP output indicates this.

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 termination.

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 (VMBLOK), virtual channel blocks (VCHBLOK), virtual control unit blocks (VCUBLOK), virtual device blocks (VDEVBLOK), and virtual console control blocks (VCONCTL). These are followed by Segment tables, Page tables, and Swap tables (SEGTABLE, PAGTABLE, SWPTABLE), respectively, that are applicable to the associated user's virtual machine activity.

## DUMPSCAN

DUMPSCAN creates an environment that lets you interactively inspect dumps, formatted as CMS files by IPCSDUMP. DUMPSCAN prompts you for the dump filename and filemode. Once the dump is located, subcommands can be entered.

Using DUMPSCAN to look at the dump processed by the IPSCDUMP command you can:

- Display:

  – Any chosen area specified directly (or indirectly) by its address.

  – Registers, PSWs, timers, and clocks.

  – The address in a chain of homogeneous control blocks.

  – Any module or entry point by entry name.

  - The symptom record.
- Locate:
  - A string of hexadecimal or EBCDIC data between two addresses.
  - The module name containing a given address.
- Print:
  - The displayed data resulting from the subcommand.

In addition, dump dependent subcommands may be available to allow other functions. For example:

- In a CP dump, you can display:
  - The trace table entries, by number of entries, and starting address.
  - Real and virtual device control blocks by device address.
  - A list of all logged-on users with their VMBLOK addresses and status.
  - Formatted information from a selected user's VMBLOK.
  - The formatted contents of the CORTABLE entry for any real address.

The DUMPCSAN command and subcommands are described in *VM/SP Interactive Problem Control System Guide*.

# Appendix A. Using EREP With VM

The following EXEC can be used to perform these system support functions for a privilege class F user.

1. Emergency offload of the error recording area (ERA) onto a tape (the EREP history file).

2. Generate a system summary and clear/reset the error recording area.

3. Generate some reports from the EREP history file.

*Notes:*

1. *use this EXEC can create large amounts of printed output, depending on the size of the EREP history file, review the options specified before using it. See EREP User's Guide and Reference.*

2. *If you specify or imply ACC=Y for an EREP run, CPEREP rewinds tape 181, spaces forward over the existing file, and backspaces over the tape mark before writing and records to the file. Therefore, if a tape is to be used for the first time, you should write a tape mark (you can use the CMS TAPE command) at the beginning of the tape before invoking CPEREP.*

```
TRACE E;
ADDRESS COMMAND;
/* Find/preserve existing status of virtual printer */
'CONWAIT';
'DESBUF';
'EXECIO * CP ( FIFO STRING QUERY VIRTUAL 00E';
PULL 'CL' prtclass prthold 'COPY' prtcopy 'FORM' prtform .;
PULL '00E' prttofor prtuser 'DIST' prtdist .;
'CONWAIT';
'DESBUF';
'CP SPOOL PRINTER NOCONT CLOSE';
'CP SPOOL PRINTER CONT OFF DIST IBMCE CLASS A HOLD FORM STANDARD COPY 1';
/* Get/format a temporary minidisk for work files */
tccu = GETTDSK();
tfm = FREEDISK();
fm3 = tfm||'3';
fm4 = tfm||'4';
PUSH 'SCRTCH';
PUSH 'YES';
'FORMAT' tccu tfm;
/*  Issue FILEDEFs for DDNAMES required by EREP */
'FILEDEF EREPPT PRINTER (NOCHANGE BLOCK 133 PERM';
'FILEDEF SYSIN DISK SYSIN EREPWORK' fm3 '(NOCHANGE PERM';
'FILEDEF SERLOG DISK SERLOG EREPWORK' tfm '(NOCHANGE BLOCK 4096 PERM';
'FILEDEF TOURIST TERMINAL (NOCHANGE BLOCK 133 PERM';
'FILEDEF DIRECTWK DISK DIRECTWK EREPWORK' fm4 '(NOCHANGE PERM';
'FILEDEF ACCDEV TAP1 (NOCHANGE BLOCK 12000 RECFM VB DEN 1600 PERM';
'FILEDEF ACCIN TAP2 (NOCHANGE BLOCK 12000 RECFM VB DEN 1600 PERM';
/* Make sure that tape is available */
'CP QUERY VIRTUAL 181';
IF rc ¬= 0 THEN SIGNAL notape;
'TAPE MODESET ( DEN 1600';
'CP REWIND 181';
/* Function 1 - Offload ERA data to tape */
/* Function 2 - Generate a system summary and clear/reset the ERA */
'CONWAIT';
'DESBUF';
QUEUE ' ACC=Y SYSUM=Y ZERO=Y ';
QUEUE '';
'EXEC CPEREP';
'CP DEFINE 181 182';
/* Function 3 - Generate some additional reports */
QUEUE ' ACC=N HIST=Y TABSIZE=512K SYSEXN=Y ';
QUEUE '';
'EXEC CPEREP';
EXIT;
QUEUE ' ACC=N HIST=Y TABSIZE=512K TRENDS=Y ';
QUEUE '';
'EXEC CPEREP';
QUEUE ' ACC=N HIST=Y TABSIZE=512K EVENT=Y ';
QUEUE '';
'EXEC CPEREP';
QUEUE ' ACC=N HIST=Y TABSIZE=512K PRINT=AL ';
QUEUE '';
'EXEC CPEREP';
SIGNAL cleanup;
gettdsk: /* Get a temporary disk equivalent to 20 cylinders of 3330 */
  /* Establish DASD types */
  PUSH 'XXXX'; /* End of list marker */
  PUSH '2314 35';
  PUSH '3310 8360';
  PUSH '3330 20';
  PUSH '3340 50'
  PUSH '3350 10'
  PUSH '3370 8360'
  PUSH '3375 14'
  PUSH '3380 10'
  PULL dasdtype dasdamt .;
  DO UNTIL dasdtype = 'XXXX';
    DO i = 1 to 599 by 1;
      'CP DEFINE T'||dasdtype i dasdamt;
      IF rc = 0 THEN RETURN(i);
    END;
    PULL dasdtype dasdamt .;
  END;
freedisk: /* Find first available filemode */
  PUSH '$'; /* End of list marker */
  PUSH 'Z';
  PUSH 'X';
  PUSH 'W';
  PUSH 'V';
  PUSH 'U';
  PUSH 'T';
  PUSH 'R';
  PUSH 'Q';
  PUSH 'P';
  PUSH 'O';
  PUSH 'N';
  PUSH 'M';
  PUSH 'L';
  PUSH 'K';
  PUSH 'J';
  PUSH 'I';
  PUSH 'H';
  PUSH 'G';
  PUSH 'F';
  PUSH 'E';
  PUSH 'D';
  PUSH 'C';
  PUSH 'B';
  PULL xfm .;
  DO UNTIL xfm = '$';
    'QUERY DISK' xfm '( LIFO )';
    PULL 'NOT' accstat .;
    IF accstat = 'ACCESSED.' THEN RETURN(xfm);
    PULL .;
    PULL xfm .;
  END;
notape:
  SAY "There is no tape available for the EREP accumulation data.";
  SAY "Have the EREP accumulation/history tape attached to you and try again.";
cleanup: /* Clean up time */
  'CONWAIT';
  'DESBUF';
  'RELEASE' tfm '( DET )';
  'FILEDEF EREPPT CLEAR';
  'FILEDEF SYSIN CLEAR';
  'FILEDEF SERLOG CLEAR';
  'FILEDEF TOURIST CLEAR';
  'FILEDEF DIRECTWK CLEAR';
  'FILEDEF ACCDEV CLEAR';
  'FILEDEF ACCIN CLEAR';
  'CP SPOOL PRINTER NOCONT';
  'CP CLOSE PRINTER NAME CPEREP REPORTS';
  'CP DEFINE 182 181';
  'CP REW 181';
  EXIT;
```

# Index

SC19-6205-3

VM/SP OLTSEP and Error Recording Guide (File No. S370/4300-37) Printed in U.S.A. SC19-6205-3

VM/SP OLTSEP and Error
Recording Guide
SC19-6205-3

**READER'S COMMENT FORM**

This manual is part of a library that serves as a reference source for systems analysts, programmers, and operators of IBM systems. You may use this form to communicate your comments about this publication, its organization, or subject matter, with the understanding that IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you.

Your comments will be sent to the author's department for whatever review and action, if any, are deemed appropriate. Comments may be written in your own language; English is not required.

**Note:** *Copies of IBM publications are not stocked at the location to which this form is addressed. Please direct any requests for copies of publications, or for assistance in using your IBM system, to your IBM representative or to the IBM branch office serving your locality.*

| | *Yes* | *No* |
|---|---|---|
| • Does the publication meet your needs? | □ | □ |
| • Did you find the material: | | |
| Easy to read and understand? | □ | □ |
| Organized for convenient use? | □ | □ |
| Complete? | □ | □ |
| Well illustrated? | □ | □ |
| Written for your technical level? | □ | □ |

- What is your occupation? ______________________________

- How do you use this publication:

| | | | |
|---|---|---|---|
| As an introduction to the subject? | □ | As an instructor in class? | □ |
| For advanced knowledge of the subject? | □ | As a student in class? | □ |
| To learn about operating procedures? | □ | As a reference manual? | □ |

**Your comments:**

*If you would like a reply, please supply your name and address on the reverse side of this form.*

Thank you for your cooperation. No postage stamp necessary if mailed in the U.S.A. (Elsewhere, an IBM office or representative will be happy to forward your comments or you may mail directly to the address in the Edition Notice on the back of the title page.)

Note: Staples can cause problems with automated mail sorting equipment.
Please use pressure sensitive or other gummed tape to seal this form.

SC19-6205-3

## Reader's Comment Form

Fold and Tape

Please Do Not Staple

Fold and Tape

NO POSTAGE NECESSARY IF MAILED IN THE UNITED STATES

**BUSINESS REPLY MAIL**

FIRST CLASS PERMIT NO. 40 ARMONK, N.Y.

POSTAGE WILL BE PAID BY ADDRESSEE:

International Business Machines Corporation
Department G60
P. O. Box 6
Endicott, New York 13760

Fold

Fold

If you would like a reply, *please print:*

*Your Name* ____________________
*Company Name* ____________________ *Department* __________
*Street Address* ____________________
*City* ____________________
*State* ____________________ *Zip Code* __________
*IBM Branch Office serving you* ____________________

IBM®

Cut or Fold Along Line

SC19-6205-3

VM/SP OLTSEP and Error Recording Guide (File No. S370/4300-37) Printed in U.S.A. SC19-6205-3

SC19-6205-3