

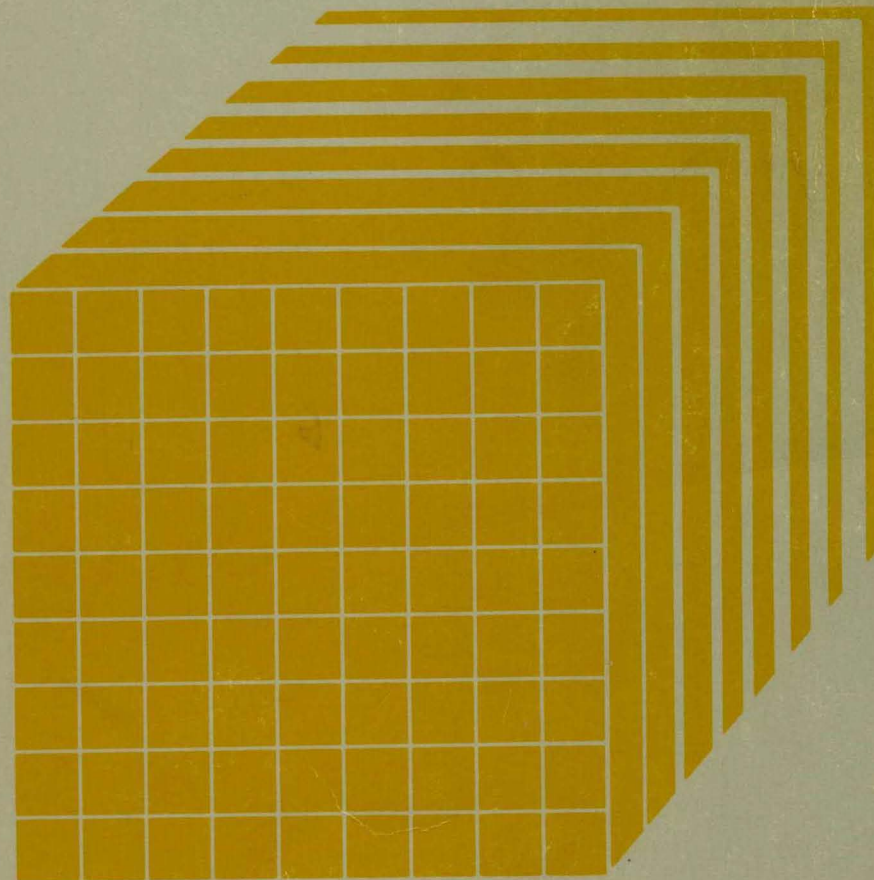
IBM

Virtual Machine/
System Product

Operator's Guide

Release 4

SC19-6202-3



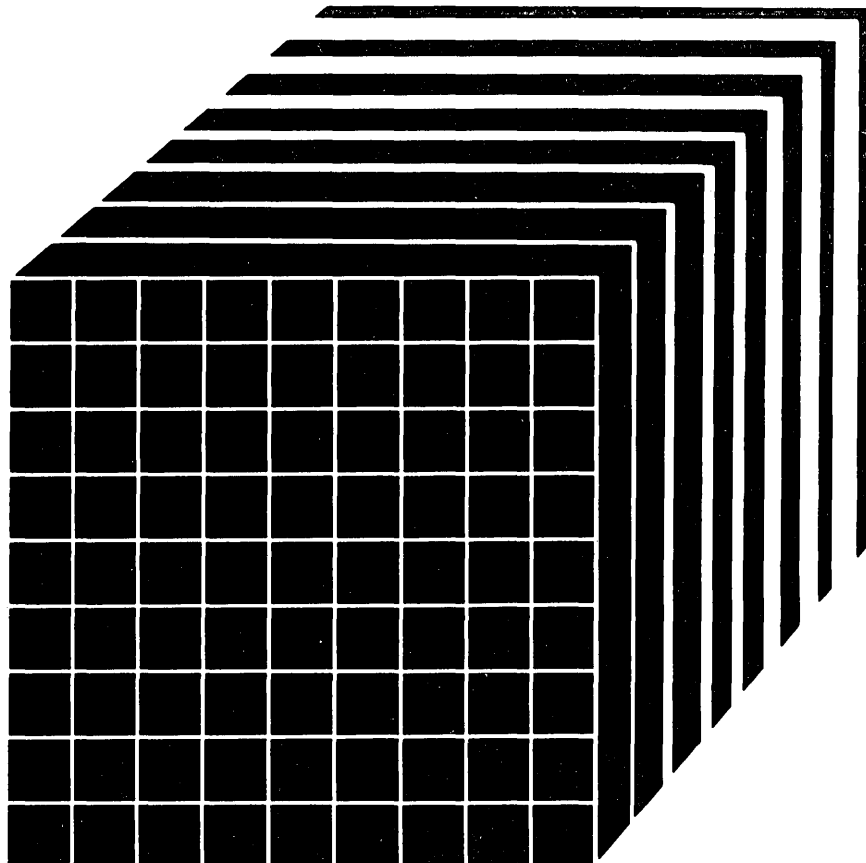


Virtual Machine/ System Product

Operator's Guide

Release 4

SC19-6202-3



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

| Fourth Edition (December 1984)

| This edition, SC19-6202-3, is a major revision of SC19-6202-2, 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 list of changes, see the following page.

Changes or additions to 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 product may be used. Any functionally equivalent program may be used instead.

Ordering Publications

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

Summary of Changes for SC19-6202-3 for VM/SP Release 4

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 ST00-1351
- Release 2, order SQ19-6202.

Hardware Support

Information was added for the support of:

- The 3290 Information Panel.
- The 4361 Model Group 4 and 5 Processors.
- The 4381 Model Group 1 and 2 Processors.
- The 3370 Direct Access Storage Models A2 and B2.
- The 4248 Printer.
- The 3800 Model 3 Printing Subsystem - Existing programs designed to produce 3800 Model 1 printer output may produce output for the 3800 Model 3 printer with little or no program change. Use of this support provides improved print quality (240 x 240 pel resolution) and the addition of a 10 line-per-inch (LPI) vertical space option.
- The 3480 Magnetic Tape Subsystem.

Program Support

Saved System 8 M Byte Limit Removal

Changed: SAVESYS, VMSAVE and IPL functions have been modified to let a page image copy of up to a 16 M byte virtual machine be saved and restored. Also, the NAMESYS MACRO was changed to enforce the 16 M byte limit.

Shared/Nonshared Restriction

Changed: Any attempt to construct a virtual device configuration that would mix SHARED and NONSHARED device types on the same virtual control unit is rejected. To permit migration of data on 3420 tapes (a shared device) to the 3480 (a nonshared device), this restriction will not be enforced.

VM/SP Interactive Problem Control System

Changed: The base VM/IPCS component is enhanced to include the functional equivalent to the VM/IPCS Extension licensed program.

System Initialization

Changed: A new operand was added to the SHUTDOWN command.

Stand-Alone Dump

New: This facility enhances serviceability by letting support personnel dump up to 16 M of real storage.

Programmable Operator

Changed: This facility can now be used in a mixed environment and the commands can be used from an NCCF operator station. Slight changes were added to the LOADTBL and QUERY commands, and a new command, LGLOPR, was added.

CPTRAP Command

Changed: Two new operands were added to the CPTRAP command: Groupid and Wrap.

Miscellaneous

Changed Format: The format of the command descriptions in "Section 3. CP Commands" on page 3-1 has changed to include a "When to Use" section. Former usage notes have now become a part of the operand description or have been included in a "Things You Should Know" section. This new organization is not marked with revision bars since it effects all of Section 3.

Summary of Changes for SC19-6202-2 for VM/SP Release 3

Program Support

New: A "text" option was added to the SET command to enhance the capability of the LOGMSG operand.

Information was added for using the MSGNOH command.

Improved functions for the programmable operator facility are explained.

Hardware Support

New: Information was added for the support of the 3262, 3289-E, and 4245 printers and the 3430 tape drive.

The IBM 3088 Multisystem Communication Unit interconnects multiple systems using block multiplexer channels. The 3088 uses an unshared sub-channel for each unique address and is fully compatible with existing channel-to-channel protocol.

Miscellaneous

Changed: The information for the INDICATE, NETWORK, QUERY, and SET commands was reorganized to make it easier to determine the operands that can be issued by each privilege class user and to locate information about the command operands and system responses.

Various minor technical and editorial changes have been made throughout the publication.

**Summary of Changes
for SC19-6202-1
as Updated by SN24-5735
for VM/SP Release 2**

Program Support

New: The missing interrupt handler 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 the correction action was successful.

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

Miscellaneous

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

**Summary of Changes
for SC19-6202-1
for VM/SP Release 2**

Programmable Operator Facility

New: This facility provides the capability to: log messages, suppress messages, redirect messages, execute messages, or preprogram message responses. The capabilities are under control of an editable message routine table in a CMS file.

Inter-User Communication Vehicle (IUCV) enhancements for message handling are also included.

CMS Nucleus Restructure, and Removal of the CMS Tokenization Eight-Byte Restriction

New: The restructured nucleus provides a CMS system that is more flexible and extendable for development, serviceability and maintenance purposes.

The eight-byte tokenize restriction has been removed for parameter passing.

Trace Table Recording Facility

New: This facility allows service personnel and system programmers to create a chronological READER spool file of CP trace entries by type, VMBLOK address, interrupt code, and device address.

Miscellaneous

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

Preface

This book is for those persons responsible for the operation and administration of a Virtual Machine/System Product (VM/SP) system.

This operator's guide describes some of the hardware of the System/370 as well as some user virtual machine requirements. For general information on how to run batch jobs in VM/SP virtual machines (other than CMS), refer to the *Virtual Machine/System Product: Running Guest Operation Systems*, GC19-6212.

To manage VM/SP resources effectively, a system operator should be familiar with the operation and hardware requirements of System/370, 4300, and 303X input/output devices and should know the broad concepts of computer operations. In addition, the system operator should be familiar with the use and capability of the Interactive Problem Control System component of VM/SP. Details of this major component are in the *VM/SP Interactive Problem Control System Guide*, SC24-5260.

Note: For VM/SP users, VM/SP IPCS is more effective than the IPCS Extension Program Product (5748-SA1).

If your installation uses Remote Spooling Communications Subsystem (RSCS) Networking, the system operator is generally the operator of the RSCS virtual machine as well as of the real machine at the installation. For details on how to operate the RSCS virtual machine, refer to the *VM/SP Program Reference and Operations Manual*, SH24-5005.

Note: For more effective use of RSCS in VM/SP, the RSCS Networking Program Product (5748-XP1) is recommended.

To help the system operator gain insight into virtual machine concepts and capabilities, refer to the *Virtual Machine/System Product: Introduction*, GC19-6200. This book briefly discusses virtual storage, virtual machine features and facilities of VM/SP.

This publication describes VM/SP system startup, logon, VM/SP commands, special service programs, user privilege classes, as well as other data necessary to manage the resources of the system on a daily basis. Operator commands and other operator related tasks for the Group Control System (GCS) in *VM/SP Group Control System Guide*, SC24-5249.

There are five sections in this publication. The first section discusses the Operational control of VM/SP under normal and adverse conditions. The remaining four sections are the "how-to-do-it" sections. Wherever possible, subject matter adheres to an alphabetic arrangement. The beginning of Section 3, "CP

Commands” on page 3-1 lists the operator commands and summarizes what each command does.

Terminology

The user privilege classes referred to in this book are IBM-defined classes. If your installation restructures the classes, see your installation administrator.

The following terms in this publication refer to the indicated support devices:

- “2305” — IBM 2305 Fixed Head Storage, Models 1 and 2.
- “270x”
 - IBM 2701, 2702, and 2703 Transmission Control Units
 - The Integrated Communications Adapter (ICA) on some IBM CPUs.
- “3330”
 - IBM 3330 Disk Storage, Models 1, 2, or 11
 - IBM 3333 Disk Storage and Control, Models 1 or 11
 - 3350 Direct Access Storage operating in 3330/3333 Model 1 or 3330/3333 Model 11 compatibility mode.
- “3340”
 - IBM 3340 Disk Storage, Models A2, B1, and B2.
 - IBM 3344 Direct Access Storage Model B2.
- “3350” — IBM 3350 Direct Access Storage Models A2 and B2 in native mode.
- “FB-512” or “FBA” — IBM 3310 and 3370 Direct Access Devices.
- “3289” — IBM 3289 Model 4 Printer.
- “3262” — IBM 3262 Printer, Models 1 and 11.
- “3704,” “3705,” “3725,” or “37xx” — IBM 3704, 3705, and 3725 Communications Controllers.
- “3705” — 3705 I and the 3705 II, unless otherwise noted.
- “2741” — IBM 2741 and the 3767, unless otherwise specified.
- “3800” — IBM 3800 Printing Subsystems, Models 1, 3, and 8. A specific device type is named only when a distinction is required between device types.
 - “3800 Model 3” — IBM 3800 Models 3 and 8, unless otherwise stated. (The IBM 3800 Model 8 is available only in selected world trade countries.)

- “3850” — IBM 3850 Mass Storage System.
- “3270” — A series of display devices, namely the IBM 3275, 3276, 3277, 3278 and 3279 Display Stations, and the 3290 Information Panel. A specific device type is named only when a distinction is needed among device types.
- “3370” — IBM 3370 Direct Access Storage Models A1, A2, B1, and B2.
- “3375” — IBM 3375 Direct Access Storage Device.
- “3380” — IBM 3380 Direct Access Storage device. Information on the IBM 3380 Direct Access Storage device is for planning purposes only until the availability of the product.

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 extended count-key-data channel programs.

- “3081” — IBM 3081 Processor complex.
- “3033” — all 3033 processor models including the IBM 3033 Model Group S processors. Information about the IBM 3033 Model Group S processors is for planning purposes only until the availability of the product.
- “3880” — IBM 3880 Storage Control Unit.
- “3480” — IBM 3480 Magnetic Tape Subsystem.
- “4248” — IBM 4248 printer.
- “Attached processors” also refers to multiprocessors unless otherwise indicated.
- “VSM” refers to the VTAM service machine, a service virtual machine that lets SNA (Systems Network Architecture) terminals access the VM/SP system.
- “VSE” refers to the combination of the DOS/VSE system control program and the VSE/Advanced Functions program product. Also includes the VSE/System Product.
- “DOS,” in certain cases, 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.
- “CMS/DOS” continues to be used when referring to the DOS-like simulation environment provided under the CMS component of the VM/System Product.

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

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

For the relationship of the *Virtual Machine/System Product: Operator's Guide* to other VM/SP prerequisite and corequisite publications, refer to "The VM/SP Library."

Prerequisite Publications

Virtual Machine/System Product:

Introduction, GC19-6200

Terminal Reference, GC19-6206

CMS User's Guide, SC19-6210

If the 3767 Communications Terminal is used as the alternate system console, the *IBM 3767 Operator's Guide*, GA18-2000, is also a prerequisite.

The *Virtual Machine/System Product: CMS User's Guide* is a prerequisite only if the operator edits files used for system control or accounting purposes.

If the VTAM Communications Network Application (VM/VCNA) product is used, the following publications are prerequisites:

Communication Network Application General Information Manual, GC27-0501

Communication Network Application Installation, Operation, and Terminal Use, SC27-0502.

If the ACF/VTAM Release 3 (ACF/VTAM Version 3 for VM/SP with VM SNA Console Support (VSCS)) product is used, the following publications are prerequisites:

ACF/VTAM General Information (for VM/SP), GC30-3246

ACF/VTAM Network Program Products Planning (MVS Base), SC23-0110

ACF/VTAM Installation and Resource Definition (MVS Base), SC23-0111

ACF/VTAM Customization (MVS Base), SC23-0112

ACF/VTAM Operation (MVS Base), SC23-0113.

Corequisite Publications

Virtual Machine/System Product:

Planning Guide and Reference, SC19-6201

CMS Command and Macro Reference, SC19-6209

CP Command Reference for General Users, SC19-6211

System Product Editor User's Guide, SC24-5220

Group Control System Guide, SC24-5249

System Product Editor Command and Macro Reference, SC24-5221

Installation Guide, SC24-5237

System Product Interpreter User's Guide, SC24-5238

System Product Interpreter Reference, SC24-5239

Release 4 Guide, SC24-5240

Distributed Data Processing Guide, SC24-5241

System Logic and Problem Determination Guide Volume 1 (CP), LY20-0892

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

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

Device Support Facility User's Guide and Reference, GC35-0033

If your installation has a 3850 Mass Storage System, the following are corequisite publications:

IBM 3850 Mass Storage System (MSS): Principles of Operation: Theory, GA32-0035

IBM 3850 Mass Storage System (MSS): Principles of Operation, GA32-0036

IBM 3850 Mass Storage System (MSS): Introduction and Preinstallation Planning, GA32-0038

Operators Library: IBM 3850 Mass Storage System (MSS) Under OS/VS, GC35-0014

OS/VS Message Library: Mass Storage System (MSS) Messages, GC38-1000.

Information on commands that control the virtual machine and its resources applicable to the G privilege class user are contained in the following publications:

Virtual Machine/System Product:

CP Command Reference for General Users, SC19-6211

Running Guest Operating Systems, GC19-6212

System Programmer's Guide, SC19-6203

System Messages and Codes, SC19-6204

OLTSEP and Error Recording Guide, SC19-6205.

The latter publication contains information on the VM/SP error recording process. It also describes briefly the use of CPEREP, the CMS command that uses EREP for editing and printing SYS1.LOGREC and VM/SP error recording records. Detailed information on the use of the operands of the CPEREP command is contained in *Environmental Recording Editing and Printing (EREP) Program*, GC28-1178.

Other corequisite publications include:

2821 Control Unit Component Description, GA24-3312

IBM 3211 Printer 3216 Interchangeable Train Cartridge & 3811 Printer Control Unit Comp. Desc. and Operator's Guide, GA24-3543

3262 Printer Models 1 and 11 Component Description, GA24-3733

Virtual Machine/System Product: Library Guide, Glossary, and Master Index, GC19-6207

Virtual Machine/System Product: Program Reference and Operations Manual, SH24-5005

3800 Printing Subsystem Programmer's Guide, GC26-3846

OS/VS and DOS/VSE Analysis Program-1 (AP-1) User's Guide, GC26-3855

VM/SP Interactive Problem Control System Guide, SC24-5260

The following are corequisite publications if your installation has a 3704, 3705 or 3725:

Guide to Using the IBM 3704 Communications Controller Control Panel, GA27-3086

Guide to Using the 3705 Control Panel, GA27-3087.

ACP/NCP-SSP, V3 Installation and Resource Definition Guide, SC30-3253.

EP/3725 Installation and Resource Definition Guide and Reference, SC30-3172.

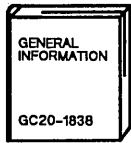
EP/3705 Generation and Utilities Guide and Reference, GC30-3242.

ACF/NCP V4, ACF/SSP V3 Diagnosis Guide, SC30-3255.

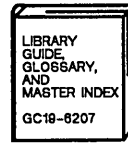
Note: References in the text to prerequisite and corequisite VM/SP publications will be given in abbreviated form.

The VM/SP Library

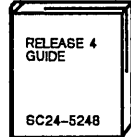
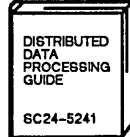
Evaluation



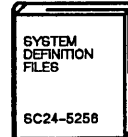
Index



Planning



Installation



Administration



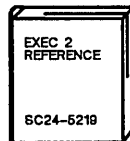
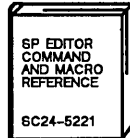
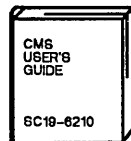
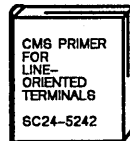
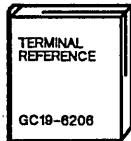
Operation



Applications

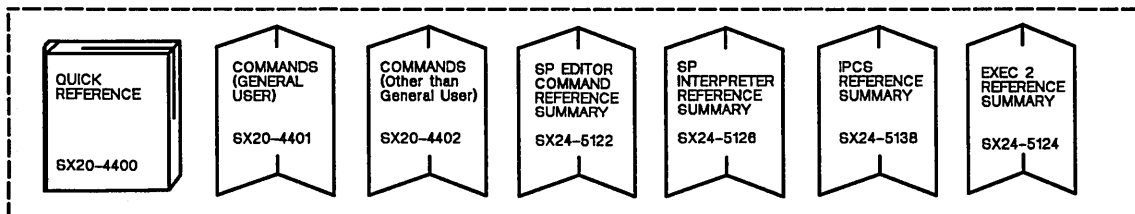


End Use

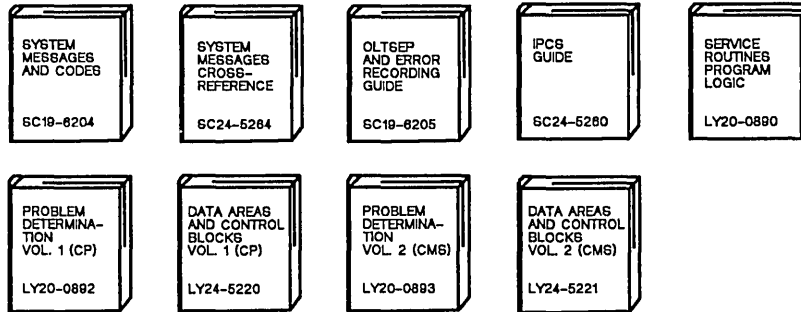


Reference Summaries

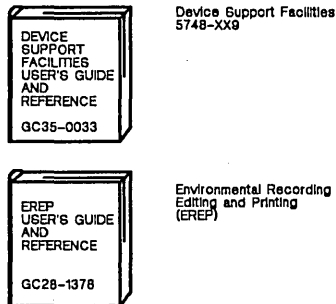
To order all the Reference Summaries, use order number 380F 3221



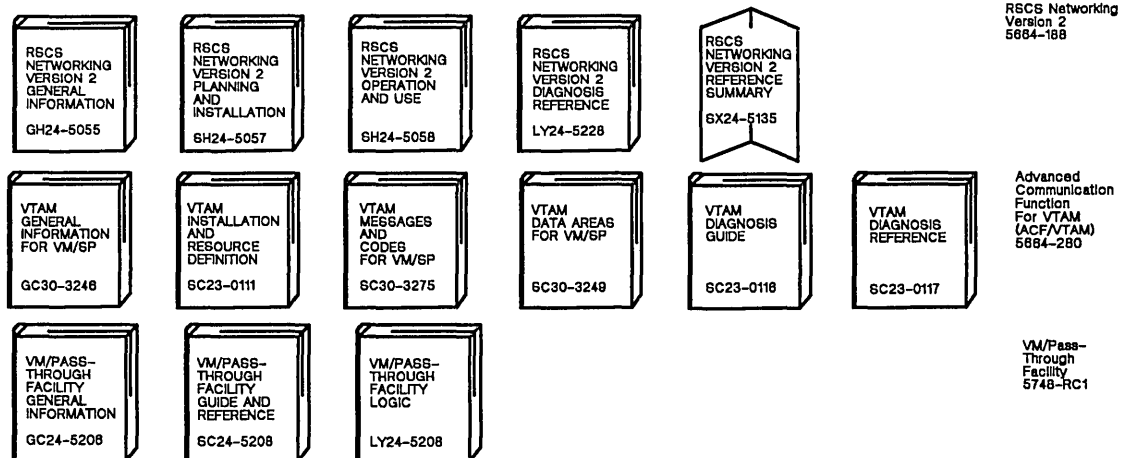
Diagnosis



Auxiliary Service Support



Auxiliary Communication Support



Contents

Section 1. Introduction to Operational Control of the VM/SP System	1-1
System/370 recovery management support	1-2
Machine Check Handler (MCH)	1-3
Channel Check Handler (CCH)	1-4
Missing Interrupt Handler	1-4
I/O Error Recording and SVC 76	1-4
VM/SP Recovery Features	1-5
Recording Facilities	1-5
VM/SP Repair Facilities	1-5
VM/SP Restart Facilities	1-6
Section 2. Starting VM/SP Operation	2-1
Using CP and CMS	2-1
Console Procedures	2-1
Operating Procedures (Operator Tasks)	2-2
Labeling Disks	2-2
System Definition	2-3
System Initialization	2-3
Mass Storage System Initialization	2-14
Multiprocessor Initialization	2-16
Attached Processor Initialization	2-16
Console Definition	2-17
Disabling the VM/SP Primary Console	2-19
Selecting a VM/SP Alternate Console during VM/SP Initialization	2-19
Controlling the 370x Communications Controller	2-20
Considerations for Loading All 370x Control Programs	2-21
Special Considerations for Loading the EP 370x Control Program	2-21
Messages On Startup and Initialization	2-22
Control Messages after Startup	2-22
System Operation Status Messages	2-22
Messages from Virtual Machine Users	2-22
Hardware and Program Status Messages	2-22
3480 Magnetic Tape Subsystem Support	2-24
3800 Printing Subsystem Support	2-24
Dedicated 3800 Printing Subsystem Support	2-25
Performance Options	2-26
Reserved Page Frames Option	2-27
Favored Execution Option	2-27
Virtual=Real Option	2-28
Locked Pages Option	2-31
Priority	2-31
VM/370 Assistance - Hardware Assist	2-32

Affinity	2-34
Queue Drop Elimination	2-35
System Termination	2-35
System Abend Dumps	2-37
Section 3. CP Commands	3-1
Privilege Classes for CP Commands	3-1
How to Use this Chapter	3-3
ACNT	3-6
ATTACH	3-8
AUTOLOG	3-15
BACKSPAC	3-17
CHANGE	3-20
CPTRAP	3-25
DCP	3-31
DEFINE	3-35
DETACH	3-37
DISABLE	3-41
DMCP	3-43
DRAIN	3-47
ENABLE	3-49
FLUSH	3-51
FORCE	3-54
FREE	3-56
HALT	3-58
HOLD	3-59
INDICATE	3-61
LOADBUF	3-71
LOCATE	3-76
LOCK	3-78
MESSAGE	3-82
MIGRATE	3-84
MONITOR	3-86
MSGNOH	3-100
NETWORK	3-102
ORDER	3-113
PURGE	3-115
QUERY	3-117
QVM	3-151
REPEAT	3-152
SAVESYS	3-154
SET	3-155
SHUTDOWN	3-169
SPACE	3-171
SPMODE	3-172
SPTAPE	3-173
START	3-179
STCP	3-184
TRANSFER	3-186
UNLOCK	3-189
VARY	3-191
WARNING	3-195

Section 4. VM/SP Service Programs	4-1
Controlling the 3704/3705/3725 Communications Control Program	4-1
Using CMS To Alter VM/SP Files	4-2
Other CMS Commands	4-2
DASD Dump Restore (DDR) Service Program and How to Use It	4-2
Invoking DDR under CMS	4-3
Invoking DDR as a Standalone Program	4-4
DDR Control Statements	4-4
I/O Definition Statements	4-5
INPUT/OUTPUT Control Statement	4-5
SYSPRINT Control Statement	4-7
Function Statements	4-8
PRINT/TYPE Function Statement	4-14
Maintaining the 3800 Image Library	4-19
The Image Library	4-19
The GENIMAGE Service Routine	4-19
The IMAGELIB Service Routine	4-21
The IMAGEMOD Command	4-21
Device Support Facilities	4-23
Formatting Volumes--General Information	4-24
Format/Allocate Service Program (DMKFMT)	4-24
Format/Allocate Program Card Input	4-29
Format/Allocate Console Input	4-35
How To Print a CMS Dump File	4-40
How To Print a CP ABEND Dump from Tape	4-40
Network Dump Operations	4-41
NCPDUMP Service Program and How To Use It	4-42
ZAP Service Program and How To Use It	4-43
ZAP Input Control Records	4-45
Special Considerations for Using the ZAP Service Program	4-52
ZAPTEXT Service Program	4-52
ZAPTEXT Input Control Records	4-53
EXPAND Command	4-54
The Programmable Operator Facility	4-56
Use in a Single System	4-57
Use in a Distributed VM Environment	4-57
Use in a Mixed Environment	4-58
Invoking the Programmable Operator Facility	4-59
Invocation of Programmable Operator Facility Commands	4-61
Issuing Commands in the Local Environment	4-62
Issuing Commands In The Distributed VM Environment	4-63
Issuing Commands in a Mixed Environment	4-65
Helpful Hints	4-67
The Log File	4-67
Ensuring A Complete Log	4-68
Programmable Operator Facility Commands	4-70
CMD Command	4-71
FEEDBACK Command	4-73
GET Command	4-74
LGLOPR Command	4-75
LOADTBL Command	4-77
LOG Command	4-79

QUERY Command	4-80
SET Command	4-83
STOP Command	4-86
Stopping the Programmable Operator Facility	4-86
Stopping the PMX	4-87
Problem Determination - DEBUG MODE	4-87
Stand-Alone Dump Facility	4-88
Overview	4-88
Devices that You Can Use to IPL Stand-Alone Dump	4-89
Devices to which You Can Send Dump Output	4-89
Stand-Alone Dump Program Generation	4-91
Using the Stand-Alone Dump Facility	4-92
IPL and Dump Formats	4-96
Error Handling	4-99
Section 5. Operator Spooling Functions	5-1
Spooling	5-1
Spooling Considerations	5-1
Spooling Functions	5-2
Spool Buffers	5-5
Spool Files	5-5
Output Classes	5-6
Output Forms	5-7
Spooling Commands	5-8
Spooled Card Input	5-9
Spool Files for Virtual Machines Running Batch Jobs	5-12
Using the CMS Batch Facility	5-12
Starting the Batch Virtual Machine	5-13
Controlling the Batch Facility	5-14
Purging, Reordering, and Restarting Batch Jobs	5-15
Stopping the Batch Virtual Machine	5-16
Appendix A. Summary of CP Commands	A-1
Index	X-1

Figures

2-1.	DOS 160K Virtual=Real Storage Assignments	2-30
2-2.	Command Control of VM/370 Hardware Assist	2-34
3-1.	CP Privilege Class Descriptions	3-2
3-2.	CP Trace Event Type Codes	3-28
4-1.	Annotated Sample of Output from the TYPE and PRINT Functions of the DDR Program	4-18
4-2.	VM-Supplied Character Arrangement Tables for the 3800 Model 1 and Model 3 Printers	4-20
4-3.	VM-Supplied Character Arrangement Tables for the 3800 Model 3 Only	4-20
4-4.	TEXT Deck Names Output by the GENIMAGE Command	4-21
4-5.	Format of 3330 Cylinders for Use by CP	4-26
4-6.	3330, 3340, or 3350 Cylinder 0 Format	4-27
4-7.	Using the Format Program Label Function	4-36
4-8.	Using the Format Program Allocate Function	4-36
4-9.	Using the Format Program Allocate Overlap Function	4-37
4-10.	Using the Format Program Label Function for FBA Devices	4-37
4-11.	Using the Format Program Allocate Function for FBA Devices	4-38
4-12.	Using the Format Program Allocate Overlap Function for FBA Devices	4-39
4-13.	Valid Options and Their Output	4-45
4-14.	Example of a Programmable Operator Facility Configuration.	4-58
4-15.	A single system with NCCF	4-59
4-16.	Example of Communication in the Local Environment	4-63
4-17.	Example of Communication in the Distributed Environment	4-64
4-18.	Example of Communication in the Mixed Environment	4-66
4-19.	Stand-Along Dump Facility Tape Format	4-97
4-20.	Stand-Alone Dump Facility DASD Format	4-98
5-1.	CP Spooling Commands	5-8

Section 1. Introduction to Operational Control of the VM/SP System

System operators, in almost all large System/370 installations, serve in an apprentice capacity until they have acquired the skill and knowledge to maintain the installation's operating system, be it DOS, OS, or VM/SP.

In an apprentice capacity, an operator acquires a basic proficiency in mounting, making ready, and loading tape, DASD, line control, and other hardware devices. Novice operators also learn the address designations of all hardware devices attached to the system. They become aware of those system resources that can be switched or patched via alternate channel or telecommunication path to expedite system operations. The operator becomes aware of system console terminal operation as well as the function of most of the indicators, switches, buttons, and alarms that are part of the processor control. All of this knowledge is basic and must be acquired before considering the operational control of any system control program (SCP).

The second phase of an operator's training is to be knowledgeable about the SCP that is controlling the data processing operations; he must be aware of its capability and its operating philosophy. The operator must also be aware of the priorities and the demands placed upon the system.

Much of this basic knowledge of VM/SP is described in the *VM/SP Introduction*. In that document there is discussion on virtual machines; how they are built, identified, and used. Each facet of the virtual machine is described: the virtual processor, virtual storage, virtual I/O devices, and the virtual system console. Concurrent virtual machine usage and spooling operations are also described.

With an understanding of VM/SP concepts, the power and versatility of VM/SP is soon realized. The system operator is not just controlling one process and its related storage and I/O with control and application programs; the operator is exercising control over the resources of multiple systems (virtual systems). Each virtual system with its own system operation, in turn, is being controlled by a user via a terminal console. When additional facilities and resources or priorities are needed by the virtual system, the virtual system operator must request these services from the VM/SP system operator. The system operator then delegates additional system resources to the virtual machine, if feasible.

Some of the operations that are exclusive functions performed by the VM/SP system operator are as follows:

- Reorder, purge, or copy any closed spool files

Operational Control

- Issue warning and high priority messages
- Attach and dedicate devices to specified virtual machines
- Automatically logon virtual machines
- Force users off the system when warranted
- Change any virtual machine's dispatch priority and operating characteristics.

For details on how many of these and other functions are accomplished, see "Section 3. CP Commands" on page 3-1.

Depending on the system installation, the system operator may also be the operator of a large DOS/VSE, OS/MVS, or OS/VS1 batch processing system that is run in the virtual machine environment. Another duty of the system operator is to control the data traffic from remote work stations. Information on VM/SP control of such data transfer is detailed in the *VM/SP Program Reference and Operations Manual*.

Regardless of the real system resources and the tasks and procedures used, systems are subject to hardware and program malfunction. The Interactive Problem Control System of VM/370 provides a method of recording and maintaining a history of VM/370 and virtual machine program problems. The *VM/370 Interactive Problem Control System (IPCS) User's Guide* provides the details on how to use this system.

Problems that are a result of a hardware fault are usually detected by the hardware itself, and trigger the recording of register contents and sense values (related to the occurrence of the error) in VM/SP's error recording area.

Regardless of whether the malfunction origin is hardware or software, the system operator (in most cases) is notified of the seriousness of the situations by means of console indicators/alarms or by console diagnostic messages. These all serve to inform the operator that:

- The system operation can continue as before.
- The system operation can continue with reduced power/resource.
- System restart and recovery is commencing.
- System operation is terminated.

The type of recovery that is attempted is described in "System/370 Recovery Management Support."

System/370 Recovery Management Support

IBM System/370 attempts correction of most machine errors without program assistance. CP is notified, via an interruption, of both intermittent and permanent machine errors to allow error recording and recovery procedures to start.

The following recovery features are implemented in the IBM System/370 hardware:

- Retry of the failing processor operations

- Validity checking on processor and control storage to correct all single-bit errors
- I/O operation retry facilities including an extended channel status word (ECSW), which provides channel retry data to channel and control unit retry procedures
- Expanded machine check interruption facilities to improve error recording and recovery procedures.

Machine Check Handler (MCH)

You can set the recording mode to record errors corrected by processor retry (logically termed as processor retry) and Error Correction Code (ECC) with the SET MODE command. In attached processor or multiprocessor applications, recording mode can be set for either or both processors. For processor retry, the default setting is record mode.

Note: The SET MODE MAIN command is invalid for 3031, 3032, 3033, and 308X processors.

When processor retry or ECC succeed in correcting errors, and the processor is in record mode, the machine check handler records the error. When processor retry or ECC fail, the machine check handler:

- Attempts to isolate the failure to one page frame and makes that page frame invalid or unavailable for paging
- Attempts to isolate the failure to one virtual machine and logs off or resets that virtual machine
- Attempts to isolate the failure to portions of the system and to continue system operation in degraded mode
- Abnormally terminates the system when recovery is not possible; or, if VM/SP is operating in attached processor mode and the malfunction is isolated to the attached processor and to a particular virtual machine, then, system operation continues in uniprocessor mode.

Note: If VM/SP is operating in multiprocessor mode and the following conditions exist:

- The malfunction can be isolated to one processor and one virtual machine.
- No system-owned device has its only online path on the failing processor.

The system can continue operation in uniprocessor mode.

Operational Control

MCH records an error whenever any of the following conditions occur:

- Processor retry occurred¹
- ECC corrected data¹
- Hardware reported a buffer or DLAT (Data Look Aside Table) error
- Multiple-bit storage failure
- External damage
- Storage protection feature damage
- Timer error
- System damage
- Instruction processor damage.

Channel Check Handler (CCH)

Whenever a channel control check, channel data check, or interface control check occurs, the channel check handler (CCH) constructs an error record and records the results in an IOERBLOK. The error recovery procedures use this IOERBLOK to retry the error. Recovery is not attempted for channel errors associated with virtual machine I/O events.

Missing Interrupt Handler

The missing interrupt handler monitors system I/O activity on specific device classes for interruptions that do not occur within a specified period of time. When a missing interruption is detected, the control program attempts corrective action, then notifies the system operator of the condition (either cleared or pending), and writes a record to the error recording area. The operator is notified so that he can take manual action if the control program's corrective action was unsuccessful.

I/O Error Recording and SVC 76

VM/SP maintains an error recording area that captures I/O, CCH, and MCH error records. Device and control unit detected unit checks during VM/SP spooling, paging, and virtual machine I/O errors generate the I/O records.

VM/SP and the virtual machine's SYS1.LOGREC data set contain recorded I/O errors; this double recording occurs when the virtual machine's operating system does not invoke SVC 76.

If the virtual machine operating system invokes SVC 76 and passes the correct parameters to VM/SP, VM/SP records the error in its own error recording area. VM/SP then passes control back to the virtual machine operating system, thus bypassing virtual machine error recording facilities.

¹ VM/SP records these errors only under specific conditions. The conditions for recording these errors are detailed in the *VM/SP OLTSEP and Error Recording Guide*.

VM/SP Recovery Features

The VM/SP recovery features are described more fully in the *VM/SP OLTSEP and Error Recording Guide*.

Recording Facilities

The Environmental Recording Editing and Printing (EREP) program is executed when the CMS CPEREP command is invoked. The output of the CPEREP command consists of printed reports whose content depends upon the specified (or defaulted) CPEREP operands and upon the input system error records. The reports generated by CPEREP have the same format as those generated on an MVS system. The input system error records may be from the VM/SP error recording area or from a history tape. The history tape may have been produced earlier by CPEREP from the VM/SP error recording area data or by an OS/VS system from SYS1.LOGREC data. Unlabeled tapes produced on OS/VS systems by EREP and on VM/SP systems by CPEREP are compatible and can be transported between systems. Data from both systems can also be accumulated on the same tape. For more details on CPEREP, refer to the following publications: *VM/SP OLTSEP and Error Recording Guide* and *Environmental Recording Editing and Printing (EREP) Program*.

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 recording area can be collected on an accumulation (ACC=YES) tape for further processing by the VS System Data Analyzer Program (SDA). Because MSS logged-out data is voluminous and the interrelationships of MSS components are complex, it is imperative that this service program be used to effectively diagnose and isolate mass storage problems.

VM/SP Repair Facilities

The Online Test Standalone Executive Program (OLTSEP) and associated Online Tests (OLT) execute in a virtual machine that can run concurrently with normal system operations. These programs provide online diagnosis of I/O errors for most devices that connect to the System/370.

The service representative (with a CP command privilege class of F) can execute online tests from a terminal as a user of the system; CP console functions, including the ability to display or alter 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

VM/SP tries to reload CP when a system failure causes:

1. An abnormal termination that does not result in a disabled WAIT state
2. The Dump to be directed to DASD.

Often, the operator does not need to do anything. The system attempts to execute a warm start, thus allowing user's terminals to be reconnected (for logon reinitialization by users) and completed spool files as well as open console spool files to be maintained. In the event of a warm start, device reconfiguration (such as varying a device off-line) that was performed by the real computing system operator is remembered by CP for system spooling devices only. Storage reconfiguration data acquired during the process of recovering from real storage errors is lost. After a VM/SP system failure, each user must re-access VM/SP (LOGON), and each virtual machine must be reloaded (IPLed).

If the operator was logged off, running disconnected, or logged on to a machine other than the primary system console, the operator will be restored as disconnected with the console spooled; CP issues the message:

```
"DMKOPE967I DISCONNECT userid - SYSTEM RESTARTED (mmmnnn) AND  
SYSTEM CONSOLE NOT VM OPERATOR CONSOLE:"
```

Otherwise, the operator will be logged on to the system console.

Termination of a virtual machine, whether caused by a real computing system malfunction or a virtual machine program error, normally does not affect the execution of other virtual machines unless the error involves shared segments.

With virtual storage preservation, the system programmer can tell VM/SP to automatically save the contents of up to a 16M byte virtual machine (main storage and registers) if:

- VM/SP terminates the virtual machine.
- VM/SP itself terminates.

The system programmer must specify at system generation time which virtual machines are going to be saved. The contents of the virtual machine or virtual machines are saved on DASD space that the VM/SP system programmer has previously allocated; users can then restore the contents via the IPL command. Normal recovery procedures for the virtual machines can then be initiated by their respective owners.

Virtual storage preservation is designed to make the automatically-saved virtual machine available only to one or both of two previously specified userids. This is for the privacy and security of the virtual machine. The saved virtual machine can be loaded into either a V=R or normal non-V=R area. A system generation macro instruction (NAMESYS) enables the system programmer to assign priorities

which indicate the order in which multiple virtual machines should be saved. To create an environment for saved systems requires both the use of the NAMESYS macro instruction and enabling the VMSAVE function via the CP SET command or the directory OPTION statement.

Note: VM/SP restart operations on systems employing virtual storage preservation can consume more time than is usual because the system must sequentially page out the contents of the saved virtual storage to the previously allocated DASD space. If an I/O malfunction prevents the saving of a VMSAVE system, the system operator is notified.

In VM/SP attached processor operations, the priorities and restart activities are similar to a uniprocessor mode of operation. However, if system damage assessment indicates an unrecoverable operation on the attached processor and the error can be isolated to a virtual machine, VM/SP will be continued in uniprocessor mode on the main processor. All virtual machines with an AFFINITY setting to the attached processor are automatically reset to AFFINITY OFF for subsequent virtual machine processing. (Note, the affected virtual machines are placed in console function mode.) If a virtual machine was running when the malfunction occurred, its operation will be terminated.

Note: See the VARY command in "Section 3. CP Commands" for instructions on how to vary the attached processor and multiprocessor back online.

If a similar malfunction occurred on the main processor while VM/SP was operating in attached processor mode, VM/SP terminates. VM/SP cannot switch to uniprocessor operation on the attached processor unit because the attached processor unit has no hardware interface for input/output capabilities.

Note: In certain 303x attached processor environments the channel set switching facility is present. In the case of a malfunction on the main processor in these environments, VM/SP can continue system operation in uniprocessor mode by switching the channel set from the failing processor to the remaining processor.

In VM/SP multiprocessor operations, if an unrecoverable error occurs on one processor and the error can be isolated to one virtual machine, it may be possible for VM/SP to continue operation in uniprocessor mode on the nonfailing processor. System operation can continue if all system-owned devices have online paths from the nonfailing processor. Virtual machines with affinity set to the failing processor have their affinity set off and are placed in console function mode.

A MESSAGE or WARNING appearing on the screen during restart, puts the screen in MORE status. Normally, this would put the screen in HOLDING status. However, if the screen is in MORE status during restart operations, pressing the ENTER key will put the screen in HOLDING status.

Operational Control

Section 2. Starting VM/SP Operation

Using CP and CMS

CP and CMS commands are never more than eight characters long and can be truncated to the minimum size indicated in the command format description. The operands, if any, follow the command on the same input data line. Most CP and CMS commands may not extend beyond one line except on the 3270. Generally, the operands are positional, but some commands have keywords to assist in the translation of the command line. One or more blanks must separate the command from any operands (except for some EDIT subcommands).

CMS indicates successful command processing with the CMS READY message. Error messages occur for erroneous operands. CP does not recognize CMS commands. However, a CMS user can issue CP commands without leaving the CMS environment. Although not required, CP commands entered while in the CMS environment should be prefixed with "CP" or "#CP" to decrease VM/SP table and disk search time. CP then handles the command and passes control back to CMS. If the CMS user wants to enter CP mode, he can do so by keying in CP or #CP with no operands (that is, no CP commands) or with an attention interruption. For details, refer to "Attention Handling" in the *VM/SP Terminal Reference*.

If a user requires communication with CP before responding to specific error messages issued by some virtual machine operating systems, he should enter CP mode by keying in #CP with one or more CP command lines separated by a logical line end character, or by signalling attention. Then perform the necessary console functions. When returning to the virtual machine environment, the virtual machine read will have been canceled by a unit exception, and the virtual operating system normally responds by reissuing the read. The user can then enter the required virtual machine response.

Console Procedures

You, as the VM/SP system operator, are like other system users, but with some important differences:

- You are allowed to perform certain console functions that the normal user is not.

Starting Operation

- Your terminal normally stays in CP mode and is usually ready to receive messages from the VM/SP system and other users.

Unless you are using a 3270 terminal, before you can type a command, you must press the REQUEST key (or its equivalent) on the designated VM/SP system console. VM/SP responds with the time, and sets up to read data from the console keyboard. You can then enter CP commands into the system.

You can enter all CP commands in lowercase or uppercase, or a mixture of both. All system responses are prefixed with the clock time at which you entered the command from the primary console. Full descriptions of CP commands with a class of A through F are in Section 3, "CP Commands" on page 3-1.

Enter commands into the VM/SP system by pressing the end-of-line function key on the console to terminate the command line. To cancel a miskeyed or inappropriate VM/SP command that you partially or totally entered on the console, enter the designated logical line delete character (normally, the ϵ symbol) or press the CANCEL key (or its equivalent).

Note: The CANCEL key on the 3215 cannot cancel lines of input until you finish defining the VM/SP system during system generation. Also, the PA2 key or the CANCEL key on the 3270 and 3066 cannot cancel input lines.

For details on supported VM/SP typewriter-like terminal consoles, display terminal consoles and the associated keys that manage the display screen, signal attention, and process line records, consult the *VM/SP Terminal Reference*.

Operating Procedures (Operator Tasks)

Labeling Disks

CP and CMS, with the exception of the special CP service programs and certain frequently used CP functions, get all of their system functions from disk. Each of the disks may or may not be labeled. If the disks are for CMS file residence, or CP paging and spooling (temporary use), you must label each pack (count-key-data pack) with a six-character label in real cylinder 0 track 0 record 3 by using the CP Format/Allocate program. For FB-512 devices, this label is placed in block 1.

The VM/SP system is device-independent; each of the volumes that have CP labels can reside on any available and defined 2314, 2319, 3330, 3333, 3340, 3375, 3380, 3344, 3350, 2305 or FB-512 direct access storage device. This definition occurs at system generation time. The CP system residence (SYSRES) volume must be mounted on an available IPL device; then use the normal IPL procedure for your processor. If you are using CP-owned volumes, they should be mounted and ready at this time. All other resident volumes should be mounted and ready when the system is started, although they can be added later (via the CP ATTACH command) to the system, while it is in operation.

Assignment of Special Performance Options to Logged-On Users: FAVORED, AFFINITY, SASSIST, CPASSIST, RESERVE, PRIORITY, LOCK, and UNLOCK commands can be assigned only by an operator with privilege class A.

Handling of Spooled Input and Output: Card decks for users must be fed into the real card reader, printed and punched output properly distributed, and the unit record equipment and spool data files controlled. The control functions for the system's unit record equipment and spool data files can be performed only by a class D operator. For 3800 virtual printers, the operator must mount the designated paper stock if directed to do so.

Attaching and Detaching of User and System Volumes: Devices used by virtual machines in dedicated mode must be attached and detached as appropriate. Control over the real System/370 computing system's I/O can be performed by an operator with privilege class B.

Operators with these privilege classes must be logged on to perform these functions. An example of a directory entry that allows an operator to perform all of these functions is:

```
USER OPERATOR OPASS 512 1M ABDG
```

The system and spooling operators do not require virtual devices or options in their VM/SP directory entries unless CMS is used. Multiple virtual machines for operators may be set up, each with all or some of the associated privilege classes. The primary system operator must have class A assigned in the directory to properly initiate VM/SP operation.

System Definition

If you intend to define or attach more than 64 virtual devices for a single virtual machine, be aware that any single request for free storage in excess of 512 doublewords (a full page) may cause the VM/SP system to issue an appropriate error message if the extra storage is not available on a contiguous page. Therefore, two contiguous page frames of free storage must be available in order to log on to a virtual machine with more than 64 virtual devices (three contiguous page frames for a virtual machine with more than 128 virtual devices, etc.). Contiguous page frames of free storage are sure to be available only immediately after IPL, before other virtual machines have logged on. Therefore, as a system operator you can inhibit the logon of many virtual machines until the virtual machine or machines with many I/O devices have accomplished logon and are operating.

| System Initialization

| Before you invoke any type of start:

- | • The following devices must be powered up and ready:
 - | – The processor
 - | – Processor storage

Starting Operation

- All necessary devices (and disk volumes) for a minimum VM/SP system with system residence, paging, and spooling requirements.
- The system cannot encounter any unrecoverable hardware errors in the initialization and startup phase of bringing the VM/SP system online.
- The generated VM/SP SCP and the System/370 hardware and its attached features must have matching compatibility. If they are not compatible you will get an error message in addition to VM/SP performance degradation.

To simplify system initialization, be sure that:

1. All VM/SP resident volumes (specified in the SYSOWN list for paging and spooling) are mounted and ready at IPL time.

If the volumes specified in SYSOWN are not mounted when you IPL VM/SP, they are not available to you. If you need the volumes, use the VARY ONLINE command and the ATTACH command to mount and attach them.

2. Volumes that have user minidisks (such as the CMS system residence volume) are mounted and ready at IPL time. You can, however, vary online and attach volumes not required for startup after the VM/SP program is up and running.

Overview: The Steps Involved with System Initialization

1. Initialization Begins: To start the operation of the VM/SP system, load the CP system residence volume.

Note: When you turn system power on, you may need to run a CS (control storage) patch routine. This is a System/370 operating procedure and is not discussed further in this book.

CP calculates the real machine's storage size. If a real machine has more storage than what you specified on SYSCOR, the system does not use the extra storage. For example, if a real machine's storage size is 1 megabyte and you set SYSCOR to 512K, the system uses 512K of real storage. If less storage is available in the real machine than what you specified on SYSCOR, the system uses the lesser figure.

The system checks for a valid microcode level. If the microcode level is invalid, the system disables the microcode assists and stores an error message for display later. Also, CP searches for the devices defined during system generation and checks to make sure the volume labels on each DASD are correct. CP varies the devices online.

Note: If the configuration differs from that specified during system generation (SYSOWN volumes not mounted or storage size not equal to SYSCOR), the system stores the message for display later and continues to operate. Label errors also cause the system to store an error message for later display, and the system startup may abnormally end.

CP starts the segment, page, core, and swap tables. It also checks to see if VM assist and the 370E facility are available.

2. Locating the Operator's Console: CP locates the operator's console. During VM/SP system generation, your installation will assign a console or terminal as the primary operator's system console. Your installation can specify more than one alternate console in the RIOGEN macro instruction during system generation. If the specified primary system console is not operational, CP will use the first designated alternate console. If the first alternate console is not operational and your installation specified more than one alternate console, CP will use the next alternate console.

The primary and alternate consoles must be one of the following devices:

- A real primary or alternate system console.
- A locally attached 3270.
- In addition, the alternate console can be any VM/SP supported remote terminal on a leased line connected to a 270x transmission control unit.

The system does not support, as primary or alternate VM/SP consoles, terminals on switched lines or leased line terminals connected to 3704/3705/3725s.

Next, the following message tells you what release and service level of VM/SP you are using, and when it was created:

```
VM/SP RELEASE x, SERVICE LEVEL xxxx, CREATED ON mm/dd/yy AT  
hh:mm:ss
```

3. Time-of-Day Clock: CP then checks the status of the time of day (TOD) clock. If the clock is not set, CP requests that you set it. Refer to "Time-of-Day Clock" on page 2-7 and "Setting the Clock (Uniprocessor Application)" on page 2-8. If the clock is set, CP lets you change the clock. The following lines are displayed:

```
IT IS NOW 08:28:00 EST THURSDAY 01/26/86  
CHANGE TOD CLOCK (YES|NO):
```

If you enter "YES", CP responds with prompts like those explained in "Setting the Clock (Uniprocessor Application)" on page 2-8. If you press the ENTER key, CP assumes the response is "NO".

For AP/MP, if the other processor is available, CP begins prefixing and synchronizes the TOD clocks.

Note: If the SYSCOR macro had specified AP=YES or MP=YES and the second processor is unavailable for startup, the system operates in uniprocessor mode instead of attached processor or multiprocessor mode.

CP initializes the user directory.

CP verifies that the interval timer is running. If it is not, CP requests that you start it.

At this point, the system displays any delayed messages. Also, if any CP owned volumes are malfunctioning or not mounted, the system displays one or more mes-

Starting Operation

sages. This lets you select a shutdown start in the next step of the IPL process if you need to.

4. Selecting the Type of Start: The system then prompts you for the type of system start you want:

```
START ((WARM|CKPT|FORCE|COLD) (DRAIN))|(SHUTDOWN) :
```

You can respond in the following ways:

COLD or COLD DRAIN

WARM or WARM DRAIN

CKPT or CKPT DRAIN

FORCE or FORCE DRAIN

DRAIN

SHUTDOWN

WARM is the default. Specify DRAIN if you do not want the system to automatically start the real unit record devices during initialization. The types of system starts are explained in:

- “Cold Start” on page 2-11
- “Warm Start” on page 2-9
- “Checkpoint Start” on page 2-10
- “Force Start” on page 2-10
- “Shutdown Start” on page 2-11.

CP then allocates DASD dump space. If CP cannot allocate the dump space, it issues the following message:

```
DMKIDU953I UNABLE TO ALLOCATE SYSTEM AUTO DUMP
```

5. Logging on the System Operator: Next, the system automatically logs on the system operator.

The system operator’s userid, password, and command privilege class or classes are usually stored on the CP system residence volume during system installation. The primary system operator must have class A command privileges. CP compares the userid supplied by the SYSOPR macro (used in system generation) with the user directory. If the userids do not match, CP stops the automatic operator logon process and gives system operational control to the first privilege class A user who logs onto the system.

If the primary or alternate devices do not exist or are not ready, automatic logon stops, and the VM/SP system enters the disabled wait state. If the primary or alternate consoles are operational, but some processing error occurs, you must intervene to bring up the VM/SP system. CP considers the first user to log on VM/SP with privilege class A as the primary system operator. For more information see “Manual Logon” on page 2-12. The system does not accept any users until the primary system operator logs on.

Starting Operation

The following message is then displayed:

```
DMKCPJ957I STORAGE SIZE = xxxxx K,      NUCLEUS SIZE = xxx K,  
          DYNAMIC PAGING SIZE = xxxxx K, TRACE TABLE SIZE = xxx K,  
          FREE STORAGE SIZE = xxxxx K,   VIRTUAL=REAL SIZE = xxxxx K
```

If MONITOR is enabled, the system starts MONITOR. Also, the system logs on the AUTOLOG1 virtual machine.

Finally, CP displays the following message:

```
DMKCP1966I INITIALIZATION COMPLETE
```

At this point, the system is ready for normal use. You can now:

- Write a message of the day for the system to issue when users log on.
- Enable communications lines so users can log on.

Time-of-Day Clock

The Time-of-Day (TOD) clock provides an accurate measure of time, independent of system events or activities. It makes accurate measurements available for programming applications. When system power is turned off, the clock value is lost on some System/370 models. Once CP uses the set clock instructions and you use the TOD ENABLE SET switch to make the time-of-day (TOD) clock operational, the system increments the clock at a constant rate. The following do not affect the timing operation:

- Any normal activity or event in the system
- Wait state
- Stopped state
- Instruction-step mode
- Single-cycle mode
- Test mode¹
- System reset
- Initial program load procedure.

If TOD clock hardware errors occur, the System/370 hardware posts a machine check and CP enters a disabled WAIT state.

Virtual machines may use the System/370 STORE CLOCK instruction to find the current clock value. This instruction causes VM/SP to store the current clock value in the storage location specified in the instruction. Thus, virtual machine users can use the value of the TOD clock for any purpose. However, virtual machine users cannot set or change the TOD clock.

VM/SP uses the TOD clock to establish:

- Certain accounting records for the virtual machines

¹ The current clock value is lost when the time-of-day microdiagnostic tests are being run, or when system power is turned off.

Starting Operation

- Queuing and prioritizing tasks for CP and virtual machine operations.

The privileged System/370 instruction, SET CLOCK, sets the clock to a specific value. If certain operating or program conditions exist, CP executes SET CLOCK when the system is initialized. SET CLOCK replaces the current clock value with the value you specified. The SET CLOCK instruction changes the clock value only when the TOD ENABLE SET switch is enabled.

If your installation has a second processor, you may need to synchronize the TOD clocks in the initialization procedure. The system may request that you ready the TOD ENABLE SET key twice: once to set the clock on the main processor and once to SET and SYNC the clock on the second processor. If the clock value has previously been established for the main processor, the system will only issue one prompt message.

Note that the TOD ENABLE SET key is ORed to the other processors' TOD ENABLE SET key. This lets you set the clock from either processor. If the low-order 32 bits of the TOD clock are not synchronized, a TOD Sync check will be received. Again, the system will request that you synchronize the TOD clocks by pressing the TOD ENABLE SET key.

4361 processors are offered with an Auto Start feature. Using a battery operated clock, this feature maintains the time while the power is off. For these processors, if you shut down the system using the SHUTDOWN command with the POWEROFF parameter, during the next IPL, you will not be prompted to set the time-of-day clock.

Details on the parts of VM/SP initialization that are unique to the attached processor or multiprocessor applications are explained later in this chapter.

Setting the Clock (Uniprocessor Application)

If the clock is not set, CP issues the following message to prompt you to enter the date:

```
SET DATE (MM/DD/YY) :
```

Enter the date.

Note: If the two digit number for the year is less than 50, the system assumes the century is 2000. If the number is equal to or greater than 50, the system assumes the century is 1900. Do not use the numbers 41 through 49, because they do not represent anything.

CP then requests the time:

```
SET TIME (HH:MM:SS OR HH.MM.SS) :
```

Enter the time. Note that this is a 24 hour clock.

Notes:

1. Enter data immediately following the colon (:). If you are using a 3270 or a VM/SP supported processor display console, enter the data in the user input area. The data then appears in the output display area on the line following the SET TIME request.
2. Because CP requests that you press the TOD ENABLE SET key to set the clock, enter a time value that is 30 to 60 seconds ahead of the current time.

For example, if the date is 01/24/86, and the time will soon be 08:28 (EST), the exchange on a printing terminal looks like this:

```
VM/SP RELEASE 4,SERVICE LEVEL 0000,CREATED ON 01/01/86 AT 11:15:00
SET DATE (MM/DD/YY) :01/24/86
SET TIME (HH:MM:SS OR HH.MM.SS) :08.28.00
PRESS "TOD ENABLE SET" KEY AT DESIGNATED INSTANT
```

When you press the TOD ENABLE SET key, CP responds with:

```
IT IS NOW 08:28:00 EST FRIDAY 01/24/86
CHANGE TOD CLOCK (YES/NO):
```

Note: The previous example run on a display terminal would show your responses on separate lines.

You can now change the clock's value. If you enter "yes," another series of prompting messages let you enter a new date and time. If you enter "no," or press the END function key on the console, clock initialization ends.

Programming Note: To VM/SP users, the epoch clock value of all zeros is the beginning of the day January 1, 1900. To OS and MVS users, the epoch clock value of all zeros is the beginning of the day January 1, 1980. Thus, OS virtual machine users will find that stored clock values differ by 80 years. You can, however, use 1900 in OS as the base epoch clock value.

Warm Start

Warm start means that the previous VM/SP session ended with an orderly shutdown procedure. Use a warm start procedure to recover accounting data and access previously closed spooled output files. CP tells you when a warm start cannot be done and requests an alternate recovery method.

Begin a warm start:

1. If you are not alerted by console messages or visual signs that a checkpoint or cold start is required.
2. After VM/SP has abnormally stopped and a system dump operation has completed.
3. After an orderly shutdown has been performed.

Starting Operation

Warm start restores previous spool controls and files. Accounting records remain valid and are to be kept as a base for continuing operations.

Key in `WARM` or simply press the `END`, `ENTER`, or `RETURN` key (depending on the console device) when the following message appears at the console:

```
START ((WARM|CKPT|FORCE|COLD)(DRAIN))|(SHUTDOWN) :
```

(You can also specify `WARM DRAIN` to warm start the system and drain all unit record devices.)

If you try a warm start, but the system has not saved any warm start data, the system will enter a wait state `PSW` (code `009`). In this case, try a checkpoint start.

In many cases, `VM/SP` recovers from system failures and restarts itself. When this occurs, you will see the following message at the console:

```
VM/SP SYSTEM RESTART DUE TO SYSTEM FAILURE
```

When you enter the `SHUTDOWN REIPL` command, you will see the following message at the console:

```
VM/SP SYSTEM RESTART DUE TO SHUTDOWN REIPL
```

Checkpoint Start

If the system could not do a warm start because of I/O errors or invalid data in the warm start area, you can request a checkpoint start.

This option tries to start the system using the information that has been dynamically checkpointed during system operation and stored in the checkpoint area. Checkpoint start reconstructs the spool file chains, but the original order of the files is lost. Accounting and system message data, recovered under a warm start, is lost under a checkpoint start. Also, because the system must read each spool buffer to reconstruct the record allocation blocks, a checkpoint start takes longer than a warm start.

To begin a checkpoint start, respond with `CKPT` to the following message:

```
START ((WARM|CKPT|FORCE|COLD)(DRAIN))|(SHUTDOWN) :
```

Force Start

If the system cannot do a checkpoint start because of I/O errors or invalid data in the checkpoint area, you can request a force start. This option operates like the checkpoint start, except that any spool file that has unreadable or invalid data is dropped from the system.

To begin a force start, respond with `FORCE` to the following message:

```
START ((WARM|CKPT|FORCE|COLD)(DRAIN))|(SHUTDOWN) :
```

Cold Start

You may want to use cold start procedures if you have tried a warm start, a force start, and a checkpoint start, and all have failed. This may happen when:

- You migrate to a new release of VM/SP.
- All closed spool files and accounting records are lost or unreadable because of a hardware error that prevented valid system recovery and continuation.

VM/SP issues a message telling you to do cold start procedures when it recognizes that unrecoverable errors have occurred. You may have failures in system component power, building power, hardware logic, or the CP program. Also, some (rare) kinds of user programs can be classified as system operating procedure errors. All these types of failures can require that you start the system with a cold start.

VM/SP cold start procedures start after the condition, which caused the System/370 or the operator to abnormally stop VM/SP, has been corrected.

To start a cold start, respond with COLD to the following message:

```
START ((WARM|CKPT|FORCE|COLD)(DRAIN))|(SHUTDOWN) :
```

After the cold start procedure is complete, enter into the LOGMSG the time of the unplanned shutdown and that a cold start was done. This tells the user that his spooled files were lost, and that he may need to reconstruct some of his other files.

Shutdown Start

A shutdown start stops the initialization process. You may want to do a shutdown start if you receive a message that one or more vital CP owned volumes are not mounted or malfunctioning. For a shutdown start, the system does not process warm start or checkpoint data. Following a shutdown start, previously saved warm start and checkpoint data is still valid.

To begin a shutdown start, respond with SHUTDOWN to the following message:

```
START ((WARM|CKPT|FORCE|COLD)(DRAIN))|(SHUTDOWN) :
```

If you cannot key in the SHUTDOWN command, press the LOAD button on the System/370 console without clearing storage. After you see the shutdown message, press the LOAD button a second time to reload the system with VM/SP.

Automatic IPL

This feature is supported for 4361 processors only. 4361 processors are available with an Auto Start feature. Using a battery operated clock, this feature maintains the time while the processor is powered off and sets the system time-of-day clock after power on.

The 4361 processor with the Auto Start feature can be set to automatically power on at a specified time, or it can be powered on by telephone from a remote location.

Starting Operation

Once the processor has been powered on, if you previously shut down the system using the SHUTDOWN command with the POWEROFF parameter, you will not be prompted to set the time-of-day clock. In fact, you do not need to be present for the IPL to occur.

Automatic Logon

Automatic logon is normally part of an IPL. When the system operator is logged on, the following message shows the date and the time of the last LOGMSG setting:

```
hh:mm:ss LOGMSG- hh:mm:ss EST FRIDAY mm/dd/yy
```

The following message is displayed next only if queued, closed spool files exist.

```
hh:mm:ss FILES: xxx RDR xxx PRT xxx PUN
```

Notes:

1. A MESSAGE or WARNING on the screen during system IPL puts the screen in MORE status. Normally, this would put the screen in HOLDING status. If the screen is in MORE status during system IPL operations, you may press ENTER to put the screen in HOLDING status.
2. If the automatic logon fails, the user (privilege class A) will have to explicitly log onto the system. If the system console is either a 3210 or 3215 (or a 3138, 3148, or 3158 console in printer keyboard mode), the user must protect the security of his password because there are no print-masking or print-inhibiting features on these devices.

Manual Logon

If the automatic logon of the primary system operator fails because the VM/SP directory defines no system operator, the following message occurs:

```
OPERATOR NOT LOGGED ON; EXPLICIT LOGON REQUIRED
```

Check your directory entry to confirm your userid and password. A logon prompt will not be displayed. Log on with your userid:

```
LOGON user id
```

In the above response, you enter the CP LOGON command, followed by a blank and your identifier name (userid), followed by pressing the end function key. CP checks this entry with the values in the VM/SP directory. If the userid is valid and you have class A command privileges, the following message on the console prompts you for the next response:

```
hh:mm:ss ENTER PASSWORD
```

You then key in your password, which is checked against the password stored in your VM/SP directory entry. The system is not accessible to other users until you log on and enable the necessary teleprocessing lines. Maintaining password secu-

urity on real printer-keyboard system consoles is your responsibility because password inhibiting or masking is not possible on these devices.

If the password is valid, the VM/SP system can be started.

Note: If the system operator as defined in SYSOP in DMKSYS logs off, the next class A user who logs on will become the system operator. That is, the system will send operator messages to this class A user, even if the userid originally specified in SYSOP logs back on. The class A user will continue to receive any operator messages that the system sends until he logs off.

When this class A user logs off, then there is no operator. A user sending a message to the operator will get the message, DMKMSG045E 'OPERATOR NOT LOGGED ON'. The next class A user to log on will become the system operator.

Unit Record Spool Files

After starting the VM/SP system, the system responds with the status of the system spool files:

```
hh:mm:ss FILES: 031 RDR, 039 PRT, 001 PUN
```

The response indicates that a total of 31 spooled reader files, 39 spooled print files (which also includes closed spooled console files), and 1 spooled punch file were saved at the last system shutdown.

If the unit record devices are drained as a result of the DRAIN option during IPL, then you should start the unit record devices at this time. For example, the CP command

```
START ALL
```

starts all real unit record equipment. The system then responds with the status of each unit record device.

| 37xx Resources

| At this point, if the following conditions are met:

- | • 3705 control units are part of the VM/SP system support of remote terminals
- | • The automatic load function for the 3705 was not generated into the VM/SP system
- | • The 370x control program was generated under VM/SP

| invoke the NETWORK LOAD command as follows:

```
| NETWORK LOAD raddr ncname
```

| Raddr is the base address of the 3705 and ncname is the desired copy of the 3705 Emulator Program. For a description of the NETWORK command, see "NETWORK" on page 3-102. If you are using 3725 communication control units, use the ACF/NCP-SSP loader.

Starting Operation

You can then enable any or all of the EP teleprocessing lines and 3705/3725 resources for VM/SP users by issuing the CP ENABLE command and NETWORK ENABLE command, respectively. For example, pressing the REQUEST function key and entering:

```
hh:mm:ss ENABLE 50
```

enables only line 50, while:

```
hh:mm:ss EN ALL
```

enables all 3270 and 270x lines. CP responds with:

```
hh:mm:ss COMMAND COMPLETE
```

when the requested command has been serviced.

Note: The same message occurs even if no lines connect to the machine. If a patch panel is in use, make sure that all desired lines connect to the system.

The VM/SP system is now operational and waits for users to log on. Similarly, if:

```
NETWORK ENABLE ALL
```

is invoked, the total resources of all VM/SP controlled 3705 control units are enabled if the system is equipped with the devices.

Mass Storage System Initialization

If an MSS port is attached to a virtual machine running MVS with MSS support and the MSS communicator program is running in the virtual machine, the control program can cause automatic 3330V volume mount and demount in response to any of the following:

- A virtual machine logs on with a minidisk defined on a system disk that is not mounted. The control program attempts to mount an MSS volume with the correct volume label.
- A virtual machine logs on having a dedicate directory statement specifying a valid for a volume that is not mounted. The control program again attempts to mount an MSS volume with the specified volume label.
- An operator ATTACH command is issued specifying a device address that is an MSS 3330V address and valid. The control program attempts to mount the valid on the device address as part of the ATTACH processing.
- The operator DEFINE command is used to change the feature (SYSVIRT or VIRTUAL) of a 3330V containing a mounted volume. The control program demounts the volume before completing the DEFINE process.

- A virtual machine issues a LINK command to link to a virtual device that is all or part of a 3330V volume. The control program attempts to mount the volume as part of the LINK process.

The VM/SP control program does not issue orders directly to the mass storage control (MSC). Rather, it passes requests to an MVS system with MSS support that is operating in a virtual machine with an MSC port dedicated to it. The MVS system then issues orders to the MSC and passes response information back to the control program.

In order for the installation to run with full MSS support, an MVS system must be IPLed in a virtual machine. An MSC port must be dedicated to this virtual machine. There are no special requirements for this IPL. The standard installation procedures for IPLing MVS in a virtual machine should be followed. After the system is IPLed, the VM/SP communicator program must be started. If the standard MSS installation procedures as documented in the *VM/SP Planning Guide and Reference* have been followed, the communicator program is started in the virtual machine through the MVS operator command:

```
START DMKMSS [.pn]
```

The [.pn] operand is used only in OS/VS1 systems to specify the partition in which the program is to be started.

The absence of error messages received from DMKMSS indicates that MSS support is initialized. The communicator virtual machine may now be disconnected if the installation desires.

There are two error messages that the DMKMSS program can produce. If either of these messages is received, then the VM/SP control program will not be able to communicate with the MSS. The first possible message is:

```
DMKMSS ENDING ERROR - MSS NOT INITIALIZED
```

Explanation: The MSS did not get initialized as it should during the MVS IPL process. Either the virtual machine does not have access to the MSC, or there was an MSS error.

Operator response: Correct the problem, use the MVS VARY command to initialize the MSS, then reissue the START DMKMSS command.

The second possible error message is:

```
DMKMSS ENDING ERROR - DCB FOR COMM. DEVICE NOT OPENED
```

Explanation: The DMKMSS program was not able to establish communication with the VM/SP control program because the MVS DCB control block could not be opened. The probable cause is the definition of the communicator device for the virtual machine.

Operator response: Contact the system programmer. After the problem has been corrected, reissue the START DMKMSS command.

Starting Operation

Multiprocessor Initialization

In order for the installation to run with a multiprocessor, it must do the following:

- Specify MP=YES in the SYSCOR macro during VM/SP system generation
- Have the second processor online during system IPL.

If the second processor was generated as part of the system but was not online, and the multiprocessing feature is installed, the following message is issued:

```
SECOND PROCESSOR NOT ONLINE
```

If the second processor is not required, processing can continue. However, if the second processor is required, the operator should ready the unit and reload the system.

Assuming MP=YES and the second processor was online the following message is issued:

```
PROCESSOR xx IPLED; PROCESSOR yy INITIALIZED
```

No operator action is required; the system will now utilize the second processor.

If the installation is running with a second or an attached processor, the system operator will be issued the following message if the clocks are to be synchronized.

```
CLOCK SYNCHRONIZATION-PRESS 'TOD ENABLE SET'
```

From this message on, see “Attached Processor Initialization” because the procedure is the same.

Attached Processor Initialization

In order for the installation to run with an attached processor, it must do the following:

- Specify AP=YES in the SYSCOR macro during VM/SP system generation
- Have the attached processor online during system IPL.

If the attached processor was generated as part of the system but was not online, the following message is issued if the multiprocessing feature is installed:

```
ATTACHED PROCESSOR NOT ONLINE
```

If the attached processor is not required, processing can continue. However, if the attached processor is required, the operator should ready the unit and reload the system.

The operator will then press the TOD ENABLE SET key on either processor to synchronize the TOD clocks. Note that once the clocks are synchronized, it will not be necessary to resynchronize the clocks on subsequent IPL operations unless:

- FE diagnostics are run.
- Power is turned off on either processor.
- The clocks are running but not synchronized.
- A clock's error is encountered during initialization.

If the clocks should run out of low-order synchronization, after the clocks are synchronized (during or after system initialization), you will receive the following message:

```
TOD CLOCK SYNC CHECK RECEIVED
CLOCK SYNCHRONIZATION-PRESS 'TOD ENABLE SET'
```

As before, press the TOD Enable SET Key to synchronize the clocks.

Interval Timer

VM/SP uses the interval timer to support time slicing; therefore, during initialization, the system verifies that the interval timer is running before virtual machine logon operations are performed. The system initialization routine will loop issuing the following message if the interval timer is found not running when tested:

```
TURN ON THE INTERVAL TIMER
TURN ON THE INTERVAL TIMER
:      :      :      :
:      :      :      :
:      :      :      :
```

The messages cease when the operator complies by enabling the interval timer and the system initialization process continues.

In attached processor application the text of the message indicates the processor owning the disabled interval timer. This message reads as follows:

```
TURN ON THE { MAIN } PROCESSOR INTERVAL TIMER
              { ATTACHED }
```

To continue, the interval timer must be enabled on the requested processor.

Console Definition

The terms “system console,” “primary system console,” and “alternate system console” apply to those console devices that exercise primary control of the processor.

The VM/SP primary system console can be a real system console, a real alternate system console, or a local 3270.

The VM/SP alternate system console can be a real system console, a real alternate system console, a local 3270, or any VM/SP supported remote terminal on a leased

Starting Operation

line connected to a 270x transmission control unit (TCU). The VM/SP alternate system console cannot be a remote terminal connected to a 37xx TCU.

Note: If the 37xx is loaded with the 270x Emulation Program prior to initializing VM/SP, any line defined by EP will appear to VM/SP as 270x TCU.

The following examples show the various configurations that can control VM/SP operations:

- **First Configuration:**

Real system console = VM/SP system console
Real alternate system console = VM/SP alternate system console

- **Second Configuration:**

Real system console = Batch virtual machine
Real alternate system console = VM/SP system console
Remote terminal = VM/SP alternate system console

- **Third Configuration:**

Real system console = VM/SP system console
Remote terminal = VM/SP alternate system console

- **Fourth Configuration:**

Real system console = Batch virtual machine
Two locally attached 3277 Display Stations = VM/SP system console and alternate system console

- **Fifth Configuration:**

Enabled remote terminal (not 3270) = VM/SP alternate system console
(No valid VM/SP primary system console is designated in SYSGEN)

Note: The real system consoles must be turned off to prevent VM/SP from using them.

- **Sixth Configuration:**

System console = VM/SP system console (No VM/SP alternate console designated in SYSGEN)

The first four configurations allow disabling of the VM/SP primary console and selection of an alternate VM/SP console; or, conversely, reselection of the VM/SP primary console and disabling of the alternate console.

Configurations five and six indicate that VM/SP system generation specified no console substitution. Therefore, if the designated VM/SP system console is inoperative at VM/SP startup, the system enters the wait state and cannot continue.

After the console becomes operative again, VM/SP must be reloaded into the real system.

Disabling the VM/SP Primary Console

If the VM/SP primary system console is inoperative, VM/SP automatically selects the first alternate console (specified in the RIOGEN macro instruction) as the console for primary system operation. If the first alternate console specified is not operational, an attempt will be made to start the next alternate console. If an operational console is found, it will be used as the VM/SP system operator's console.

If intermittent errors occur at the VM/SP console, and all alternate consoles are unavailable (or not specified in the RIOGEN macro instruction), issue the DISCONN command and log on again at some other terminal.

If the VM/SP system console becomes inoperative to the point where you cannot log off, press the System/370 external INTERRUPT button to disconnect you from the failing system console (or alternate console if that console is currently controlling the system.) This allows you to log on and reconnect to a terminal with a communications line that has already been enabled. This terminal can be another system console (VM/SP supported device) or a remote terminal attached to a 270x TCU. When the primary console is again operative, control can be regained from the alternate console by pressing the INTERRUPT button and reconnecting using the primary console.

When the device specified as the alternate console is a communication line, VM/SP enables the line. You can then establish a link with the processor through a terminal connected to the line. Once the line is established, VM/SP proceeds with the normal system initiation. The alternate console must not be a telecommunications line on a real IBM 37xx Communications Controller. If the alternate console is an IBM 2741 Communication Terminal, it must use the EBCDIC transmission code.

Note: VM/SP 37xx supports multiple alternate console selection only in 270x emulator mode.

Selecting a VM/SP Alternate Console during VM/SP Initialization

If the VM/SP system console is inoperative, a VM/SP alternate system console can start VM/SP. To do this, place the primary VM/SP system console in a not-ready status.

Load VM/SP in the normal way (with the System/370 console switches and the LOAD key). When the primary system console (assuming it was generated as the VM/SP system console) indicates "not ready", VM/SP rings the console alarm bell and enables the alternate system console (such as a 3210 Model 2, 1052 Model 7, or local 3270 if one is specified) or one 2701, 2702, or 2703 transmission control unit (TCU) line that has a leased line terminal attached, to allow the system operator to log on. The TCU line address must be defined in the REALIO module during VM/SP system generation. If the real system consoles are dedicated to virtual machine use only, the system can be generated with a fictitious VM/SP

Starting Operation

primary system console address and the same results occur as the load operation with the not ready console described above.

You can now establish a link with VM/SP by using the designated device. Once CP identifies the terminal in the normal procedure, the operator presses the BREAK, ATTN, RESET, or ENTER key, depending on the terminal type. The VM/SP system then performs automatic or manual logon procedures, as previously discussed.

Controlling the 370x Communications Controller

This section only applies to EP gens as defined, created and loaded with VM/SP. If you have a 3725 communication controller or a 3705 that has been loaded by ACF/NCP-SSP, refer to the manuals listed in the preface for control information.

Because the 3704 and 3705 Communications Controllers are programmable devices, in order for them to be controlled by VM/SP, they must be loaded with an appropriate control program before VM/SP uses them for system telecommunications. This loading is normally an automatic function of VM/SP system initialization, but it can be performed manually.

To load a 370x program into the 370x (either manually or automatically), the 370x program must previously have been saved on one of the VM/SP system volumes. Specifications in the VM/SP configuration module, DMKRIO, describe whether the load process is automatic or manual. For more information on saving or loading a 370x control program, see the *VM/SP Planning Guide and Reference* and the *VM/SP Installation Guide*.

If manual loading is necessary or desirable for a 370x control program, you should use the NETWORK LOAD command as follows:

```
NETWORK LOAD raddr ncname
```

where raddr specifies the physical device address of the 370x, and ncname is the name of a 370x control program image.

If the specified 370x Communications Controller does not require loading (that is, it already has an active control program loaded), the following message appears on your console:

```
CTLR raddr IPL NOT REQUIRED; ENTER 'YES' TO CONTINUE:
```

You should verify that you specified the correct 370x, and reply "YES" only if the load is to be attempted. (Note that execution of the NETWORK LOAD command causes the 370x to be reset, thus discontinuing any current teleprocessing activity.)

Other messages relating to the operation of the 370x can appear on your console. These messages are described in *VM/SP System Messages and Codes*. For more information about the facilities of the NETWORK command, refer to “Network Dump Operations” on page 4-41, Section 3, “CP Commands,” and “NCPDUMP Service Program and How To Use It” on page 4-42.

Considerations for Loading All 370x Control Programs

When using the NETWORK LOAD command to load EP for the 370x, the following events occur:

- Active users are disconnected.
- Active I/O operations are reset.
- Dedicated devices are detached and released.
- Dialed lines are released.
- Devices enabled but not dedicated are reset.
- Binary synchronous communication are reset.

Special Considerations for Loading the EP 370x Control Program

If a 370x Emulation Program (EP) is automatically reloaded after a 370x failure, the system may loop after the restart. The message

```
CTLR xxx UNIT CHECK; RESTART IN PROGRESS
```

and two responses

```
CTLR xxx DUMP COMPLETE
```

```
CTLR xxx ncpname LOAD COMPLETE
```

indicate that the 370x has been reloaded. If the system loops after the second response, you must reset all emulator lines from the 370x control panel.

If the automatic dump feature is not enabled, one of the messages

```
CTLR raddr UNIT CHECK; IPL REQUIRED
```

```
CTLR 'raddr' CC=3; DEPRESS 370X 'LOAD' BUTTON
```

indicates a 370x abnormal termination. You must reload the 370x Emulation Program with the NETWORK LOAD command. If the system loops when an attempt is made to enable the lines, you must reset all emulator lines from the 370x control panel.

The *Guide to Using the IBM 3704 Communications Controller Control Panel* and the *Guide to Using the 3705 Control Panel* describe the procedure for resetting emulator lines from the 370x control panel in their “Generating Channel End/Device End with Emulator Program” section.

Starting Operation

Messages On Startup and Initialization

In the course of VM/SP initialization or startup, in addition to the normal expected messages that are a part of loading the system, you may see other messages to which you must respond. These messages relate to functions of initialization, checkpoint, warm start, and automatic checkpoint requirements. These messages and the recommended course of action are described in *VM/SP System Messages and Codes*.

Control Messages after Startup

System console messages are in three categories:

- System operation status messages
- Virtual machine user messages
- Hardware and program status messages.

The operator action required depends on the type and content of the message.

System Operation Status Messages

These messages inform the system operator of the logging on and logging off of users, number of spooled files, output printer or punch accounting data, device attention conditions, and command completion indications. Some of these messages require operator action, while others do not.

Messages from Virtual Machine Users

These messages include requests and inquiries from virtual machine users on the system; a sampling of these user messages follows:

```
MSG FROM JONES: CAN YOU GIVE ME A TAPE
MSG FROM SMITH: CAN YOU PUT UP A SCRATCH 181
MSG FROM SMITH: PLEASE TAKE RING OUT OF 181 NOW
MSG FROM BROWN: PLS ATTACH 382
```

Hardware and Program Status Messages

These messages include VM/SP error recording messages, VM/SP system error information, and I/O error messages related to users of virtual machines who have invoked SVC 76. SVC 76 causes VM/SP to send a message to the system operator (for I/O error records only) and to record the error data in VM/SP's error recording area. The *VM/SP System Messages and Codes* manual describes these messages.

For more information about error detection and the recording and editing of errors, refer to the *VM/SP OLTSEP and Error Recording Guide*.

The following summarizes the action that the operator must take in response to an error message:

1. Determine the severity of the error.
2. Take a storage dump if the automatic dump routine failed.
3. Try to preserve spooling and accounting data by issuing the SHUTDOWN command if it has not been done automatically.
4. Try to force error recording if it did not occur.
5. Attempt to analyze the problem by whatever means at his disposal and if necessary, reallocate resources and priorities.
6. Call for technical assistance if necessary.

Whenever there are indications that I/O errors require further analysis or that basic integrity of a device needs to be determined, other service programs may be invoked to help make such an assessment. For example, to test the operational 3344 or 3350 DASD devices, the operator may invoke the OS/VS and DOS/VS Analysis Program-1 (AP-1). This utility program, if resident in the OS/VS or DOS/VS library of a virtual machine, can be used to check the proper operation of the device. The use of this program along with error messages and user action is detailed in the *OS/VS and DOS/VSE Analysis Program-1 (AP-1) User's Guide*, GC26-3855.

In the event that technical assistance is required, it can be obtained from the following sources:

- The customer's own system programmer(s).
- The service representative, who has special test equipment for hardware problem isolation.
- The IBM Program Support Representative, who can also assist with VM/SP system problem diagnosis.

You can assist with problem diagnosis by being alert for the following error recording messages:

```
I/O ERROR RECORDING AREA 90 PERCENT FULL; RUN CPERE
```

```
I/O ERROR RECORDING AREA FULL; RUN CPERE
```

You should then dump the error recording data to some other device (usually a tape drive) and clear the error recording area to insure enough space to record additional error records. Information on how to run CPERE is contained in *VM/SP OLTSEP and Error Recording Guide*. Additional information on EREP, the program to which CPERE passes control, is detailed in *Environmental Recording Editing and Printing (EREP) Program*.

Starting Operation

Other errors may not be severe enough to cause VM/SP system failure, but might cause a specific virtual machine to abnormally terminate. In most cases, the user can IPL the virtual machine again.

Should the VM/SP machine enter an unexpected WAIT state or loop, see the section entitled "How To Use the VM/SP Facilities To Debug" in *VM/SP System Programmer's Guide*.

3480 Magnetic Tape Subsystem Support

VM/SP supports the 3480 magnetic tape subsystem with the following restrictions:

- If you IPL a standalone program from a 3480, the 3480 is not assigned to the processor that IPLed the 3480. Because of this, do not IPL any other processor that has a path to the 3480, since the other processor might steal the 3480 by assigning it to itself.
- When a 3480 is assigned to one processor, another processor's attempt to perform I/O to the device (for example, write, read, forward space file) will receive an assigned Elsewhere Unit Check. The processor that owns the device assignment will not be able to perform I/O to the device until the other processor clears the contingent connection that the Unit Check established (for example, issuing a SENSE command or Signal System Reset). This problem can occur if a processor tries to IPL a 3480 device that is assigned to another system, but fails to clear the resulting unit check.
- When running Single Processor Mode, the MVS MP system can perform its I/O to a 3480 device from only one of its processors, either the V=R processor or the MVS native processor. Before the MVS native processor can use a 3480 device, VARY OFF the device from the CP system if the device is not already offline.

3800 Printing Subsystem Support

The 3800 models 1 and 3 are supported as dedicated, spooled, and virtual devices. The 3800 model 8 is supported as a dedicated and spooled device. The 3800 models 3 and 8 are supported in 3800 model 1 compatibility mode and provide the following:

- An improved pel density of 240 x 240 pels
- A 10 line per inch vertical spacing.

Following is a list of features for the 3800 printers:

- FORMS CONTROL lets the user select the amount of vertical space between printed lines. Lines can be vertically spaced at 6, 8, or 12 lines per inch. Users of the 3800 Model 3 printer have an additional option of 10 lines per inch. You can also mix different spacings on the same page.

- **FORMS OVERLAY** permits photographing of predefined data (forms) on one or more pages. The 3800 printer supports the creation of a form on a page and the immediate printing of data on that form.
- **PREDEFINED CHARACTER SETS** lets the user select previously designated character sets that have up to 64 characters each. Different character sets provide varied character types, sizes, and horizontal spacings between the characters. The 3800 supports the simultaneous use of two character sets and optionally supports the use of two additional character sets. The user can mix characters from the different character sets on a single page. Use the translate tables to reference the characters within the 3800. You can load up to four translate tables at any one time.
- **CHARACTER SET MODIFICATION** lets users modify and extend character sets.

Note: Due to the change in pel density, customized 3800 Model 1 character sets are not interchangeable with the 3800 Model 3 character sets. To convert 3800 Model 1 character sets, you can:

- Recode the customized character sets in the 3800 Model 3 pel resolution, and use the CMS GENIMAGE command to build new modules.
 - Use the MVS Character Conversion Aid to convert customized character sets to the 3800 Model 3 pel density. Then use the CMS GENIMAGE command to build new modules.
- **COPY MODIFICATION** permits printing of predefined data (for example, printing the same header on each page) or the suppression of printing for selected data.
 - **MULTIPLE COPIES** permits numerous copies of files and datasets to be printed without the use of multiple-ply paper. Numerous copies of a single 3800 buffer can also be printed.

The formatting of 3800 output is controlled by character arrangement tables, library character sets, graphic character modification modules, and forms control buffers. These are stored in an image library and are loaded into the 3800 before it prints a spool file. They may also be imbedded directly in the spool file. Use the START command to control information (including the default character set and FCB) for the 3800.

Dedicated 3800 Printing Subsystem Support

VM/SP lets virtual machines, including virtual VM/SP, attach the 3800 printer as a dedicated I/O device. When attached, the full capability of the 3800 is available to the virtual machine. All of the functions previously described are available when the 3800 is used as a dedicated device, provided the device is specified in the RDEVICE macro instruction at system generation. See the *VM/SP Planning Guide and Reference*.

Performance Options

VM/SP offers the following performance options to enhance the operating efficiency of one or more virtual machines. This enhancement is normally at the expense of the performance of other virtual machines. In certain instances, enhancements to a specific virtual machine (or multiple virtual machines) are detrimental to the operating efficiency of other virtual machines. The performance options are:

- Virtual machine assist
- Extended Control-Program Support: VM/370
- Reserved page frames
- Favored execution
- Virtual=real
- Locked pages
- Priority
- Affinity in an attached processor environment
- Queue drop elimination.

You can dynamically assign or withdraw any of these options except for the virtual=real option (it can only be dynamically withdrawn) from virtual machine users during system operations. You can assign to one virtual machine a “mix” of performance options, or the options may be assigned to separate virtual machines. Efficient use of these options depends upon the characteristics of the program applications; the number, size, and complexity of the virtual machines being run; and the size, I/O configuration, and processor model of the real computer being used.

In addition to using these performance options, you can also use the SET PRIORITY command to change a virtual machine’s execution priority and, thereby, change its performance.

For example, you could assign the following mix of options.

- One virtual machine:
Reserved Pages=10
Favored Execution=80%
Virtual Machine Assist
- Another virtual machine:
Virtual=Real
Priority=1
- A third virtual machine:
Locked Pages=2

The VIRT=REAL and reserved page options are unique in that both options cannot be applied to multiple virtual machines at the same time. However, no such

restrictions apply to the locked pages, virtual machine assist, Extended Control-Program Support: VM/370, priority, or favored execution options; they can be applied to multiple virtual machines.

Reserved Page Frames Option

The CP paging routine uses chained lists of available and pageable pages to control real storage use. CP assigns pages for users from the available list, which it replenishes from the pageable list.

Pages that are temporarily locked in real storage are not available or pageable. Paging proceeds using demand paging to select the best page for swapping. The reserved page option gives a particular virtual machine an essentially private set of pages. The pages are not locked; they can be swapped, but normally only for the virtual machine for which they are reserved. You specify the number of reserved pages for the virtual machine as a maximum. When the page selection routine selects an available page for a reserved user, it marks that page reserved if the maximum specified for the user has not previously been reached. If CP encounters an available (idle) reserved page for the reserved user during page selection, it is used whether or not the maximum has been reached.

Only an operator with privilege class A can specify the number of reserved pages. Issue the CP SET RESERVE command in the following format:

SET RESERVE *userid* *xx*

userid

is the name of the VM/SP user to whose virtual machine you are to assign the reserved page option.

xx

is the maximum number of pages you assign.

If the CP page selection routine cannot locate an available page for other users because they are all reserved, the routine uses one of the reserved pages as a last resort.

Favored Execution Option

The favored execution option guarantees that the specified virtual machine gets up to a fixed percentage of processor time. Favored in this sense means that CP provides execution time up to the specified percentage to that virtual machine, provided that virtual machine can use that much processor time. At regular time intervals, the CP dispatcher checks the processor time used by the favored virtual machine. If the guaranteed percentage is exceeded, the machine gets its normal priority for the remainder of the time interval. If the percentage used is less than the guaranteed value, the favored virtual machine has the highest execution priority until it reaches that threshold. You, as the system operator with privilege class A, specify the percentage of processor time assured by the favored option in the CP SET command. The format of the command is:

Starting Operation

SET FAVORED *userid* [*xxx*]
[**OFF**]

userid

is the name of the VM/SP user whose virtual machine is to be assigned the favored option.

xxx

is any percentage value from 1 to 100.

OFF

operand stops favored operation.

You can assign multiple virtual machines the favored execution option with, and/or without a percentage value.

Example:

```
set favored operatns 75
set favored payroll 25
set favored revenue
set favored hotjobs
set favored hotjobs 10
```

Although the SET FAVORED command will prevent the specifying of more than 100% for a particular virtual machine, nothing is done to prevent more than 100% being allocated to a number of virtual machines. In situations where more than 100% has been allocated, the individual favored virtual machines will compete for the available resources on a pro-rata basis: an individual virtual machine's allocation will be roughly proportional to the percentage allocated to it divided by the total percentage allocated to all virtual machines. The effect that allocating more than 100% of the system has on interactive (Q1) response is unpredictable.

Virtual=Real Option

The virtual=real option permanently relocates the CP nucleus (except for real page 0) to provide space in real lower storage to contain the largest virtual=real machine. For the virtual machine, each page from page 1 to its last or *n*th page is in its true real storage location; only its page zero is relocated. The virtual machine still runs in relocate mode, but because each virtual page frame address is the same as the real page frame address, the virtual machine requires no CCW translation for the virtual machine. Because no CCW translation is performed, no check is made of the I/O data addresses. The virtual machine must ensure that no I/O data transfer occurs into real page zero or beyond the upper boundary of the virtual machine's storage. Failure to observe these limits can cause damage to the VM/SP system or to other virtual machines.

Initial program loading of a named or shared system is not allowed in the virtual=real machine; the hexadecimal device address must be used.

Several aspects of the operation of the virtual=real option can affect overall system operation:

1. The area of contiguous storage built for the virtual=real machine must be large enough to contain the entire addressing space of the largest virtual=real machine.
2. Only virtual machines with the virtual=real option can use the real storage reserved for virtual=real machines. Only one virtual=real virtual machine can run at a time. The storage is not available for other users, nor for VM/SP use, unless the virtual=real machine is not logged on and you issue an UNLOCK command with the VIRT=REAL option. When you issue this command, all page frames previously reserved for the virtual=real machine are permanently relinquished to the available-page pool. Once these virtual=real page frames are available to the page pool, no virtual=real machine can run until you load VM/SP again (via IPL). For this reason, it is suggested that the virtual=real machine should be a machine with high availability and heavy workload demands. It is not possible to relinquish only some of the virtual=real page frames (if, for example, a smaller virtual=real machine is to be run). All virtual=real page frames are locked until all are released to the pool.
3. The virtual=real machine operates in its allocated storage area with normal CCW translation in effect until the virtual machine operator issues the CP command:

```
SET NOTRANS ON
```

At this time, all subsequent I/O operations occur from the virtual CCWs in the virtual=real machine without translation. With NOTRANS on, the virtual machine must not perform I/O operations into page zero or beyond its addressable limit. As mentioned previously, violation of this requirement can cause damage to the VM/SP system or to other virtual machines.

4. If the virtual=real machine performs a reset or IPL, the normal CCW translation routine regains control until the virtual machine operator again issues the SET NOTRANS ON command. Only the virtual=real machine operator can use the command. An operator message occurs if the virtual machine enters normal translation mode.

The virtual=real option is used with programs or operating systems that dynamically modify channel programs, or that wish to avoid CP virtual channel program translation and paging degradation. If virtual=real operation is desired, it must be specified at VM/SP system generation time. Virtual=real assigns the same real storage locations to the virtual machine that exist on the real machine (except for page zero, which is relocated).

For example, in Figure 2-1 on page 2-30, if a DOS system needs 160K bytes to run virtual=real, CP puts all but the first 4096 bytes of the program at the real storage address locations. Page 0 of the DOS supervisor is the only part of the DOS virtual machine that can be relocated, and it follows the last real page frame of the DOS machine. Figure 2-1 on page 2-30 also shows that the major portion of the CP nucleus is permanently relocated into upper storage (except for page

Starting Operation

zero) whether the virtual=real system is operating or not. The real page frame zero remains under control of VM/SP. CP dedicates 160K nonpageable bytes to the DOS machine.

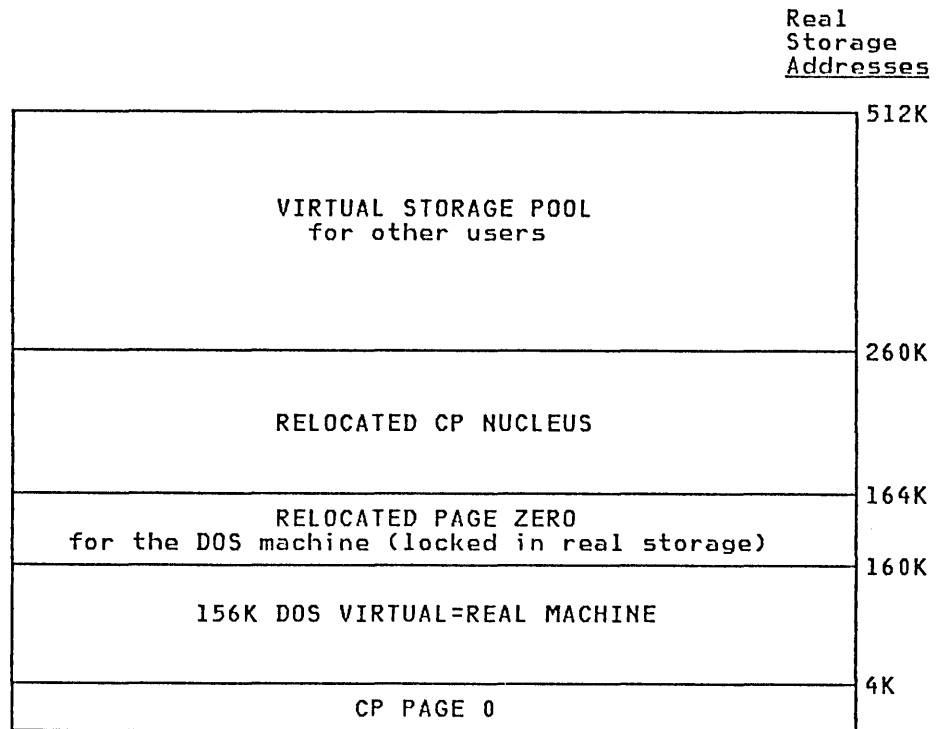


Figure 2-1. DOS 160K Virtual=Real Storage Assignments

However, when the user no longer needs the virtual=real machine performance option and he logs off, the assigned virtual=real storage can be reallocated as available pageable storage for use by other virtual machines (assuming that no other users with the VIRT=REAL directory have need of the area). The reallocation of virtual=real storage as usable and available storage for other virtual machines is controlled by the UNLOCK command and the VIRT=REAL operand. Once the pages are released, no virtual=real machine can be run unless VM/SP itself is reloaded. If more than one virtual machine needs the capabilities of running virtual=real (at different times), VM/SP system generation parameters must specify the largest virtual machine that is to run in the virtual=real environment. A more practical approach may be to generate a different version of VM/SP for each virtual=real machine, if the storage sizes are substantially different.

Locked Pages Option

If sufficient page frames are available for other users, you can choose to lock (fix) one or more specific pages of a virtual machine or the pageable CP nucleus in real storage. This performance option differs from the other options in that the LOCK command can apply to several virtual machines. It is also possible to have the reserved page frames option apply to one user's virtual machine, and then have the lock option apply to another user's virtual machine. A locked page is effective from the time you invoke the lock option until the user logs off the system, or until you issue the UNLOCK command. If the page or pages that are locked have a high frequency of reference or change activity, the lock option, like the reserved option, can cause an improvement in the user's virtual machine performance or overall system performance because of the reduction in paging activity.

Notes:

1. If a named system currently has the locked pages option in effect, reloading of the same named system, loading of another named system, or a system reset does not affect the locked pages option of a virtual machine. The loading of named systems over other named systems without using the UNLOCK command beforehand results in pages being made unavailable for use if the first named system has locked pages in effect.
2. In systems generated for AP and MP operation, shared pages cannot be locked.
3. You should always prefer the reserved page frames option to the locked pages option, if it is not already in use. The LOCK command locks specific pages into real storage regardless of activity, while the reserved page frames option keeps all the currently most active pages in real storage.

Priority

You can assign specific (dispatch) priority values to different virtual machines to override the assigned or default priority value for each user in the VM/SP directory. In doing so, the virtual machine with a lower *nn* value is considered for dispatching before a virtual machine with a higher *nn* value. You set user priorities by using the following class A console function:

SET PRIORITY *userid nn*

userid

is the user's identification.

nn

is an integer value from 1 to 99. The higher the *nn* value, the lower the dispatching priority.

Starting Operation

VM/370 Assistance - Hardware Assist

The overhead associated with CP's handling of virtual machines and their collective I/O resources is extensive and time consuming. To reduce this overhead, CP is assisted by System/370 processor hardware that duplicates some of the functions provided by CP routines. This hardware assistance to program function is called VM/370 hardware assist. Some form of VM/370 hardware assist is available on most, but not all VM/SP-supported processors. For a complete list of these processors that support VM/370 hardware assist and ECPS:VM/370 see *VM/SP Planning Guide and Reference*. VM/370 hardware assist on certain processors can consist of virtual machine assist; on other processors VM/370 hardware assistance is composed of virtual machine assist and VM/370 Extended Control Program Support (ECPS:VM/370).

VM/370 hardware assist is activated when VM/SP is loaded or by the use of CP commands, and it is deactivated by CP commands. For the command that activates/deactivates the VM/370 hardware assist process, refer to the SET command described in Section 3, "CP Commands." The status of VM/370 hardware assist may be determined by the use of the QUERY command also described in Section 3, "CP Commands."

Virtual Machine Assist

The virtual machine assist facility can be ordered via feature number, it is available as an RPQ (Request for Price Quotation), and is standard on some processors. For a list of VM/SP supported processors and how they support virtual machine assist and ECPS:VM/370 see *VM/SP Planning Guide and Reference*. On attached processor or multiprocessor systems, virtual machine assist may be installed on one or both processors. Virtual machine assist relieves CP of some of the processing overhead caused by the simulated execution of certain instructions, privileged operation instructions, and interrupts of virtual machines. On/off system control of virtual machine assist is provided by the A privilege class command:

```
SET SASSIST { ON } [ [PROC] nm ]  
            { OFF }
```

Users of virtual machines, in turn, have discrete control over the functions of virtual machine assist by means of directory entry options or by command. For control of this function on a virtual machine level, refer to the privilege class G command SET ASSIST, described in *VM/SP CP Command Reference for General Users*.

Extended Control-Program Support:VM/370

Extended Control-Program Support:VM/370 is an expansion of the function provided by virtual machine assist and consists of three major parts:

- Expanded virtual machine assist
- CP assist
- Virtual interval timer assist.

Expanded virtual machine assist is an expansion of function provided by virtual machine assist in that, via hardware, CP is relieved of processing some conditions of CP simulated instruction execution that were not covered by the virtual machine assist process. Expanded virtual machine assist also assists CP in the processing of other instructions and privileged operations not emulated by virtual machine assist.

Expanded virtual machine assist is activated for all virtual machines when VM/SP is loaded. It is also activated when both class A commands are invoked as shown:

```
SET SASSIST ON
```

```
SET CPASSIST ON
```

Expanded virtual machine assist is deactivated when either SET SASSIST OFF or SET CPASSIST OFF is invoked.

The CP assist portion of the Extended Control-Program:VM/370 Support duplicates via hardware the function provided by various high-usage significant overhead portions of CP routines. CP assist is activated upon system initialization or by invoking the privilege class A command:

```
SET CPASSIST { ON } [ [PROC] nn ]  
             { OFF }
```

When the OFF operand of CPASSIST is invoked, CP assist is deactivated (as well as expanded virtual machine assist portion of ECPS, if previously active).

For attached processor applications, specify PROC nn to activate or deactivate CP assist on the desired main or attached processor. If you desire the same setting for both processors, omit the PROC nn option and use the ON or OFF options to activate or deactivate CP assist for both the main and attached processor at the same time.

The third function provided by ECPS:VM/370 is virtual interval timer assist. This hardware assist provides virtual machines with a more accurate method of updating the virtual machine's interval timer (location X'50'). Overall system control of this function is by the class A commands SET SASSIST ON and SET SASSIST OFF. Control of this facility on the individual virtual machine level is by the class G command SET ASSIST TMR and SET ASSIST NOTMR. For more details on this, refer to the *VM/SP CP Command Reference for General Users*.

To summarize the use of the SET CPASSIST and SET SASSIST in relationship to virtual machine assist and Extended Control-Program Support:VM/370, see Figure 2-2 on page 2-34.

Starting Operation

VM/370: Hardware Assist	SET SASSIST ON SET CPASSIST OFF	SET SASSIST ON SET CPASSIST ON	SET SASSIST OFF SET CPASSIST ON	SET SASSIST OFF SET CPASSIST OFF
Virtual Machine Assist	X	X		
Virtual Interval Timer Assist	X	X		
Extended Virtual Machine Assist		X		
CP Assist		X	X	
Legend: X = function activated				

Figure 2-2. Command Control of VM/370 Hardware Assist

Note: When you install the CPFRET Trap, CP disables CP Assists DSP1, DSP2, UNTFR, FREE, and FRET. This causes performance to lower for systems that use CP Assists. However, this lowering in performance is not expected to be a problem when CP traps suspected free storage problems. When the trap is not installed, system performance remains the same.

VM/370 Hardware Assist Restrictions

A virtual machine running in certain virtual machine modes or running certain operations cannot use certain facilities provided by VM/370 hardware assist. When these modes or operations are encountered, the action taken is dependent upon the processor facilities of either virtual machine assist or Extended Control-Program Support:VM/370.

For all VM/SP supported processors that have VM/370 hardware assist active when the CP ADSTOP or certain CP TRACE command functions are encountered, the virtual machine assist is deactivated during the interval of their use.

For System/370 Models 135-3, 138, 145-3, and 148, the use of the DOS emulator or PER (Program Event Recording) curtails some of the function provided by Extended Control-Program Support:VM/370. For more details on VM/370 hardware assist, see the *VM/SP System Programmer's Guide*.

Affinity

In attached processor or multiprocessor systems a virtual machine may be designated to execute only on one specified processor. This may be advantageous for performance or functional requirements. Performance gains might be realized with affinity because of the nature of the workload or the availability of performance assists on one processor. Functional requirements could be the existence of required operation codes on one processor. In either case, CP code executed on behalf of the virtual machine may execute on either processor.

Queue Drop Elimination

VM/SP attempts to optimize system throughput by monitoring the execution status of virtual machines. When a virtual machine becomes idle, VM/SP drops it from the active queue. The virtual machine's page and segment tables are scanned, and resident pages are invalidated and put on the flush list.

In certain cases, where cycle of queue dropping and reactivation is repeatedly executed, there is a significant increase in the overhead involved in invalidating and revalidating pages. One example of this is the SNA service virtual machine processing IUCV messages.

The SET QDROP userid ON/OFF class A command allows an installation to control this situation. If SET QDROP OFF is in effect for a virtual machine, that virtual machine's pages are not scanned or flushed when the machine becomes idle. Specifying SET QDROP OFF for a service virtual machine can improve system performance and throughput when queue dropping would otherwise occur frequently. Specifying SET QDROP OFF USERS allows the QDROP OFF status to be extended to any virtual machine communicating via VMCF or IUCV to a (server) virtual machine for which SET QDROP OFF has been specified. Thus, performance gains can be realized in systems with heavy usage of products such as IFS or PVM (invoked via the CMS PASSTHRU command). Note, however, that indiscriminate use of the SET QDROP OFF command could degrade system throughput by forcing page stealing to take place.

System Termination

The VM/SP system may be shut down in one of two ways:

1. By an operator initiated shutdown
2. By an abnormal termination of operation.

The normal shutdown is initiated by a class A operator entering the SHUTDOWN command from the operator's console. The format of the SHUTDOWN command is:

```
SHUTDOWN [REIPL  
          [POWEROFF]
```

If the SHUTDOWN command is ineffective, press the LOAD button on the system console once to shut down the system (a second pressing of the load button reloads VM/SP).

Note: When the LOAD button on the system console is used, spooling and accounting files can be lost because the files are not closed properly.

The POWEROFF option is valid only for 4361 processors. When you specify this option for a 4361 processor, the system shuts down normally, and the processor is powered off. If the 4361 processor has the Auto Start feature, you will not be prompted to set the time-of-day clock when the next IPL takes place.

Starting Operation

The shutdown operation checkpoints spooling and accounting functions; in addition, user activity is terminated by automatically logging off all virtual machines and disabling communication lines. Open and active files on the spool unit record devices are purged and, if applicable the SUSPEND order is issued to all 3851 device addresses (suspends MSC interface activity to the host virtual machine). Note that the SUSPEND order issued to a Mass Storage Control interface may cause the SHUTDOWN command to run longer than usual.

Note: Before issuing the SHUTDOWN command, a message or warning should be sent to all logged-on users so that they can quickly bring their virtual machines to an orderly halt. Time should also be allotted to the class D (spooling) operator so that he too can terminate in an orderly manner the active files being processed on the unit record devices.

When you specify the REIPL option with the SHUTDOWN command, the system shuts down and does an automatic warm start. For this type of automatic restart, you do not need to set the time-of-day clock or choose the type of start you want the system to do. In fact, you do not need to be present for the automatic warm start to occur.

The dump unit for VM/SP system failures is specified in the SYSDUMP operand of the SYSOPER macro during VM/SP system generation, but can be changed by a privilege class A or class B system operator.

If the dump unit is set to disk (by default or via the SET DUMP AUTO command) at the time of system failure, the system dumps all or parts of real storage to the specified disk and automatically restarts the VM/SP system.

When automatically restarted, the system preserves all accounting information and spool file data on disk, performs an automatic logon of the primary system operator, restores the system LOGMSG, and continues system operation. It is not necessary to re-enable the lines, since dump and re-IPL automatically re-enable them. The VM/SP online message is sent to the terminal users, indicating that they must log on again.

When the system is automatically restarted after a system failure, the system operator is automatically logged on only if he was logged on the primary system console at system failure time. If logged off, disconnected or logged on elsewhere, the operator must explicitly log on.

If, at the time of the system failure, the dump unit is set to a printer or tape, the VM/SP system writes the dump on the specified unit, preserves the spooling and accounting data, and stops. The operator must then re-IPL the VM/SP system as for normal system start up, specifying a WARM START to preserve the accounting and spooling file data. The operator must re-enable the communication lines to permit users to log on again.

System Abend Dumps

Conditions can occur within the CP program that may force an abnormal ending condition (ABEND) and cause the dumping of system registers and storage. The device that receives these records can be a tape, printer, or disk device.

Dumping operations are caused by any program interruptions or system restart condition. These interruptions cause routines to gather data from registers and storage and place this data on a previously defined device. The system CP command SET DUMP defines the quantity of data to be dumped. For example:

```
SET DUMP AUTO
SET DUMP raddr
SET DUMP raddr ALL
SET DUMP AUTO ALL
```

SET DUMP AUTO places the VM/SP system dump on a preselected file device. (The disk dump area is automatically selected at system initialization time if sufficient contiguous space is available.) You can verify the device type and address by entering the QUERY DUMP command.

Use SET DUMP *raddr* when the dump device is to be a high speed printer or tape device. Substitute the real hexadecimal address for *raddr*.

The ALL operand used with SET DUMP AUTO or SET DUMP *raddr* dumps all of storage onto to the DUMP device. If the ALL operand is not specified, the system defaults to dumping only those areas that pertain to CP, and not those areas that pertain to virtual machine operations.

If you dump to a disk file, an additional operation is necessary to transform these records into readable output for programmers or system analysts. You can do this with the CMS IPCSDUMP program.

Note that expanded function is available to the IPCSDUMP command as generated for the IPCS virtual machine.

If the records are dumped onto a tape drive, other CMS command options must be invoked for printout.

Only abend dumps that are a result of using the SET DUMP AUTO command are spooled as a special virtual card reader file. This card reader file is assigned during system generation to a specific virtual machine user via the SYSOPR macro. The CMS IPCSDUMP command formats and prints these CP abend dumps.

The IPCSDUMP command creates a CMS file from the CP disk dump data and prints the dump from the CMS file.

Starting Operation

VM/SP Interactive Problem Control System (VM/SP IPCS) provides installations with expanded facilities for reporting and diagnosing software failure. The VMDUMP command allows the user to dump a virtual machine's storage area. The VM/SP IPCS component or a user written program must process the file created by the VMDUMP command. For additional information, see the *VM/SP Interactive Problem Control System Guide*.

Section 3. CP Commands

The CP commands are interactive console functions that you can use to control the VM/SP system. CP commands let you control your virtual machine. The commands that you can issue depend on your assigned privilege class(es), as described in the following section. If appropriate, CP shows its processing results with responses (for example: `COMMAND COMPLETE`, `MISSING ARGUMENT`, or `INVALID OPTION`).

Privilege Classes for CP Commands

Each CP command has one or more of the following function types:

1. Operations
2. Resource
3. Programmer
4. Spooling
5. Analyst
6. CE (Customer Engineer - Service)
7. General.

The IBM-defined class structure is based on these seven function types. A command keeps its function type even if your installation establishes its own class structure. In most cases, each command class (A-G) has a corresponding function type (O,R,P,S,A,C or G). Some commands fall into more than one class. Figure 3-1 on page 3-2 shows the function of each privilege class and function type. Refer to the appendix for a list of each CP command, its IBM-defined privilege class, function type, and a description of what each command does.

Your installation will assign each user, as part of the user's entry in the directory, one or more privilege classes. The exceptions are users with a password of `NOLOG`. These users have no privilege class and can only:

- Send messages
- Receive spooled output as punched cards or printed forms.

The `NOLOG` password identifies them to receive spooled output when a virtual machine user spools output for them.

If a user tries to issue a command that does not have his command class, CP does not execute the command and issues an error message. The CP command

CP Commands

descriptions for classes A-F are in this book. For descriptions of the class G commands refer to the *VM/SP CP Command Reference for General Users*.

Note: If your installation adds or removes any commands from the general user class (IBM-defined class G), your installation should update the HELP files to show these changes. See *VM/SP CMS User's Guide* for information on "Tailoring the HELP Facility."

IBM-Defined Class	Function	Function, Primary User, and Use
A ¹	O	<p>1. Operations - Primary system operator The system assigns class A to the user at the VM/SP console during IPL. The class A user is responsible for VM/SP's availability and its communication lines and resources. These commands control system accounting, broadcast messages, run virtual machine performance options and affect VM/SP performance.</p> <p>Note: The Class A system operator who is automatically logged on during CP initialization is designated as the primary system operator.</p>
B ¹	R	<p>2. Resource - System Resource Operator These commands control allocation and deallocation of real resources of the VM/SP system, except those that the primary system operator and the spooling operator control.</p>
C ¹	P	<p>3. Programming - System programmer These commands update functions of the VM/SP system and change real storage in the real machine.</p>
D ¹	S	<p>4. Spooling - Spooling operator These commands control spool data files and specific functions of the system's unit record equipment.</p>
E ¹	A	<p>5. Analyzing - System analyst These commands examine and save certain data in the VM/SP storage area.</p>
F ¹	C	<p>6. CE -Service Representative (Customer Engineer) These commands get and examine data about input and output devices connected to the VM/SP system.</p>
G ²	G	<p>7. General - General User These commands control functions to run users' virtual machines.</p>
Any ²	None	<p>These CP commands are available to any user. These are to gain and take away access to the VM/SP system.</p>

¹This book describes classes A, B, C, D, E, and F.
²Classes G and Any are described in *VM/SP CP Command Reference for General Users*.

Figure 3-1. CP Privilege Class Descriptions

How to Use this Chapter

This chapter describes CP commands available to users with the privilege classes of A, B, C, D, E, and F. These privilege classes are assigned to users who manage the VM/SP system and its spooling and hardware resources.

Note: The privilege class G commands control the functions of the general user's virtual machine. Refer to the *VM/SP CP Command Reference for General Users* for information on all G privilege class commands and for information on commands with the class of "Any." The Group Control System (GCS) operator commands are described in the *VM/SP Group Control System Guide*.

Command Description Formats

The descriptions of each command in this chapter have the following sections:

1. **When to Use the Command** — Describes what you can do with the command and when to use it.
2. **Command Format and Operands** — Lists the command format and command operands. Refer to "Commands, Subcommands and Variables" for information about abbreviations, etc.
3. **Things You Should Know about the Command** — Lists notes about the command that pertain to more than one subcommand, or that pertain to specific hardware, etc. Some command descriptions may not require this section.
4. **System Response** — Lists the standard system responses from invoking the various commands and subcommands. Some command descriptions may not include this section. In those cases, there is either no system response, or the system response is self-explanatory.

Note: For the QUERY command description, the system responses for each parameter are included with the parameter description, instead of a separate section titled "System Response".

Commands, Subcommands and Variables

When you issue a command you may not need to enter the entire command name. In many cases, you may either truncate or abbreviate the name.

- For truncating, in the command format, uppercase letters represent the shortest version of the command name that you can enter. (Remember, however, that you can enter VM/SP commands with any combination of uppercase and lowercase letters.)

Example:

INDicate means that you can enter **IND INDI, INDIC, INDICA, INDICAT,** and **INDICATE** to execute this command.

CP Commands

If there is no minimum truncation listed, enter the entire word (represented by all capital letters).

- You can also abbreviate the name of the command. Abbreviations for command names appear below the full name in the format box. Abbreviations for operands are with the operand descriptions that follow the format box.

Example:

Reader

RDR means that the abbreviation is **RDR**. Thus, you enter **READER** as **R, RE, REA, READ, READE, READER, or RDR**.

Lowercase letters, words, and symbols that appear in the command format box in *italics* represent variables that you will substitute with specific information. For example, *hexloc* indicates that you should enter a hexadecimal storage location address.

Symbols Used in Command Descriptions

The following symbols help define the command format. Do not type them when you enter the command.

- Braces { }

When the operands are in braces, you **must** select one of the operands.

Example:

$\left\{ \begin{array}{c} A \\ B \\ C \end{array} \right\}$

means that you **must** specify either A, B, or C. If a list of choices is not enclosed by brackets or braces, treat it as if it is enclosed by braces.

- Brackets []

When the operands are in brackets, you **may** select one of the operands, but it is not a requirement.

Example:

$\left[\begin{array}{c} A \\ B \\ C \end{array} \right]$

means that you can enter A, B, or C, but you may omit the field.

- Ellipsis ...

When you see an ellipsis following an operand, you may use the operand or group of operands more than once in succession.

Example:

`{spoolid1 spoolid2...}`

means that you can code more than the amount of spoolids shown.

- Underscore _____

Defaults are represented with underscores.

Example:

A

B

C

means that B is the default. You can choose A, B, or C.

However, if you do not enter any choice the system will assume B.

Use the following symbols as specified in the format box.

asterisk	*
colon	:
comma	,
equal sign	=
hyphen	-
parentheses	()
period	.

When there are nested braces or brackets on the text lines, the nested operands that you choose depend on the higher level operand that you choose. The highest level of nesting is the operand that is enclosed in only one pair of brackets, and the lowest level of nesting is the operand that is enclosed by the maximum number of brackets.

Example:

```
[level 1  [level 2  [level 3] ] ]
[filename [filetype [filemode] ] ]
```

Lets you select a file by:

- Filename only
- Filename and filetype only
- Filename, filetype, and filemode.

You cannot select filetype alone because filetype is nested within filename. The same is true if you want to select filemode; you must also select filename and filetype.

ACNT

ACNT

Privilege Class: A

When to Use ACNT

Use the ACNT command to do the following:

1. Process accounting data
2. Create accounting records for logged-on users
3. Reset the accumulated accounting data
4. Close the spool file that is accumulating accounting records.

CP sends a message to each user that indicates the accumulated accounting data, and resets the accumulated accounting data for each user to zero. (Enter the CP command, SET ACNT OFF, to suppress the accounting message.)

ACNT Format and Operands

ACNT	$\left\{ \begin{array}{l} \text{userid1 [userid2 ...]} \\ \text{ALL [CLOSE]} \\ \text{CLOSE} \end{array} \right\}$
------	--

userid

specifies the user(s) for which you want accounting records created. You can specify more than one user on the command input line.

For each user that you specify, ACNT creates an accounting record. This record gives the time since the user logged on or the time since you last issued the ACNT command for the user. The record that ACNT produces is identical to the record that the system gives when the user logs off of VM/SP.

ALL

creates accounting records for all logged-on users.

CLOSE

immediately closes the spool file that ACNT used to accumulate accounting records. CP sends the records to the virtual punch or reader (whichever you specified as the output device on the SYSACNT macro instruction).

If you specify CLOSE with ALL, CP will perform the ALL function first. Then, CP will do the CLOSE as described.

System Response

CONNECT=hh:mm:ss VIRTCPU=mmm:ss.hh TOTCPU=mmm:ss.hh

CONNECT hh:mm:ss

is one of the following:

- The actual clock time spent in the user's current terminal session
- The time since you issued the last ACNT command for the user.

VIRTCPU mmm:ss.hh

is the amount of processor time to execute instructions within the virtual machine.

TOTCPU mmm:ss.hh

is the total of:

1. The amount of processor time to execute instructions within the virtual machine, and
2. The amount of processor time that CP used to maintain the virtual machine (paging, I/O).

ATTACH

ATTACH

Privilege Class: B

When to Use ATTACH

Use the ATTACH command when the user requests that you do any of the following:

- Logically connect a real device to a virtual machine for that virtual machine's exclusive use
- Logically connect a DASD device for CP access and control
- Dedicate all devices on a certain channel to the specified user.

ATTACH Format and Operands

ATTach	$raddr$ TO $\left\{ \begin{array}{ll} \textit{userid} & [\textit{AS}] \textit{vaddr} [\textit{R}/\textit{O}] \\ \textit{SYSTEM} & [\textit{AS}] \textit{volid} \end{array} \right\} [\textit{3330V}] [\textit{volid}]$ $\left\{ \begin{array}{l} \textit{raddr...} \\ \textit{raddr-raddr} \end{array} \right\} [\textit{TO}] \textit{userid} [\textit{R}/\textit{O}] [\textit{3330V}]$ CHAN nel <i>c</i> [PROC <i>nn</i>] [TO] $\left[\begin{array}{l} \textit{userid} \\ * \end{array} \right]$
---------------	---

raddr

raddr...

raddr-raddr

is the real address of the device you want to attach. On the command line, you can specify:

- One real device address
- More than one real device address
- A range of real device addresses.

With the range form of *raddr* (*raddr-raddr*), enter the values *cuu-cuu* with no imbedded blanks. For the range, you must supply contiguous addresses, and the second address in the range must be greater than the first.

When you specify more than one device or a range of devices, ATTACH processes all real device addresses, even if ATTACH meets errors while it is trying to attach the devices.

You can specify up to 48 multiple addresses on the ATTACH command line. If you try to attach more than 48 addresses, the system will:

- Only process the first 48 addresses
- Ignore the remaining addresses and will not issue an error message for these device addresses.

You may specify more than 48 addresses, however, by specifying the addresses in a range.

Example:

```
attach 108-10F user1 r
      if real devices 10A and 10D were not ready and online, this
      command would attach devices 108, 109, 10B, 10C, 10E, 10F.
```

[TO] *userid*

identifies the user of the virtual machine that will receive the device(s). The user must be logged on to receive the device(s).

If you want to attach the device to your own virtual machine, specify *userid* as an asterisk (*). You must use TO when:

- You specify a *userid* with 3 numbers or less in a range of addresses or multiple device addresses.
- You specify a *userid* that could also be a valid hexadecimal device address (for example, virtual machine id CE).
- You specify a *userid* as "to" or "t".

[AS] *vaddr*

is the virtual address (cuu) you will assign to the device. If you omit the optional keyword AS, *vaddr* cannot be A.

Be careful to avoid a subchannel protocol conflict in the virtual control unit (VCU) when you specify *vaddr*. Subchannel protocols for all devices that VM/SP supports are listed in Appendix B under "Virtual Device Characteristics" in the *VM/SP Planning Guide and Reference*. CP checks for a conflict in the VCU. If CP finds a conflict, CP does not do the operation and sends an error message to you. The first device attached to the VCU sets the subchannel protocol. For the ATTACH command with multiple devices, CP creates the devices in the order that you specify.

The following is an ATTACH statement that, when trying to attach real I/O devices to a virtual machine, is restricted by the subchannel protocol.

USER Wilson:

```
cp define graf as 101 3270
```

OPERATOR:

ATTACH

CP ATTACH ZERO TO WILSON AS 10E

CP SENDS THE OPERATOR THE FOLLOWING MESSAGE:

DMKVDE331E DEVICE 10E NOT ATTACHED - USE A NONSHARED VCU
INSTEAD

In the above example WILSON already had a device in the VCU range (100-10F) that needed SHARED support. In this case, you should contact WILSON and do one of the following:

- ATTACH the device to a different virtual address.
- Ask WILSON to DETACH or move (with DEFINE) all devices in the 100-10F range (i.e. device 101).

Note: Such conflicts can also arise when the ATTACH command creates an address on a VCU, which was created during LOGON by a device defined in the user's directory entry.

If a device that has minidisks is mounted and ready after the system loads VM/SP, attach the device to the system. After you have done a DASD ATTACH, users who had been logged on to VM/SP and who have a directory entry for the minidisk can issue the LINK command to access the device.

R /O

(or **R**) lets you apply read-only protection to a user's disk by using it for attached DASD devices.

If you specify R/O with a range of addresses or with more than one address, it will only apply to DASD addresses. CP ignores the R/O operand for all other device addresses.

[TO] SYSTEM

tells ATTACH, along with *valid*, to make the DASD device available to CP. CP can then access users' disks.

[AS] *valid*

is the volume label on the device that CP is attaching. When you use *valid* to attach:

- A *raddr* that is a 3330V to a userid, CP mounts the MSS volume with the label *valid* on the *raddr* during the attach.
- A *raddr* that is a 3330V to the system, CP mounts the MSS volume with the label *valid* on the *raddr* during the attach.
- A *raddr* that is not a 3330V to a userid, the *valid* operand is invalid.

- A *raddr* that is not a 3330V to the system, CP verifies the volume label of the volume on *raddr*.

3330V

tells CP to treat the *vaddr* as a 3330V. CP assumes the virtual machine has MSS device support and does not invoke MSS cylinder fault support for the *vaddr*.

Do not attach virtual 3330 volumes (3330V) to the system; this applies only to 3330V devices that were generated (at system generation) with FEATURE=VIRTUAL on their RDEVICE macro. In this case, you cannot use 3330V for minidisks, CP residence, paging, or spooling functions. On the other hand, if the system programmer generated 3330V devices with FEATURE=SYSVIRT on their RDEVICE macro, you can and should attach the 3330V devices to the system.

CHANnel *c*

is the real address of the channel you will attach to the user. *c* can be any hexadecimal character from 1 through 9 or A through F. You can only use CHANNEL for real channels whose devices do not have defined alternate paths.

The device addresses that the virtual machine user specifies must be the same as the real addresses (he cannot use minidisks). You do not have to attach the devices one at a time on the attached channel to the user; he can use any or all of the devices. However, even if the devices are idle, you cannot attach the devices on the channel to another user.

When you issue CHANNEL, CP checks if the SHARED and NONSHARED device types are defined to the same virtual control unit (VCU). If CP detects a conflict, it does the operation for all devices except those that would cause a subchannel protocol conflict. CP sends an error message for the devices not attached. However, to permit migration of data on 3420 tapes (a shared device) to the 3480 (a nonshared device), VM/SP will not enforce this restriction.

The first device applied to the VCU sets the subchannel protocol. CP attaches the devices in ascending order by device address. See Appendix B in the *VM/SP Planning Guide and Reference* for a complete listing of the Subchannel Protocols.

Notes:

1. If you issue an ATTACH CHANNEL command right after a DETACH CHANNEL command, you may get a message that the specified channel is in use. DETACH CHANNEL processing schedules I/O at different times for each dedicated device; but, DETACH CHANNEL processing does not wait for all I/O to finish before it returns control to the terminal operator. Because of the timing, the ATTACH CHANNEL command may fail.

ATTACH

2. A virtual machine should not issue a Clear Channel to any dedicated channel; to do so would cause unpredictable results.

PROC *nn*

is the address of the processor that owns the channel. CP does not require this operand in uniprocessor or attached processor mode. If you do specify PROC, CP will check its validity.

[TO] [*userid*] *

identifies the user of the virtual machine that will use the dedicated channel. The user must be logged on to get control of the channel.

If you want to attach the channel to your virtual machine, specify the *userid* as an asterisk (*). If you omit the optional keyword TO, *userid* cannot be TO or T.

Things You Should Know about the ATTACH Command

1. You can attach a disk to "SYSTEM". If no one accessed the disk when it was on the system, you can attach this same disk to a user without first detaching the disk.
2. If a user finds he is using certain devices many times, he may want to attach these devices to his virtual machine with the DIRECTORY DEDICATE statement. Thus, when the user logs on, these devices (if online) automatically attach to his virtual machine.

When virtual DASD volumes are associated with the 3850 Mass Storage System, the ATTACH command can attach 3330V devices to a virtual machine, even if the volume is not mounted. This is because CP does not stage cylinder 0 on a virtual volume, until you issue an MSC mount order that associates the virtual unit address with a volume serial. Even if you issue a mount order before the ATTACH, the host system may issue another MSC mount order to change the data staged on the virtual volume.

3. If CP attaches a device with advanced features before the device is on, the advanced features will not work.
4. If the device you want to attach is:
 - A 370x, VM/SP cannot use it as a 370x unless the system programmer loaded the 370x by VM/SP, DOS/VS, OS/VS, or ACF/SSP with the 270x EP program.

The ATTACH and DETACH commands reset the loaded status.

- A 370x in PEP mode, first disable all emulator lines with the DISABLE command. If you reload a 370x after an ATTACH, all EP lines will become disabled.

- A 2305, specify both the real device address and the virtual device address as the first exposure on the 2305 (that is, device address 0 or 8). When CP attaches or detaches a dedicated 2305 to a user, CP processes all eight exposures.
- A device that has the ECMODE option, any address up to X'FFF' is valid as *vaddr*; otherwise, any address up to X'5FF' is valid.

System Response

CP sends messages to the following to tell them the ATTACH was successful.

- The virtual machine user
- The operator who issued the command (if different from the user)
- The primary system operator (if different from the operator who issued the command).

```
type raddr ATTACHED
raddr... ATTACHED
raddr-raddr ATTACHED
```

Depending on the form that the user uses on the command line, the response indicates that the real device or devices are attached to the virtual machine. If you specify more than one address or a range of addresses, CP does not display the device type.

```
type raddr ATTACHED TO userid vaddr
type raddr ATTACHED TO SYSTEM void
raddr... ATTACHED TO userid
raddr-raddr ATTACHED TO userid
```

Depending on the form that you use on the command line, the response shows that the real device(s) are attached to CP or a specified user's virtual machine (and, if appropriate, the virtual address of the device). If you specify more than one address or a range of addresses, CP does not display the device type.

```
type raddr ATTACH TO SYSTEM void BY userid
type raddr ATTACH TO userid vaddr BY userid
```

CP sends a response to the primary system operator, if he did not issue the command, to show that CP attached the real device to either:

- The specified user's machine at virtual address *vaddr*
- The CP system by the specified class B user.

```
CHANNEL c [PROC nn] ATTACHED TO userid
CHANNEL c [PROC nn] ATTACH TO userid
CHANNEL c [PROC nn] ATTACH TO userid BY operator
```

CP sends responses to the following to tell them the ATTACH CHANNEL was successful:

- The user whose virtual machine the channel is attached to
- The operator who issued the command
- The primary system operator (if someone else issued the command).

ATTACH

In the preceding responses, type is one of the following:

Type	Meaning
DASD	Direct access storage device
TAPE	Magnetic tape
LINE	Communication line
RDR	Card reader
PRT	Line printer
PUN	Card punch
GRAF	Graphics device
CONS	Console
CTC	Channel-to-channel device
CTLR	3704, 3705, or 3725 Communications Controller
DEV	Any other device

AUTOLOG

Privilege Class: A or B

When to Use AUTOLOG

Use the AUTOLOG command when you want to log on any virtual machine that is defined in the VM/SP directory. The virtual machine that you log on with AUTOLOG:

1. Must have an automatic IPL defined in its directory
2. Can issue one read to its virtual console
3. Operates in disconnected mode.

AUTOLOG Format and Operands

AUTOLOG	<i>userid password [variable data]</i>
---------	--

userid

identifies the virtual machine user that you want to log on.

password

is the password for the virtual machine that you want to log on.

If your installation has generated password suppression, CP assumes all data following *userid* is variable data. CP will prompt you to enter the password at the terminal. The only time CP lets you issue AUTOLOG with the password on the command line is when you are issuing AUTOLOG from an exec or module running under CMS.

If LOGON journaling is:

- OFF, and you issue AUTOLOG with a *password* from a virtual machine, CP checks the password and keeps track of incorrect passwords. If you enter 4 incorrect passwords, CP issues an error message and does not let you issue the AUTOLOG command from the virtual machine. You can still issue AUTOLOG directly from the terminal (that is, in CP console mode).
- ON, CP keeps track of incorrect passwords without regard to the origin of the command. If you enter the installation defined limit of incorrect

AUTOLOG

AUTOLOG passwords (specified in the SYSJRL macro of DMKSYS), CP does not let you issue AUTOLOG, regardless of how you entered it.

variable data

is the information that CP passes to the virtual machine. CP uses this data to satisfy reads to your terminal. You may place up to 108 bytes of data in this field. If you do not enter *variable data*, CP pads the field with blanks.

If you enter the AUTOLOG command to log on a virtual machine that has an IPL for CMS in its directory, the absence of *variable data* satisfies the first read to the virtual console. To prevent CP from forcing off the user because another read is being issued to the virtual console, include 'SET AUTOREAD OFF' in either:

- The variable data field, or
- The AUTOLOGed machine's PROFILE EXEC. (CP treats embedded line-edit characters in this profile exec as data.)

Things You Should Know about the AUTOLOG Command

1. Do not use the AUTOLOG command to log on the Primary System Operator's virtual machine. If you AUTOLOG the primary system operator's virtual machine, it cannot receive any messages (including CP messages) that are intended for the primary system operator.
2. If the VM/SP directory has a userid of AUTOLOG1, CP issues the AUTOLOG command for the AUTOLOG1 virtual machine. You can set up the PROFILE EXEC of the AUTOLOG1 virtual machine to have more AUTOLOG commands to log on other virtual machines (for example, the batch virtual machine).

System Response

```
hh:mm:ss AUTO LOGON *** user id USERS=nnn BY {SYSTEM }  
                                              {user id2 }
```

indicates there is no assigned line address for the virtual console.

user id

identifies the virtual machine that AUTOLOG started.

USERS=nnn

is the number of logged-on users.

SYSTEM

indicates that the AUTOLOG process is part of VM/SP system initialization.

user id2

identifies the virtual machine that invoked the AUTOLOG command.

BACKSPAC

Privilege Class: D

When to Use BACKSPAC

Use the BACKSPAC command to restart or reposition the current output on a real punch or printer. If you issue BACKSPAC when an intervention required condition is pending on the printer or punch, CP terminates the interrupted channel program and does the backspace.

BACKSPAC Format and Operands

Printer BACKSPAC Format

Backspac	<i>raddr</i> [File <i>pages</i> [EOF] <u>1</u>]
-----------------	--

Punch BACKSPAC Format

Backspac	<i>raddr</i> [File]
-----------------	------------------------------

raddr

is the address (cuu) of the real unit record device that you want CP to backspace. *raddr* must be a punch or printer address.

File

restarts the output spool printer or punch file from the beginning.

pages

specifies the number of pages that you want CP to backspace on the printer. If you do not specify a field, the printer will backspace to the beginning of the page that it was printing when it received the command.

CP counts pages in BACKSPAC operations by looking for skip to channel 1 specifications (page ejects). If a file does not have skip to channel 1 specifications, CP restarts processing at the beginning of the file, regardless of the number of pages indicated. Counting begins at the end of the current buffer.

Do not use this operand for a virtual 3800 spool file.

BACKSPAC

EOF

starts printing from the end of the spool file. Backspacing will start the requested number of pages from the end of the spool file. Printing restarts at this location.

Do not use this operand for a virtual 3800 spool file.

Things You Should Know about the BACKSPAC

1. For buffered printers, such as the 3800 printer, the BACKSPAC command does not affect files that have already been sent to the printer.
2. CP counts pages for backspacing, starting from the end of the current spooling buffer. Output that has a small number of lines per page may require a count of more than one to backspace one page.

An average print file requires approximately two spooling buffers for each printed page. However, if you issued a skip-to-channel-1 (carriage control "1") every five lines, the print file might have six printed pages in each spooling buffer.

3. Some spool files may have imbedded forms control buffers (FCBs). CP loads the imbedded FCB, which controls line skipping and spacing, into the real printer. If CP backspaces over an imbedded FCB within a spool file, printed results may be incorrect. This is because the FCB that CP backspaced over is still located in the real printer. If the file had been printed normally, that FCB would not have been loaded at this point in the spool file.

To avoid backspacing over an imbedded FCB within a spool file, use FILEFCB (START option) only for selected printers and for print files on the printers that allow FCBs. Backspace the file to insure that CP handles the FCBs properly.

System Response

```
{PRT} raddr BACKSPACE USERID FILE CLASS RECORDS RECLEFT CPY DIST SEQ
{PUN}          userid file a typ norecs   noleft  nnn distcode seq
```

raddr
is the real device address.

userid
identifies the user whose file was active and CP backspaced.

file
is the unique, system-assigned spoolid number by which VM/SP refers to the file.

- a**
is the spool file class.
- typ**
is the originating device type, PRT, CON, or PUN.
- norecs**
is the total number of logical records in the file.
- noleft**
is the number of records remaining for CP to process.
- nnn**
is the remaining number of copies of the file to be created, where 001 is the last copy.
- distcode**
is the distribution code of the file.
- seq**
is the sequence of the file (printed on the separator page).

CHANGE

CHANGE

Privilege Class: D

When to Use CHANGE

Use the CHANGE command to alter one or more of the external attributes of a closed spool file. (To find the current attributes of a file, issue QUERY Printer, QUERY PUnch, or QUERY Reader).

Before changing an output file, it must be closed but not yet selected for printing or punching. You can change an input (READER) file at any time before CP opens it, that is, before CP issues the first read for the file.

CHANGE Format and Operands

Change	<table border="0"><tr><td style="vertical-align: middle;">$\left[\begin{array}{l} \text{userid} \\ \text{SYSTEM} \\ * \end{array} \right]$</td><td style="vertical-align: middle;">$\left\{ \begin{array}{l} \text{Reader} \\ \text{Printer} \\ \text{PUnch} \end{array} \right\}$</td><td style="vertical-align: middle;">$\left\{ \begin{array}{l} \text{Class } c1 \\ \text{FORM } form1 \\ \text{spoolid} \\ \text{ALL} \end{array} \right\}$</td><td style="vertical-align: middle;">$\left. \begin{array}{l} \text{CHars } name0 \\ \left[\begin{array}{l} \text{[CHars] } name1 \\ \text{[CHars] } name2 \\ \text{[CHars] } name3 \end{array} \right] \supset^1 \\ \text{Class } c2 \\ \text{COpy[*]nnn} \\ \text{DIst } distcode \\ \text{FCB } name^1 \\ \text{FLash } name\ nnn \\ \text{FORM } form2 \\ \left[\begin{array}{l} \text{HOLD} \\ \text{NOHOLD} \end{array} \right] \\ \text{MOdify } name[n]^1 \\ \left[\begin{array}{l} \text{SYS} \\ \text{NOSYS} \end{array} \right] \end{array} \right\}$</td><td style="vertical-align: middle;">$\left[\begin{array}{l} \text{NAME}\{fn\ [ft]\} \\ \{dsname\} \end{array} \right]$</td></tr></table>	$\left[\begin{array}{l} \text{userid} \\ \text{SYSTEM} \\ * \end{array} \right]$	$\left\{ \begin{array}{l} \text{Reader} \\ \text{Printer} \\ \text{PUnch} \end{array} \right\}$	$\left\{ \begin{array}{l} \text{Class } c1 \\ \text{FORM } form1 \\ \text{spoolid} \\ \text{ALL} \end{array} \right\}$	$\left. \begin{array}{l} \text{CHars } name0 \\ \left[\begin{array}{l} \text{[CHars] } name1 \\ \text{[CHars] } name2 \\ \text{[CHars] } name3 \end{array} \right] \supset^1 \\ \text{Class } c2 \\ \text{COpy[*]nnn} \\ \text{DIst } distcode \\ \text{FCB } name^1 \\ \text{FLash } name\ nnn \\ \text{FORM } form2 \\ \left[\begin{array}{l} \text{HOLD} \\ \text{NOHOLD} \end{array} \right] \\ \text{MOdify } name[n]^1 \\ \left[\begin{array}{l} \text{SYS} \\ \text{NOSYS} \end{array} \right] \end{array} \right\}$	$\left[\begin{array}{l} \text{NAME}\{fn\ [ft]\} \\ \{dsname\} \end{array} \right]$
$\left[\begin{array}{l} \text{userid} \\ \text{SYSTEM} \\ * \end{array} \right]$	$\left\{ \begin{array}{l} \text{Reader} \\ \text{Printer} \\ \text{PUnch} \end{array} \right\}$	$\left\{ \begin{array}{l} \text{Class } c1 \\ \text{FORM } form1 \\ \text{spoolid} \\ \text{ALL} \end{array} \right\}$	$\left. \begin{array}{l} \text{CHars } name0 \\ \left[\begin{array}{l} \text{[CHars] } name1 \\ \text{[CHars] } name2 \\ \text{[CHars] } name3 \end{array} \right] \supset^1 \\ \text{Class } c2 \\ \text{COpy[*]nnn} \\ \text{DIst } distcode \\ \text{FCB } name^1 \\ \text{FLash } name\ nnn \\ \text{FORM } form2 \\ \left[\begin{array}{l} \text{HOLD} \\ \text{NOHOLD} \end{array} \right] \\ \text{MOdify } name[n]^1 \\ \left[\begin{array}{l} \text{SYS} \\ \text{NOSYS} \end{array} \right] \end{array} \right\}$	$\left[\begin{array}{l} \text{NAME}\{fn\ [ft]\} \\ \{dsname\} \end{array} \right]$		

¹You must choose one of these options; however, you may specify more than one in any order.

²The CHars, FCB, and MOdify options are valid for only the 3800 printer.

userid

identifies the user whose spool files are to be changed.

SYSTEM

lets you change all spool files in the system regardless of userid.

*******-**

lets you change your own spool files.

Reader**RDR**

changes reader files.

Printer**PRT**

changes printer files.

PUnch**PCH**

changes punch files.

Class *cl*

selects the files to change by class. *cl* is a one-character letter or number.

FORM *form1*

selects the files to change by form number. If you use the ***** option, the form number is the user form. If you specify the *userid*/**SYSTEM** option, the form number is the operator form. *form1* is one to eight characters.

spoolid

is the spoolid number of the file that you want to change. Each file has a unique spoolid.

ALL

changes files regardless of *class* or *form*.

CHars *name0* [[CHars] *name1* [[CHars] *name2*[[CHars] *name3*]]]

names the character arrangement table that CP uses to print the file. The parameters in brackets, [], are optional. *name* may be one to four characters in length with a maximum of four names per command.

Note: You only need to repeat CHars when one of the names that you specify is the same as one of the option names (such as CLASS, DIST, FLASH, etc.) If you do not specify a character arrangement table for a file, CP uses the default table used for the separator page on the selected printer to print the file.¹

You can specify this option for any file, but the change will only take affect if you send the file to a real 3800 printer for printing.

¹ A specification of NULL will nullify any previous setting.

CHANGE

Class *c2*

indicates the new class of the file.

COpy [*] *nnn*

alters the number of copies for the file. The value of *nnn* (number of copies) must be a number from 1 through 255.

When you include * with the COPY parameter, CP transmits the file to the 3800 printer only once. The 3800 performs the duplication internally, one page at a time. (For example, if you specify COPY *3, CP copies the page numbers as 111 222 333 ...). Otherwise, CP transmits the file once for each copy that you want, with VM/SP indicating the copy number to the 3800.

DIst *distcode*

changes the distribution code of the file.

distcode is a one to eight character alphanumeric identification. It appears on the output separator of printer or punch output. CP changes the distribution code for this file only; CP does not affect other files or change the normal user code.

FCB *name*

names the forms control buffer that CP uses to print the file.

name may be one to four characters, or the unique numbers 6, 8, or 12. For the 3800 Model 3 device, this unique number may also be 10. These numbers tell VM/SP to print the entire spool file at 6, 8, 10 or 12 lines per inch, regardless of the size of the paper that is currently loaded. If you use the unique numbers, you will not cause an FCB-forms mismatch. If you do not specify this operand, CP uses the value that you specified for the separator page in the START command.²

You can specify this option for any file, but the change will only affect the file if you send it to a real 3800 printer for printing.

FLash *name nnn*

specifies the flashing status of the file.

name identifies a forms overlay sheet in the 3800 printer and may be one to four characters. The number *nnn* specifies the number of copies you want CP to flash. This flash count may range from 00 to 255. The flash count defaults to zero and indicates that you do not want flashing.

You can specify this option for any file, but the change will only affect the file if you send it to a real 3800 printer for printing.

Note: The FLASH *name* must match the FLASH *name* that you specify in the START command. The exception is if the flash count is zero.

² A specification of NULL will nullify any previous setting.

FORM *form2*

changes the spool form number of the file to *form2*. If you use the * option, CP changes your form number. If you specify the `userid/SYSTEM` option, CP changes the operator form number.

HOLD

prevents CP from printing, punching, or reading a file. Instead, CP puts the file in a user HOLD status, until the NOHOLD operand releases it.

NOHOLD

releases the specified spool file that the user HOLD status is currently holding.

MOdify *name[n]*

is the name of the copy modification module that CP uses to print the file.

name may be one to four characters in length. If you do not specify *name*, CP will not use copy modification to print the file.¹ *n* selects one of the character arrangement tables specified with the CHars keyword to use for the copy modification text. The values 0, 1, 2, or 3 correspond to the order that you have specified the table names. If you do not specify *n*, CP assumes the first character arrangement table.

You can specify this option for any file, but the change will only affect the file if you send it to a real 3800 printer for printing.

SYS

prevents CP from printing or punching the file. Instead, CP puts the file in system hold status until NOSYS releases it.

NOSYS

releases the specified spool file that the system hold status is currently holding.

Note: A system hold has no effect on the processing of reader files.

NAme $\left\{ \begin{array}{l} fn [ft] \\ dsname \end{array} \right\}$

changes the name of the file. For CMS files, the filename, *fn*, that you specify (and, optionally, filetype, *ft*) becomes the new file identification. For a non-CMS file, the *dsname* that you specify (a 1 to 24-character field) becomes the new file identification.

Note: If you select the NAme function and only specify *fn*, CP sets *ft* to blanks. The *dsname* specification lets a user specify a file name suitable for OS, such as SYS1.SYSLIB.MYMAC.

CHANGE

System Response

{ nnnn } FILES CHANGED
{ NO }

is the response to the user who issued the CHANGE command.

If you issue SET IMSG OFF, you will not see the message. If you are the system operator monitoring the RSCS virtual machine for your installation, do not specify this option. You need to see information messages to accurately monitor RSCS system activity.

Note: Depending on the options you choose and the combinations you use, CP may not have actually changed some files because they had the new class before you issued the CHANGE command. The response indicates how many files you referenced.

CPTRAP

Privilege Class: C

When to Use CPTRAP

Use the CPTRAP command when you want to collect information in a reader file to use for problem determination. The information can include CP trace table entries, CP data, and virtual machine data.

CPTRAP Format and Operands

CPTrap	<i>typenum</i>	<table border="1" style="border-collapse: collapse;"> <tr> <td style="padding: 2px;">Vmblok</td> <td style="padding: 2px;"><i>address</i></td> </tr> <tr> <td style="padding: 2px;">DEVaddr</td> <td style="padding: 2px;"><i>cuu</i></td> </tr> <tr> <td style="padding: 2px;">COde</td> <td style="padding: 2px;"><i>code-value</i></td> </tr> <tr> <td style="padding: 2px;">OFF</td> <td></td> </tr> </table>	Vmblok	<i>address</i>	DEVaddr	<i>cuu</i>	COde	<i>code-value</i>	OFF				
	Vmblok	<i>address</i>											
	DEVaddr	<i>cuu</i>											
	COde	<i>code-value</i>											
	OFF												
	ALL	<table border="1" style="border-collapse: collapse;"> <tr> <td style="padding: 2px;">{ ON }</td> </tr> <tr> <td style="padding: 2px;">{ OFF }</td> </tr> </table>	{ ON }	{ OFF }									
	{ ON }												
{ OFF }													
ALLOWid	<i>userid</i>												
GRoupid	<i>group-name</i>												
STArt	<table border="1" style="border-collapse: collapse;"> <tr> <td style="padding: 2px;">TO</td> <td style="padding: 2px;">{ <i>userid</i> }</td> <td style="padding: 2px;">WRAP</td> <td style="padding: 2px;"><i>wrap-size</i></td> </tr> <tr> <td></td> <td style="padding: 2px;">{ * }</td> <td></td> <td></td> </tr> <tr> <td></td> <td style="padding: 2px;">{ - }</td> <td></td> <td></td> </tr> </table>	TO	{ <i>userid</i> }	WRAP	<i>wrap-size</i>		{ * }				{ - }		
TO	{ <i>userid</i> }	WRAP	<i>wrap-size</i>										
	{ * }												
	{ - }												
CLOSE	<table border="1" style="border-collapse: collapse;"> <tr> <td style="padding: 2px;">{ * }</td> </tr> <tr> <td style="padding: 2px;">{ - }</td> </tr> </table>	{ * }	{ - }										
{ * }													
{ - }													
STOP													

Note: Enter the CPTRAP subcommands on separate command lines. You can, however, put more than one *typenum* subcommand on the same command line.

typenum

Vmblock	<i>address</i>
DEVaddr	<i>cuu</i>
COde	<i>code-value</i>
OFF	

Use *typenum* to identify the type of entry that you want CP to collect in the CPTRAP reader file. You can specify any hexadecimal number from X'01' up to X'3F':

- X'01'-X'1B' - Corresponds to the CP trace table entry
- X'3D' - Group virtual machine entry
- X'3E' - General virtual machine entry

- X'3F' - CP entry.

See *VM/SP System Programmer's Guide* for an explanation of these entries and how to create them.

You can specify *typenum* before or after CPTRAP START. If you do not use the *typenum* subcommand or ALL ON (to specify all typenums) before you start CPTRAP, CPTRAP does not collect any entries in the reader file. If ALL ON is in effect when you enter a *typenum* subcommand, CP sets all other typenums off. You cannot specify *typenum* on the same command line with other subcommands.

Example:

CPTRAP 5 6
will collect trace table types X'05' and X'06' in the CPTRAP file along with any previous definitions.

There are three options for *typenum* that give you additional selectivity. Figure 3-2 on page 3-28 shows which options you can use for each CP trace table entry. All the options are not valid for each *typenum*.

Vmblok address

lets you select CP trace table entries based on a VMBLOK address. This option is only valid for trace table types that have a VMBLOK field. The *address* of the VMBLOK can be up to six digits.

Examples:

CPTRAP 6 collects all CP trace table entries for FREE STORAGE (X'06').

CPTRAP 6 VMBLOK 41D80
collects only CP trace table entries for FREE STORAGE (X'06') that the system did for a user with VMBLOK address 41D80.

DEVaddr cuu

lets you select CP trace table entries based on device address. This option is valid for CP trace table entries that have a real or virtual device address field. *cuu* is the device address and can be up to 4 digits.

Examples:

CPTRAP 0B
collects all CP trace table entries for START I/O (X'0B').

CPTRAP 0B DEV 250
collects only CP trace table entries for START I/O (X'0B') that the system did for device 250.

CODE *code-value*

lets you select CP trace table entries based on a specific code field in the trace entry. This option is valid for CP trace table entries that have one of the following types of code fields:

- Interrupt code
- Condition code
- Order code
- Function code
- Transaction type.

The *code-value* can have up to four digits, except CP trace type 4, which can have up to 8 digits.

Examples:

CPTRAP 2 collects all CP trace table entries for SVC interrupts (X'02').

CPTRAP 2 CODE 8
collects only CP trace table entries for SVC 8.

CPTRAP 2 CODE 8 2 CODE C
collects only CP trace table entries for SVC 8 and SVC 12.

OFF

deletes selectivity for a specified typenum.

Examples:

CPTRAP 2 OFF
does not collect X'02' CP trace table entries.

CPTRAP 2 OFF 3E OFF
does not collect X'02' CP trace table entries or X'3E' general virtual machine entries

To use more than one typenum option, assign *typenum* with each option you choose.

Example:

CPTRAP 5 DEV 190 18 DEV 190 18 DEV 192 8 VMBLOK 2D4900
collects

- Type 5 with device 190
- Type 18 with device 190
- Type 18 with device 192
- Type 8 with VMBLOK address 2D4900.

TRACE TABLE		TYPENUM OPTIONS		
EVENT TYPE	(typenum) EVENT CODE	Vmblok	DEVaddr	COde
External Interrupt	X'01'			X ³
SVC Interrupt	X'02'			X ³
Program Interrupt	X'03'			X ³
Machine Check Interrupt	X'04'	X		X ³
I/O Interrupt	X'05'		X	
Free Storage (FREE)	X'06'	X		
Return Storage (FRET)	X'07'	X		
Enter Scheduler	X'08'	X		
Queue Drop	X'09'	X		
Run User	X'0A'	X ¹		
Start I/O	X'0B'		X	X ⁴
Unstack I/O Interrupt	X'0C'	X	X	
Virtual CSW Store	X'0D'	X	X	
Test I/O	X'0E'		X	X ⁴
Halt Device	X'0F'		X	X ⁴
Unstack IOBLOK/TRQBLOK	X'10'	X		
NCP BTU	X'11'			
Spinning on Lock	X'12'	X		
SIGP Issued	X'13'		X ²	X ⁵
Clear Channel Issued	X'14'	X	X	
IUCV Communication	X'15'			X ⁶
SNA CCS	X'16'			X ⁷
Diagnose X'80'	X'17'			X ⁴
Start I/O Fast Release	X'18'		X	X ⁴
Simulated I/O Interrupt	X'19'		X	
Clear I/O	X'1B'		X	X ⁴

The following superscripts describe the fields in the CP trace table entry:

- 1 refers to RUNUSER Value.
- 2 refers to Real Processor Address.
- 3 refers to Interrupt Code.
- 4 refers to Condition Code.
- 5 refers to Order Code.
- 6 refers to Function Code.
- 7 refers to Transaction Type.

Figure 3-2. CP Trace Event Type Codes

ALL **[ON
OFF]**

ON

turns on the selection of all typenums. This is the default.

OFF

turns off the selection of all typenums.

ALLOWid *userid*

enables an individual virtual machine to send data to add to the CPTRAP file. The virtual machine must currently be logged on or disconnected. The virtual machine can enter data into the CPTRAP file until:

- The virtual machine logs off, or
- You use CPTRAP STOP to stop CPTRAP.

GRoupid *group-name*

enables all the virtual machines in the machine group that you specify, which are logged on or disconnected, to send virtual machine data to the CPTRAP file. Also, any virtual machine that enters the group after you have issued CPTRAP GROUPID is also enabled to enter data into the CPTRAP file. The virtual machine can enter data into the CPTRAP file until:

- The virtual machine logs off, or
- You use CPTRAP STOP to stop CPTRAP.

If all virtual machines in a group leave the group, issue a new GROUPID subcommand when the group re-establishes itself. This will cause CP to automatically enable any virtual machine when it enters the group.

START **[TO] {userid} [WRAP wrap-size]**

starts CPTRAP and defines how many CPTRAP records you want the SPOOL file to maintain. The system will not collect data in the CPTRAP file until it receives the START subcommand.

[TO] *userid*
*

identifies the owner of the CPTRAP file. If you use the TO operand, follow it with a *userid* or *. If you omit the TO operand you cannot have a *userid* "to" or "wrap". If you do not specify *userid* at start time, the command will default to *. This means that the virtual machine that issued CPTRAP START will own the CPTRAP file.

WRAP *wrap-size*

starts a wrap type SPOOL file. A wrap file reuses SPOOL space. *wrap-size*, a decimal number from 16 to 20000, is the number of CPTRAP records (4K blocks of data) that you want the SPOOL file to maintain. After the SPOOL file collects the number of records you specify with *wrap-size*, the newer CPTRAP records overlay the old records.

CPTRAP

You cannot change the size of the wrap file during the CPTRAP session. You can issue CPTRAP CLOSE to close the current wrap file and open a new wrap file. However, the original *wrap-size* that you specified on the CPTRAP START command will be in effect for the new file also. If you want to change the size of the wrap file, end the current CPTRAP session with CPTRAP STOP and start a new session.

If you do not specify the WRAP option, the system will start a non-wrap SPOOL file. A non-wrap SPOOL file does not reuse SPOOL space; so, the space for this file is limited by the SPOOL space available on the system. The newer CPTRAP records do not overlay the old records in this file; when the file reaches 3480 4K blocks of data, the system closes the file and opens a new file.

CLOSE

closes the current file and creates a reader file that includes the data collected up to this point. CPTRAP opens a new file so you can continue to collect data. If you issue CLOSE and there is no data in the file, CPTRAP will not create a file.

STOP

stops CPTRAP and creates a reader file if there is data. When you issue CPTRAP STOP, the system stops collecting data, discards all selectivity definitions, and completely stops the CPTRAP operation. If you issue STOP and there is no data in the file, CPTRAP will not create a file. STOP is valid when the CPTRAP facility is active or inactive.

Things You Should Know about the CPTRAP Command

1. Only one user can invoke CPTRAP at any one time. If you try to invoke CPTRAP when another user has invoked it, you will get a message saying CPTRAP is already active.
2. You can display or print the data that CPTRAP collects with the TRAPRED command. Refer to *VM/SP System Programmer's Guide* to learn how to use this command.

DCP

Privilege Classes: C or E

When to Use DCP

Use the DCP command to display the contents of real storage locations at your terminal. In a system running with an attached processor, you can also disclose the values of PSA for the main and the attached processor.

DCP Format and Operands

DCP	$\left\{ \begin{array}{l} \left[\begin{array}{l} \mathbf{MLhexloc1} \\ \mathbf{NLhexloc1} \\ \mathbf{MThexloc1} \\ \mathbf{NThexloc1} \\ \mathbf{Mhexloc1} \\ \mathbf{Nhexloc1} \\ \mathbf{Lhexloc1} \\ \mathbf{Thexloc1} \\ \mathbf{hexloc1} \\ \mathbf{L} \\ \mathbf{T} \\ \mathbf{0} \end{array} \right] \left[\begin{array}{l} \left\{ \begin{array}{l} \mathbf{-} \\ \mathbf{:} \end{array} \right\} \left[\begin{array}{l} \mathbf{hexloc2} \\ \mathbf{END} \end{array} \right] \\ \left\{ \begin{array}{l} \mathbf{.} \end{array} \right\} \left[\begin{array}{l} \mathbf{bytecount} \\ \mathbf{END} \end{array} \right] \end{array} \right. \end{array} \right\}$
-----	---

MLhexloc1
NLhexloc1
MThexloc1
NThexloc1
Mhexloc1
Nhexloc1
Lhexloc1
Thexloc1
hexloc1

L
T
0

specifies the first storage location that CP will display. If you specify *hexloc1*, CP displays, in hexadecimal, only this storage location. When you specify L or T (the designated type character), the display will start with storage location 0.

- T includes an EBCDIC translation with the hexadecimal display.
- L displays in hexadecimal only.

If *hexloc1* is followed by a period and is not on a fullword boundary, CP rounds it down to the next lower fullword. Note that *hexloc1* and *Lhexloc1* get the same results.

In attached processor (AP) systems, M causes CP to interpret addresses like the main processor generated them. Thus, M0 refers to location 0 of the PSA of the main processor. In multiprocessor (MP) systems, M corresponds to addresses on the IPL processor.

In attached processor (AP) systems, N causes CP to interpret addresses like the attached processor generated them. In multiprocessor (MP) systems, N causes CP to interpret addresses as if they were on the non-IPL processor. N is valid only if the attached processor is operational.

Notes:

1. When you do not specify M or N, CP interprets the address as an absolute address on both AP and MP systems.
2. The M and N prefixes are not meaningful for uniprocessor systems.
3. In AP applications, the PSA values for the main and attached processors are not located at absolute page zero but are displaced from it. Therefore, if you prefix the hexadecimal location with an M or N, the system will add the hexadecimal value to the displacement.

Example:

dcp m40.8
Displays the CSW of the main processor.

dcp n78-7f
Displays the I/O new PSW of the attached processor.

{ - } [*hexloc2*]
{ : } [END]

displays a range of locations. Use the colon (:) or dash (-) to display the contents of one or more storage locations by addresses that you specify.

The *hexloc2* operand must be 1 to 6 hexadecimal digits and greater than or equal to *hexloc1*. *hexloc2* should not exceed the size of real storage. If you specify END, and a colon (:) or dash (-) follows the first operand, CP will display real storage from *hexloc1* through the end of real storage. In other words, the following commands are identical:

- DCP O:END
- DCP O:
- DCP O- -END

- DCP O-

{.} [*bytecount*
END]

bytecount is a hexadecimal number that represents the number of bytes of real storage (starting with the byte at *hexloc1*) that you want CP to display. The *bytecount* operand must be 1 or greater and may not exceed six hexadecimal digits.

The sum of *hexloc1* and the *bytecount* must be an address that does not exceed the size of real storage. If this address is not on a fullword boundary, CP rounds it up to the next higher fullword.

Things You Should Know about the DCP Command

1. Normally, you'll define the beginning and ending locations of storage as follows:

```
dcp Lhexloc1-hexloc2
dcp Thexloc1-hexloc2
dcp hexloc1:hexloc2
dcp hexloc1.bytecount
dcp hexloc1:hexloc2 hexloc1.bytecount
```

Do not enter blanks between the limit or range symbols (: or - or .), or between any of the operands. The exception to this is the blank(s) between the command name and the first operand. Also, you'll need to enter a blank between each set of operands when you enter more than one set of operands on one command line.

If you put a blank immediately after the designated type character (T or L), DCP displays all of real storage. If the next operand is a colon (:), a dash (-), or a period (.) followed by a blank, CP displays all storage locations. This is because CP assumes a second set of operands. Do not put a blank on the right or left of range or length symbols, unless you want the missing operand to default.

2. All of the following produce full storage displays:

dcp l	dcp l-	dcp 0-	dcp t:end
dcp t	dcp l:	dcp 0:	dcp t:end
dcp -	dcp t:	dcp 1-end	dcp 0:end
dcp :	dcp l.	dcp t-end	dcp 1.end
dcp .	dcp t.	dcp 0-end	dcp 0.end

Each of the following produce three full dumps because of the imbedded blanks:

```
dcp l . t
```

If you want CP to apply default values (that is, the beginning and ending of storage), specify one operand.

System Response

If you enter an invalid operand, the DCP command stops. However, CP processes any previous valid operands before it terminates the command.

```
xxxxxx = word1 word2 word3 word4 [key] *EBCDIC translation*
```

CP displays locations that you request.

xxxxxx

is the real storage location of word1.

word1

is displayed (word-aligned) for a single hexadecimal specification. CP displays up to four words on a line. If required, CP will display more than one line.

EBCDIC translation

is displayed aligned to the next lower 16-byte boundary if you specify Thexloc. Note that CP displays nonprintable characters as a period (.).

If the location is at a 2K page boundary, CP also displays the key for that page. Press the ATTN key (or its equivalent) to stop the output and terminate the command.

DEFINE

Privilege Class: A or B

When to Use DEFINE

Use the DEFINE command to specify the status of a 3330V volume.

For example, a 3330V may be defined as VIRTUAL so you can dedicate it to a specific virtual machine. If you want the 3330V volume to be available to the control program for mounting system volumes, you can define it as SYSVIRT.

DEFINE Format and Operands

DEFine	$\left\{ \begin{array}{l} \text{Sysvirt} \\ \text{Virtual} \end{array} \right\} \text{ raddr1 [-raddr2]}$
---------------	---

Sysvirt

reserves the 3330V for CP. You can mount VM/SP system volumes that are MSS 3330V volumes on SYSVIRT 3330V devices. You cannot dedicate these volumes to a virtual machine by address, and you cannot attach them to anything other than the system.

Virtual

allows the 3330V device to be dedicated or attached to a virtual machine.

raddr1 [-raddr2]

specifies the device address or range of addresses you want CP to define. If you specify a range, CP sets all 3330V devices in the range to the status that you specify.

Things You Should Know about the DEFINE Command

CP demounts an existing MSS volume before completing the DEFINE command when:

- An MSS volume is already mounted.
- The MSS volume is not being used on the 3330V that is being defined.

DEFINE

System Response

CP sends responses to you when you issue the DEFINE command. The responses show the device status after CP executes the command. Possible responses are:

raddr NOT KNOWN

The raddr is not a valid device address.

raddr NOT 3330V

The raddr is not a 3330V. CP does nothing for the raddr.

raddr ALREADY SYSVIRT

raddr ALREADY VIRTUAL

The raddr already has the status that you tried to define.

DEVICE raddr IS NOT AVAILABLE

The raddr is currently dedicated to a virtual machine. CP does not change the status from VIRTUAL to SYSVIRT.

raddr IS OFFLINE

The raddr is currently offline. When you vary it online, the raddr will receive the status that you specified in the command.

raddr IN USE BY SYSTEM

The raddr is currently attached to the system and there is at least one user minidisk allocated on the volume. Do not define the raddr as VIRTUAL.

raddr HAS A MOUNTED VOL.

MSS ID NOT AVAILABLE TO DEMOUNT IT

There is a 3330V volume mounted on the raddr. The MSS communicator is not active and the CP cannot demount the volume. CP does not change the status of the raddr.

raddr - ERROR DEMOUNTING VOLUME

There is a 3330V volume mounted on the raddr. CP tried to demount the volume, but MSS encountered an error. CP does not demount the volume and does not change the status of the raddr.

INVALID RANGE - raddr

You specified a range of address, but either raddr1 or raddr2 does not exist or is not a 3330V. CP does not change the status.

raddr DEFINED AS VIRTUAL

raddr DEFINED AS SYSVIRT

CP has given the status that you specified to the raddr. If a volume was previously mounted on the raddr, CP has now demounted it.

DETACH

Privilege Class: B

When to Use DETACH

Use the DETACH command to remove a real device from the CP system. You can detach a previously attached device from a user, even if the device is still in use. When you detach a tape device, CP automatically rewinds and unloads it.

Use DETACH CHANNEL to detach a dedicated channel from a user when the virtual machine operations do not need the channel path any longer.

DETACH Format and Operands

DETach	$\left\{ \begin{array}{l} raddr \\ raddr... \\ raddr-raddr \end{array} \right\} \quad [\text{FROM}] \quad \left\{ \begin{array}{l} userid \\ \text{SYSTEM} \\ * \end{array} \right\}$ $\text{CHannel } c \text{ [PROC } nn] \text{ [FROM]} \quad \left\{ \begin{array}{l} userid \\ * \end{array} \right\}$
---------------	--

$$\left\{ \begin{array}{l} raddr \\ raddr... \\ raddr-raddr \end{array} \right\}$$

specifies the real address (cuu), multiple addresses, or range of addresses of the device or devices that you want to detach.

You can specify up to 48 multiple addresses on the DETACH command line. If you try to detach more than 48 addresses, the system will:

- Only process the first 48 addresses
- Ignore the remaining addresses and will not issue an error message for these device addresses.

You may specify more than 48 addresses, however, by specifying the addresses in a range.

$$\text{FROM} \left\{ \begin{array}{l} userid \\ \text{SYSTEM} \\ * \end{array} \right\}$$

identifies the virtual machine user to which the the real device [*raddr*] is currently attached.

DETACH

If you omit the optional keyword FROM, do not specify *userid* as from, fro, fr or f. When you specify a *userid* that could also be a valid hexadecimal device address (for example, virtual machine id CE), use the FROM operand to distinguish it as *userid*.

SYSTEM shows that the device is a DASD device, which CP is using for allocation and control. If you want CP to detach a device from your own virtual machine (class B operator), use an asterisk [*].

CHANnel *c*

is the real address of the channel that you want to detach. *c* may be any hexadecimal character from 1 through 9 or A through F. This operand works only with real channels whose devices have no explicitly defined alternate paths.

If you issue DETACH CHANnel before normal I/O activities on the dedicated channel path are complete, CP terminates those activities.

Note: If you issue an ATTACH CHANNEL command right after a DETACH CHANNEL command, you may get a message that the specified channel is in use. DETACH CHANNEL processing schedules I/O at different times for each dedicated device; but, DETACH CHANNEL processing does not wait for all I/O to finish before it returns control to the terminal operator. Because of the timing, the DETACH CHANNEL command may fail.

PROC *nn*

is the address of the processor that owns the channel. CP does not require this operand in uniprocessor or attached processor mode. If you do specify PROC, CP will check its validity.

FROM {*userid*
* }

identifies the virtual machine user from whom CP will detach the channel. If you omit the optional keyword FROM, do not specify *userid* as from, fro, fr, or f. If you want CP to detach a channel from your own virtual machine (class B operator), use the asterisk (*).

Things You Should Know about the DETACH Command

1. If the device you want to detach is:
 - A 370x, VM/SP cannot use it as a 370x unless the system programmer loaded the 370x by VM/SP, DOS/VIS, OS/VIS, or ACF/SSP with the 270x EP program.

The DETACH and ATTACH commands reset the loaded status.

- A 2305, specify both the real device address and the virtual device address as the first exposure on the 2305 (that is, device address 0 or 8). When

CP attaches or detaches a dedicated 2305 to a user, CP processes all eight exposures.

- An active shared device from the system, you must inform affected users to detach the device from their virtual systems. You can use the QUERY SYSTEM raddr command to determine when all users have detached the device from their virtual machines.
 - A device that has minidisks in use or a device with volumes in the SYSOWN list, CP cannot detach them.
2. If your installation is using DOS or OS virtual machine systems, vary the device offline before invoking the DETACH command. Similarly, if you are a CMS user, issue the RELEASE command before invoking DETACH.
 3. If you use DETACH while the virtual machine is tracing I/O, it will cause abend DSP004.

System Response

CP sends messages to the following to tell them the DETACH was successful.

- The virtual machine user
- The operator who issued the command (if different from the user)
- The primary system operator (if different from the operator who issued the command).

```
{ vaddr... } DETACH BY operator
{ vaddr-vaddr }
{ type vaddr }
```

is the response sent to the user if you detach one or more of the user's devices.

```
{ raddr... } DETACHED [user id]
{ raddr-raddr } [SYSTEM]
{ type raddr }
```

is the response if the user detaches a previously attached device(s) or if you detach the device(s) from a user or the system.

```
{ raddr... } DETACHED [user id] BY operator
{ raddr-raddr } [SYSTEM]
{ type raddr }
```

is the response issued to the primary system operator if a class B operator other than the primary system operator issued the DETACH command, and the device or devices had been previously attached.

DETACH

Note: In the preceding responses, the term “type” is one of the following:

Type	Meaning
DASD	Direct access storage device/3330V (MSS 3330 virtual volume)
TAPE	Magnetic tape
LINE	Communication line
RDR	Card reader
PRT	Line printer
PUN	Card punch
GRAF	Graphics device
CONS	Console
CTC	Channel-to-channel device
CTLR	3704, 3705 or 3725 communications controller
DEV	Any other device
MSC	3151 port address

CTC vaddr DROP FROM userid vaddr
is the response if the device detached was a virtual CTC connected (by the COUPLE command) to another CTC on the virtual machine with the userid you specified. This response is always followed by the response:

CTC vaddr DETACHED

CHANNEL c [PROC nn] DETACHED BY operator
is sent to the user from whose virtual machine the channel is being detached.

CHANNEL c [PROC nn] DETACHED userid BY operator
is sent to the primary system operator if he did not issue the command.

DISABLE

Privilege Class: A or B

When to Use DISABLE

Use the DISABLE command to prevent the following from accessing the VM/SP system:

- Low-speed communication lines, including 2701, 2702, and 2703 lines (collectively referred to as 270x lines)
- EP (Emulator Program) controlled lines in the 370x
- Locally attached 3270 lines.

If the line that you select is not active (not dialed into or logged on), CP disables the line immediately. If the line that you select is active, CP disables it when the user logs off or when the user issues a DISCONN command without specifying the HOLD operand.

DISABLE Format and Operands

DISAble	$\left\{ \begin{array}{l} raddr... \\ \text{SNA } [userid] \\ \text{ALL} \end{array} \right\}$
---------	--

raddr...

are the addresses (cuu) of the communication lines that you want to disable.

The address that you give may represent up to 2700 line addresses. CP checks these three-digit hexadecimal addresses against the line addresses stored in the system table. If CP does not find a match, CP sends an error message to the system console.

SNA [userid]

disables the SNA logical units for the VTAM Service Machine (VSM) that you specify. If you do not specify *userid*, CP disables all SNA units for all VSMs. See the VM/VCNA publications and ACF/VTAM publications listed in the Preface for information on using SNA terminals.

DISABLE

ALL

disables the following:

- 270x communication lines
- EP (Emulator Program) controlled lines in the 370x
- All SNA (Systems Network Architecture) terminals
- Locally attached 3270 lines, including the operator's line.

Note: To examine the status of the communication lines, issue a QUERY LINES and NETWORK QUERY command.

Things You Should Know about the DISABLE Command

1. Use the NETWORK DISABLE command instead of DISABLE when:
 - You want to disable a remote 3270 configuration
 - You want to disable remote 370x resources.
2. If you want to negate the DISABLE command, issue an ENABLE command to the line before you physically disable the line.

System Response

COMMAND COMPLETE

DMCP

Privilege Classes: C or E

When to Use DMCP

Use the DMCP command to print the contents of real storage locations on the user's virtual spooled printer. The output format is eight words per line with EBCDIC translation.

In a system running with an attached processor, you can also dump values that are displacements from the assigned PSA (Prefix Storage Area) locations of the main or attached processor. If you want the output to print on the real printer, use the CLOSE command to stop the virtual spooled printer.

DMCP Format and Operands

DCP	$\left(\begin{array}{l} \left[\begin{array}{l} Mhexloc1 \\ Nhexloc1 \\ Mhexloc1 \\ Nhexloc1 \\ Mhexloc1 \\ Nhexloc1 \\ Lhexloc1 \\ Thexloc1 \\ hexloc1 \\ 0 \end{array} \right] \left[\begin{array}{l} \{-\} \left[\begin{array}{l} hexloc2 \\ \underline{END} \end{array} \right] \\ \{.\} \left[\begin{array}{l} bytcount \\ \underline{END} \end{array} \right] \end{array} \right] \end{array} \right) [*dumpid]$
------------	--

Note: If you want CP to apply default operand values (the beginning and ending of storage), specify at least one operand.

MLhexloc1
NLhexloc1
MThexloc1
NThexloc1
Mhexloc1
Nhexloc1
Lhexloc1
Thexloc1
hexloc1

0

specifies the first storage location that CP will dump. If you specify only *hexloc1*, CP dumps the specified storage location. If you follow *hexloc1* with a period and it is not on a fullword boundary, CP rounds it down. When you specify L or T, the dump will start with storage location 0 and end at the location that you specify.

- T includes an EBCDIC translation with the hexadecimal display.
- L displays in hexadecimal only.

In attached processor (AP) systems, M causes CP to interpret addresses like the main processor generated them. Thus, M0 refers to location 0 of the PSA of the main processor. In multiprocessor (MP) systems, M corresponds to addresses on the IPL processor. M is valid if the system has been generated for an attached processor.

In attached processor (AP) systems, N causes CP to interpret addresses like the attached processor generated them. In multiprocessor (MP) systems, N causes CP to interpret addresses as if they were on the non-IPL processor. N is valid only if the attached processor is operational.

Notes:

1. When you do not specify M or N, CP interprets the address as an absolute address on both AP and MP systems.
2. The M and N prefixes are not meaningful for uniprocessor systems.
3. In AP applications, the PSA values for the main and attached processors are not located at absolute page zero but are displaced from it. Therefore, if you prefix the hexadecimal location with an M or N, the system will add the hexadecimal value to the displacement.

Example:

dmcp m40.8

Dumps the CSW of the main processor

dmcp n78-7f

Dumps the I/O new PSW of the attached processor.

{-} [*hexloc2*]
{:} [END]

dumps a range of real storage locations. To dump to the end of real storage either:

- Specify *hexloc2* as END, or
- Do not specify *hexloc2* at all, in which case CP assumes END by default.

When you specify more than one storage location on one command line and the line has errors, CP processes all correct operands up to the error. CP rejects the error and the rest of the command line.

{.} [*bytecount*]
[END]

bytecount is a hexadecimal number that represents the number of bytes of real storage (starting with the byte at *hexloc1*) that you want the printer to print. The sum of *hexloc1* and the *bytecount* must be an address that does not exceed the size of real storage. If this address is not on a fullword boundary, CP rounds it up to the next higher fullword.

**dumpid*

helps you identify dumps. CP prints the *dumpid* right before the dump data. You can specify up to 100 characters with or without blanks after the asterisk prefix. You must specify *hexloc2* or *bytecount* when you specify this operand. You need the asterisk (*) to identify the *dumpid*.

Things You Should Know about the DMCP Command

1. Normally, you'll define the beginning and ending dump locations as follows:

```
dmcp Lhexloc-hexloc
or
dmcp hexloc.bytecount
```

Do not enter blanks between the limit or range symbols (: or - or .), or between any of the operands. The exception to this is the blank(s) between the command name and the first operand.

If you put a blank immediately after the designated type character (T or L), CP defaults to the beginning and end of virtual storage. Similarly, if you follow the range or length symbol with a blank or END, CP dumps all real storage.

2. If you enter more than one operand on the same command line, separate operands or sets of operands with blanks. Do not put a blank on the right or left of range or length symbols, unless you want the missing operand to default.

DMCP

All of the following produce full storage dumps:

dmcp 1	dmcp 1-	dmcp t.	dmcp t-end
dmcp t	dmcp t-	dmcp 0-	dmcp 0:end
dmcp -	dmcp 1:	dmcp 0:	dmcp 1.end
dmcp :	dmcp t:	dmcp 0.	dmcp 0.end
dmcp .	dmcp 1.	dmcp 1-end	

Each of the following produce three full dumps because of the imbedded blanks:

```
dmcp l . t
dmcp - : .
```

System Response

DUMPING LOC hexloc

CP sends this as the dump proceeds. This message indicates that the dump is continuing from the next 64K boundary. If the user signals attention on the terminal while CP is displaying this message, the dump ends.

hexloc is the segment (64K) address for the dump continuation, such as 020000, 030000, 040000.

COMMAND COMPLETE

indicates normal completion of the dump.

DRAIN

Privilege Class: D

When to Use DRAIN

Use the DRAIN command to stop spooling operations on a real unit record device. With DRAIN, you can:

- Bring the spooling system or a specified device to a controlled halt.
- Halt the activities on a device whose spooling status you want to change.

For example, you usually drain all unit record devices before system shutdown; you must drain a printer before you change the contents of the UCS printer buffer.

CP immediately drains an inactive device when you issue DRAIN. If the device is active, CP drains the device when it finishes processing the current file.

DRAIN Format and Operands

DRain	<div style="border-left: 1px solid black; border-right: 1px solid black; padding: 0 10px;"> Reader Printer PUnch <i>raddr...</i> <u>ALL</u> </div>
-------	--

Reader
RDR

drains all of the system readers.

Printer
PRT

drains all of the system printers.

PUnch
PCH

drains all of the system punches.

raddr

drains the real spooled devices at the addresses (cuu) that you specify. You can specify more than one address.

DRAIN

ALL

drains every spooled unit record device in the system.

Things You Should Know about the DRAIN Command

1. For buffered printers, such as the 3800 printer, the BACKSPAC command does not affect files that have already been sent to the printer.
2. Restart a drained device with the START command. If you issue START to a device before draining is complete, the device does not enter the drained status but continues processing.

System Response

The following is an example of a response when the DRAIN is successful. For more information about the variables, see the QUERY Unit Record description.

```
PRT 001 DRAINED      CLASS A      FORM STANDARD
   001 MANUAL SEP    NO3800      FILEFCB
```

ENABLE

Privilege Class: A or B

When to Use ENABLE

Use the ENABLE command to enable the following previously disabled or nonenabled devices so users may access the VM/SP system:

- 3270 (locally attached only) and the 3138, 3148, and 3158 consoles. (Use the NETWORK ENABLE command for 3270 remote devices.)
- 2701, 2702, and 2703 (270x)
- EP (Emulator Program) controlled lines in the 370x
- Systems Network Architecture consoles.

ENABLE does not affect previously enabled lines.

ENABLE Format and Operands

ENable	$\left\{ \begin{array}{l} raddr... \\ \text{SNA } [userid] \\ \text{ALL} \end{array} \right\}$
--------	--

raddr...

are the addresses [cuu] of the lines that you want CP to enable. If you specify more than one address, separate each address by one or more blanks.

SNA [*userid*]

enables the SNA logical units for the VTAM Service Machine [VSAM] that you specify with *userid*. If you do not specify *userid*, CP enables all SNA units for all VSMs. See the VM/VCNA publications listed in the Preface for information on using SNA terminals.

ALL

enables the following previously disabled or non-enabled devices:

- Locally attached 3270 display terminals and display copy printers (3284/3286 etc.)
- All SNA (Systems Network Architecture) terminals

ENABLE

- EP (Emulator Program) controlled lines in the 370x
- 270x communication lines.

Note: Do not try to enable terminal devices that CP does not support as VM/SP virtual machine system consoles. You can only use these types of devices as dedicated devices attached to multiple-access system virtual machines.

System Response

COMMAND COMPLETE

Note: CP does not indicate an error if there are no lines available for it to enable. Issue QUERY LINES to see how many lines are enabled.

FLUSH

Privilege Class: D

When to Use FLUSH

Use the FLUSH command to halt and purge or, optionally, hold the current output on a specified real unit record device. The device resumes activity with the next scheduled spool file.

You will usually issue FLUSH when:

- You see that a mistake in output queuing has occurred, or
- You must start a high priority job before CP finishes the current job.

A mistake in output queuing might be that a user directed a file to a printer that is equipped with the wrong printer train to process that file. Or a user may misdirect a file to the real punch instead of his virtual punch unit. For example:

FLUSH E HO

FLUSH Format and Operands

Flush	<i>raddr</i> [ALL] [HOLD]
-------	---------------------------

raddr

is the address (cuu) of the real printer or punch whose activity you want to terminate.

ALL

deletes all copies of the current output file. If you specify ALL, and CP is printing or punching more than one copy of the current output file, CP deletes all copies. If you do not specify ALL, CP deletes only the current copy and prints or punches the next copy.

HOLD

places the spool file in system hold status. CP does not purge the terminated spool file. If you want to reset the status, use the CHANGE command.

If you do not specify HOLD, CP requeues the file and starts it on another device. You must issue both the FLUSH and DRAIN commands to recover the file.

FLUSH

Things You Should Know about the FLUSH Command

1. For buffered printers, such as the 3800 printer, the FLUSH command does not affect files that have already been sent to the printer.
2. If CP starts a spool file on a real output device that it cannot make ready, CP issues an error message to you. Recover the file and restart it by entering the following commands:

```
FLUSH raddr [HOLD]
DRAIN raddr
```

After you enter these commands, CP issues you the FATAL I/O ERROR and logically places the device offline. Use the VARY command to put the device back online.

System Response

```
{PRT} raddr FLUSHED USERID FILE CLASS RECORDS REC LEFT CPY DIST SEQ
{PUN}          userid file class norecs recsleft amt  userid seq
```

If you specify the HOLD option, CP gives the FILE HELD response in addition to the FLUSHED response.

raddr

is the real address of the printer or punch that you want to affect.

userid

identifies the user who issued the command. With DIST, **userid** identifies the user who will receive the output.

file

is the unique spoolid number that identifies the file to the VM/SP system.

class

was the class of the printer or punch when it started and the class that the file was spooled as.

norecs

is the total number of logical records in the file.

recsleft

is the number of records left in the file after the system executes the command.

amt

is the number of copies you request to be printed or punched.

seq

is a sequence number that the system resets when you IPL. For example, seq=5 means that this is the fifth file on this printer since IPL.

FORCE

FORCE

Privilege Class: A

When to Use FORCE

Use the FORCE command to force a logoff of any user on the system.

Note: The FORCE command is like issuing the LOGOFF command for another user in that the system does no special processing to terminate outstanding virtual machine I/O simulation. Thus, if an I/O device (such as a disk or tape drive) drops ready while it is processing virtual machine I/O activity, the virtual machine user performing the I/O on that device will not be able to continue processing. If you issue the FORCE command at this time, it will not be effective because it cannot complete until all waiting I/O is finished. Determine which I/O device is involved and make that device ready so that the user can continue operations or be forced from the system. Use the QUERY command to determine the status of all or selected I/O devices.

FORCE Format and Operands

FORCE	<i>userid</i>
--------------	---------------

userid

identifies the user who you want to log off.

In some cases, such as when system throughput is low due to thrashing, you may need to force one or more users from the system. In this case, when it is unclear as to which users you should force off, base your selection on which users have VMSAVE specified in the CP directory.

Force a VMSAVE user, if possible, because the system will restore the current state of the user's virtual machine when the user is logged back on. If you force a non-VMSAVE user, the system destroys its current virtual state and will not restore it when the user is logged back on. In this case, the user will have to re-IPL the virtual machine at LOGON.

System Response

LOGOFF AT hh:mm:ss zone weekday mm/dd/yy BY SYSTEM

The user receives the normal accounting message produced at logoff in addition to this logoff message.

```
{ GRAF raddr } LOGOFF AS userid USERS = nnn FORCED
{ LINE raddr }
{ DEV  rid  } DSCONNECT
```

raddr
is the real line address.

rid
identifies the real resource.

userid
identifies the user who CP logged off.

nnn
is the number of users still on the system.

Note: If an unrecoverable I/O error occurs on the virtual console, the system may force a virtual machine user into a disconnected state. CP then issues the preceding message with DSCONNECT instead of LOGOFF. The user may log on at any time while his disconnected machine is still running. See the *System Logic and Problem Determination Guide Volume 1 (CP)* for more information.

userid NOT LOGGED ON

You will get this response if the user is not logged on at the time when you issue FORCE, or if the user has already started the LOGOFF/FORCE process but has virtual I/O pending. (An interrupt pending condition on a virtual I/O device could cause the virtual I/O pending condition.)

FREE

FREE

Privilege Class: D

When to Use FREE

Use the FREE command to take a set of spool files out of system hold status. Only a spooling operator can free a spool file held by a spooling operator. The user or a spooling operator can free a spool file held by the user. If a spool file is in double hold status, CP can only process it after you remove both holds (system and user).

You can use the FREE command with the HOLD command to free output files that are in a hold status. Use HOLD to detain a file or a group of files temporarily to correct or change a spooling condition. You will severely impact the spooling area, if you block files from spooled output.

The following occurrences may hold a file from output:

- A device restriction
- An I/O error on the printer or punch
- A device modification.

For example, a printer may not be available, or it may have an incorrect print train mounted. When you correct the problem, CP releases the files for output.

FREE Format and Operands

FR ee	<i>userid</i> [Printer PUnch <u>ALL</u>]
--------------	---

userid

is the user whose spool files you want CP to release from a system hold status. The user does not need to be logged on, but HOLD may have previously held his files.

Printer
PRT

freed the printer files for the user that you specify.

PUnch
PCH

freed the punch files for the user that you specify.

ALL

frees all printer and punch files for the user that you specify.

System Response

None.

HALT

HALT

Privilege Class: A

When to Use HALT

Use the HALT command only in extreme cases to end any active channel program on a real device that you specify. VM/SP issues a RESET IMMEDIATE command to reset the status.

Note: Only use HALT if every other method you have tried fails. If you use HALT indiscriminately, you may cause unpredictable results.

HALT Format and Operands

HALT	<i>raddr</i>
------	--------------

raddr

is the address (cuu) of the real device that you want to stop.

Some device addresses have multiple I/O paths defined by the VM/SP system generation process. In these cases, HALT *raddr* causes CP to issue a HIO on only that specified path.

The HALT command does not affect 3704 or 3705 Communications Controllers running in NCP or PEP mode. See the class A NETWORK command.

System Response

DEVICE HALTED

indicates that VM/SP reset the status and halted the device.

HOLD

Privilege Class: D

When to Use HOLD

Use the HOLD command to place user spool files in a system hold status. You can hold files of any user on the system. You can reset both the user's files and his hold status with the FREE command.

Only a class D operator can free files that a class D user is holding. A spool file that a spooling operator or a user is holding can only be freed by the same spooling operator or user. You must remove both holds (system and user) before CP can process a spool file in double hold status.

HOLD Format and Operands

HOLD	<i>userid</i> [Printer PUnch ALL]
-------------	--

userid

identifies the user whose spool files you want CP to place in a system hold status. The user does not need to be logged on when you issue this command.

Printer

PRT

holds printer files for the user that you specify.

PUnch

PCH

holds punch files for the user that you specify.

HOLD

ALL

holds all the printer and punch files for the user that you specify.

System Response

None.

INDICATE

Privilege Class: A or E

When to Use INDICATE

Use **INDICATE** to help you monitor system performance. **INDICATE** displays at the console:

- The use of the major system resources of processor and storage
- The contention of the major system resources of processor and storage.

System analysts can be aware of heavy load conditions or low performance situations. Analysts may need to resolve these situations by using more sophisticated data collection, reduction, and analysis techniques.

Use **INDICATE** to identify:

- The users in queue1 and queue2
- The I/O devices that the users are queued up on
- The paging devices that may have been filled
- The execution characteristics of any user
- The total amount of resources that a user is using.

INDICATE Formats

Class A INDICATE Format

INDicate	FAVORed
-----------------	----------------

INDICATE

Class E INDICATE Format

INDicate	<div style="border-left: 1px solid black; border-right: 1px solid black; padding: 5px;"><p>FAVORed I/O <u>LOAD</u> PAGing <u>WAIT</u> <u>ALL</u></p><p>Queues USER <u>*</u> <u>userid</u></p></div>
-----------------	---

INDICATE Operands (Class A, E)

FAVORed

provides a list and SET FAVORED statistics for all users who have the favored execution option.

I/O

provides the userids of all the users in I/O wait state at that instant in time. This helps you foresee conditions that may lead to possible I/O contention within the system. CP also gives you the address of the real device to which the most recent virtual SIO was mapped. CP does not show the queue of users who issue SIOF to busy devices.

Because INDICATE I/O indicates a sample according to an instant in time, use I/O several times before you assume a condition is persistent. If the condition does persist, use the MONITOR SEEKS command to more thoroughly investigate the condition.

LOAD

provides information about VM/SP's operating load by displaying:

- The number of users in queue 1 and queue 2
- How real storage is being used
- The ratio of active users to users being serviced.

PAGing **WAIT** **ALL**

PAGing WAIT

displays the following:

- The userids of the users currently in page wait
- The number of page frames allocated on drum storage and on disk storage.

If your installation has 2305s as primary paging devices and other direct access devices as secondary paging devices, you may want to use this operand. If the primary device is full and the system is allocating paging space on the slower device, system performance may be lowered. Use PAGing WAIT when the Queues operand shows that a number of users in queue 1 and queue 2 are persistently in page wait.

PAGing ALL

displays the page residency data of all users of the system (including the system nucleus and pageable routines). The format of the reply message is the same as that of the PAGing WAIT operand.

Queues

displays the following:

- The active users
- The queues that they are in
- The storage that they are occupying
- The status that they are in.

This display shows the users that are currently dominating main storage. The display also includes the users waiting in eligible lists, because they are contending for main storage.

USER $\left[\begin{array}{c} * \\ \underline{\text{userid}} \end{array} \right]$

USER *

lets you determine the activity of your own virtual machine.

USER *userid*

lets you determine the resources that the specified virtual machine is using and the events that have taken place. If you are a class E user, you can access data from the VMBLOK of any user currently logged on to the system. This helps you understand an overload or poor performance situation.

You can use INDICATE USER, before and after you execute a program, to find the execution characteristics of a program and the resources that it uses. Do not, however, use INDICATE USER during the execution of a program when it is issuing spooling I/O requests, because CP updates some fields once and other fields dynamically. Do not halt or abort any spooling operation before it finishes, if you want to use the INDICATE USER command.

INDICATE

System Response

INDICATE FAVORED

CP repeats the following response three times per line. There may be as many lines as necessary to indicate the SET FAVORED options of all virtual machines logged on.

```
userid [Q] [P xxx yyy]
```

userid

identifies the user of the virtual machine that has one or both of the favored execution options.

Q

indicates that you specified SET FAVORED without a percentage for this userid.

P

indicates that you specified SET FAVORED with a percentage for this userid; both Q and P may appear in this response.

xxx

is the percentage you specified with SET FAVORED.

yyy

is the approximate percentage of the processor that the system has given the indicated virtual machine over the past eight minutes.

INDICATE I/O

```
userid1 xxx userid2 yyy
```

userid1

userid2

identifies the users.

xxx

indicates the real device address.

yyy

indicates that two users are waiting for I/O to complete on the indicated device.

Note: If a virtual machine issued multiple SIOs, the response shows the real device address corresponding to the most recent one issued.

CP issues the following response for the I/O operand when appropriate:

```
NO USERS IN I/O WAIT
```

INDICATE LOAD

```
{ CPU-nnn% APU-nnn% } Q1-nn Q2-nn STORAGE-nnn% EXPAN-nnn
{ PROC xx-nn% PROC yy-nn% }
```

PAGING-*nnnn*/sec, STEAL-*nn*%, LOAD-*nn*%

n
is always a decimal number.

CPU-*nnn*%
indicates the percentage of time that the main processor is running;
CP derives the percentage from the smoothed wait value that the
scheduler maintains.

APU-*nnn*%
is the percentage of time that the attached processor is running.

PROC *xx-*nn**%
is the percentage of time that the system is running on the IPL
processor.

PROC *yy-*nn**%
is the percentage of time that the system is running on the non-IPL
processor.

Q1-*nn* Q2-*nn*
represents the contention for the processor by the average numbers of
users in queue1 and queue2, maintained by the scheduler.

STORAGE-*nnn*%
is an approximate percentage that measures the usage of real storage.
It shows the relationship between the number of pages in storage for
in-queue virtual machines and the number of pageable pages in the
system.

Due to the algorithm that the scheduler uses to approximate the
number of pages in storage, the value of STORAGE can exceed
100%. The value may be less than 100%, even when the sum of the
estimated working set for in-queue virtual machines is greater than the
number of pageable pages.

EXPAN-*nnn*
is the total delay in response time that a virtual machine experiences,
because it contends for both real storage and the processor. If the
virtual machine does not contend for either resource, EXPAN is 1.

The scheduler contention value, EXPAN, is the measure of time it
takes a virtual machine to receive a given amount of processor time.
Its derivation follows:

$$\text{EXPAN} = \frac{QT}{Q2 \text{ slice}}$$

INDICATE

QT

is the average elapsed time between queue drops that a Q2 virtual machine spends in the Q2 dispatch list.

Q2 slice

is the amount of processor time that the system allows a Q2 virtual machine in the dispatch list.

PAGING-nnnn/sec

is the average number of page I/O operations (page reads and writes) that the system performs per second.

STEAL-nn%

is the percentage of page read operations that required CP to steal a real storage page from another in-queue virtual machine.

LOAD-nn%

is an artificial value that attempts to measure (in elapsed time) the percentage of the system devoted to paging because of real storage contention. The following contribute to the value of LOAD:

- Estimated processor time involved in paging
- The amount of time spent in pagewait
- The percentage of steals.

INDICATE PAGING

```
userid1 xxx:yyy  userid2  xxx:yyy
```

userid1

userid2

identifies the user.

xxx

are the number of pages, in hexadecimal, that CP allocates on drum storage for these users.

yyy

are the number of pages, in hexadecimal, that CP allocates on disk storage for these users.

Note: This response may indicate one or more users. This sample response indicates two users. If the two users shown in the response executed similar programs, userid1 would experience more page wait than userid2.

You should know which users are occupying most of the primary paging device space, and whether or not they are still active. CP may have allocated large amounts of primary paging device space at IPL time to a virtual machine that is executing a large operating system. If this paging space becomes inactive, the

machine is occupying a critical resource but is not using it. If you invoke the ALL operand, CP gives xxx and yyy values for all users on the system.

The system issues the following response for PAGING WAIT, when appropriate:

NO USERS IN PAGEWAIT

INDICATE QUEUES

userid aa bb sss/ttt userid

userid

identifies the user. (This response may display more than one user.)

aa

represents the eligible lists or queues that are occupied. The value is E1, E2, E3, Q1, Q2, or Q3.

bb

is one of the following status indicators:

--

runnable: the user is runnable and waiting for service by the CPU.

RU

is the current RUNUSER in uniprocessor applications. In attached processor applications, RU is the current RUNUSER on the main processor. In multiprocessor applications, RU is the RUNUSER value on the IPL processor.

RA

is the current RUNUSER on the attached processor. In uniprocessor applications, RA does not appear in the response line. In multiprocessor applications, RA is the RUNUSER value on the non-IPL processor.

PG

in page wait—the user's program is not executing because CP is trying to bring in a page from a paging device.

IO

in I/O wait—the user is in I/O wait because access to the device is not available at the moment.

EX

in instruction simulation wait—the user is waiting for the system to complete the instruction simulation.

PS

in PSW wait—the user is in an enabled wait state for high-speed I/O devices.

INDICATE

DF

in attached processor configurations, means that the system deferred the processing of a synchronous (program or SVC) interrupt for this user until the system lock is available. DF is not applicable for uniprocessor operations.

Note: If a virtual machine is in more than one of the above states, CP displays only one state. The state that CP displays is the first one it encounters in the order of priority indicated above.

sss

is a hexadecimal number that indicates the number of pages resident in real storage.

ttt

is a hexadecimal number that the dispatcher estimates to be the number of pages in the working set of the specified user.

Note: CP orders the lines of the above response as follows:

1. Eligible list E1 users in scheduling priority order
2. Eligible list E2 users in scheduling priority order
3. Eligible list E3 users in scheduling priority order
4. Q1, Q2, and Q3 users in runlist priority order (dispatching priority).

These values help you analyze system performance. For details see *VM/SP System Programmer's Guide*.

CP issues the following response for the INDICATE QUEUES command when appropriate:

NO USERS IN QUEUE

INDICATE USER

PAGES: RES-nnnn WS-nnnn READS=nnnnnn WRITES=nnnnnn MH-nnnn FH-nnnn
lists the data from the user's VMBLOK that relates to his or her virtual machine's paging activity and resource occupancy.

VTIME=nnn:nn TTIME=nnn:nn SIO=nnnnnn RDR=nnnnnn PRT=nnnnnn
PCH=nnnnnn
lists the following:

- The user's CPU usage
- The user's accumulated I/O activity counts since he logged on, or since you last issued ACNT for his virtual machine.

n

is a decimal number.

RES-nnnn

is the amount of the user's virtual storage pageable pages that are resident in real storage when the class E user issues the command.

WS-nnnn

is the most recent system estimate of the user's working set size.

READS

is the total number of page reads for this user since he logged on or since the class E user last issued the ACNT command for his virtual machine.

WRITES-nnnnnn

is the total number of page writes for this user since he logged on or since the class E user last issued the ACNT command for his virtual machine.

MH-nnnn

is the number, taken when the class E user issues the command, of virtual pages allocated on system DASD (moveable head) paging space for the particular user.

FH-nnnn

is the number, taken when the class E user issues the command, of virtual pages allocated on system DASD (or fixed head cylinder) paging space for the particular user.

VTIME=nnn:nn

is total virtual time for the user.

TTIME=nnn

is virtual and simulation time for the user.

SIO=nnnnnn

is the total number of nonspooled I/O requests that the user issued.

RDR-nnnnnn

is the total number of virtual cards that the system reads. CP gets this value from spool file control information and increments the total number of cards to be read once to the RDR=value. CP increments this value at the start of each file read operation.

PRT-nnnnnn

is the total number of virtual cards that the system prints. CP gets this value from spool file control information and increments the total number of lines to be printed once to the PRT=value. CP increments this value at the end of each file print operation.

INDICATE

PCH-nnnnnn

is the total number of virtual cards that the system punches. CP gets this value from spool file control information and increments the total number of cards to be punched once to the PCH=value. CP increments this value at the end of each file punch operation.

LOADBUF

Privilege Class: D

When to Use LOADBUF

Use the LOADBUF command for the following printers:

- The IBM 1403 — use LOADBUF to load the Universal Character Set (UCS) buffer with a print chain/train image that you specify.
- The IBM 3203, 3211, or 3262 — use LOADBUF to load the UCS buffer or the Forms Control Buffer (FCB) with an image that you specify.
- The 3289 Model 4 — use LOADBUF to load the Font Offset Buffer (FOB) with the image print belt and FCB.

Note: Do not load the 4245 and 4248 printers with the LOADBUF UCS command. Load the UCS image by the printer.

Drain the device before you issue the LOADBUF command. Use this command to load the FCB/UCS/UCSB/FOB buffer of a printer under the following conditions:

- Any time you want to change the print belt/chain/train
- If the buffer has been causing an excessive amount of parity errors
- When the printer or control unit is powered up since the last buffer load
- If the printer was previously attached to a user.

Load the FCB buffer:

- Anytime you want to change the FCB image
- If the printer was previously attached to a user.

LOADBUF Format and Operands

LOADBUF	$\left\{ \begin{array}{llll} raddr & \text{UCS name} & [\text{Fold}] & [\text{Ver}] \\ raddr & \text{FCB name} & [\text{Index}] & [\text{nn}] \end{array} \right\}$
----------------	---

raddr

is the address of the printer whose buffer you want the system to load.

LOADBUF

UCS *name*

name is a one to four character name of the UCS, UCSB, or FOB image that you want CP to load into the printer at "raddr." For the 4245 and 4248 printers, LOADBUF only verifies that the image name you specify here is really the image that the 4245 or 4248 printer loaded internally.

FCB *name*

name is a one to four character name of the FCB image that you want CP to load.

Note: When you want to load both a print chain image (UCS) and a forms control buffer (FCB), you must use two distinct LOADBUF commands for the specified printer.

The supplied UCS and FCB names are as follows:

- UCS--1403 or 3203
 - AN normal AN arrangement
 - HN normal HN arrangement
 - PCAN preferred character set, AN
 - PCHN preferred character set, HN
 - QN PL/I - 60 graphics
 - QNC PL/I - 60 graphics
 - RN FORTRAN, COBOL commercial
 - YN high speed alphameric
 - TN text printing - 120 graphics
 - PN PL/I printing - 60 graphics
 - SN text printing - 84 graphics
- UCS--3211
 - A11 Standard Commercial
 - H11 Standard Scientific
 - G11 ASCII
 - P11 PL/I
 - T11 Text Printing
- UCS--3262
 - P48 48-character belt
 - P52 52-character belt (Austria/Germany)
 - P63 63-character belt, optimized
 - P64 64-character belt
 - P96 96-character belt
 - P116 116-character belt (French - Canadian)
 - P128 128-character belt (Katakana)
- UCS--3289 - Model 4 (FOB)
 - F48 48-character belt
 - F64 64-character belt
 - F94 94-character belt
 - F127 127-character belt
- FCB--3203, 3211, 3262, 3289E, 4245, or 4248

CP provides two names for an FCB image: FCB1 and FCB8. See *VM/SP System Programmer's Guide*, for information on how to add more FCB images. Look in this section under "Things You Should Know about the LOADBUF Command" for information about FCB1 (6 lines/inch) and FCB8 (8 lines/inch).

Fold

loads the UCS or FOB buffer with the folding operation code, so the printer can print uppercase for lowercase bit configurations. Use LOADBUF UCS again without the FOLD option to turn FOLD off. Note that the LOADBUF UCS command does not load a buffer for the 4245 and 4248 printers. FOLD is valid for UCS but not for FCB.

Notes:

1. When the FOLD option compares character codes from the UCS buffer and the print line buffer, the control unit ignores bit positions 0 and 1 of the EBCDIC code. Thus, the printer only prints uppercase characters from either uppercase or lowercase data codes.
2. Use the FOLD option with discretion, since the printer may print more than one character for the same EBCDIC bit configuration, depending on the image used.
3. Depending on the image used, you may experience incorrect printing for a dash (-) and a backward slash (\), lowercase and uppercase characters, superscript numbers and standard numbers, and special character combinations.

Ver

prints the contents of the UCS/UCSB/FOB buffer on the printer that you specify to verify the buffer loading function. Ver is not valid for FCB.

You can use Ver for the 4245 and 4248 printers, but Ver performs no functions for these printers. The LOADBUF UCS command automatically verifies the band loaded in the 4245 and 4248 printers. The system displays messages for the 4245 and 4248, even if you don't specify the Ver option.

LOADBUF automatically sets the Block Data Check latch for the UCS/FOB buffer load. If you specify Ver after the system has executed the command, the printer that you specify prints the image of the UCS/FOB load. The printed output must correspond to the description of the specified buffer load in the following publications:

- *2821 Control Unit Component Description*
- *IBM 3211 Printer 3216 Interchangeable Train Cartridge & 3811 Printer Control Unit Comp. Desc. and Operator's Guide*
- *3262 Printer Models 1 and 11 Component Description.*

Use LOADBUF UCS with this option to send a block data check command code.

LOADBUF

Index *[nn]*

starts printing the output in the designated (*nn*) print position. *nn* must be a number from 1 to 31. If you specify INDEX without a value, the index value in the FCB macro becomes the index value. For a description of the FCB macro and forms control images, see the *VM/SP System Programmer's Guide*.

Index is not valid for a 3203 Model 4 or 5 printer or 3262, 3289E, 4245, or 4248 printers.

Things You Should Know about the LOADBUF Command

1. Spool files that were created on a virtual spooled 3203, 3211, 3262, 3289E, 4245, or 4248 may have virtual FCB images imbedded in them. When you print these spool files on a real 3211, 3203, 3262, 3289E, 4245, or 4248 printer, the system loads the imbedded FCB image into the real printer. The FILEFCB, CFILEFCB, or DEFFCB option that you specified on the START command dictates how the system will load the FCB image. For the FILEFCB and CFILEFCB options, the system does not use the FCB image that you loaded with the LOADBUF command.
2. The system prints spool files, created on a virtual spooled 1403 printer, with the FCB image loaded by the last LOADBUF command you issued.
3. For FCB1, when the length of page is 66 lines and there are 6 lines/inch, the following is true:

LSkip Represented	Channel Specification
1	1
3	2
5	3
7	4
9	5
11	6
13	7
15	8
19	10
21	11
23	12
64	9

For FCB8, when the length of the page is 68 lines and there are 8 lines/inch, the following is true:

LSkip Represented	Channel Specification
1	1
4	2
8	3
12	4
16	5
20	6
24	7
28	8
32	10
36	11
63	12
66	9

System Response

For all Printers Except 4245 and 4248: If you specify VER, the system prints the contents of the UCS and/or FOB buffer on the specified printer.

For 4245 and 4248 Printers:

UCS IMAGE VERIFIED

The system compares the UCS name with the UCS that is loaded in the printer, even if you don't specify VER. You get this message if the UCS name in the LOADBUF command matches the UCS that is loaded in the printer.

DMKCSB158I UCS IMAGE NAME MISMATCH.

IMAGE NAME LOADED IS xxxx

indicates that the the UCS name in the LOADBUF command does not match the UCS that is loaded in the printer.

LOCATE

LOCATE

Privilege Classes: C or E

When to Use LOCATE

Use the LOCATE command to find the addresses of CP control blocks associated with:

- A particular user
- A user's virtual device
- A real system device.

The control blocks and their use are described in the *Data Areas and Control Block Logic Volume 1 (CP)*.

LOCATE Format and Operands

LOCate	$\left\{ \begin{array}{l} \textit{userid} \textit{ [vaddr]} \\ \textit{raddr} \end{array} \right\}$
--------	---

userid

identifies the logged-on user. CP displays the address of this user's virtual machine block (VMBLOK).

vaddr

displays the following addresses with the VMBLOK address. These addresses are associated with the virtual device address that you specify:

- The virtual channel block (VCHBLOK)
- The virtual control unit block (VCUBLOK)
- The virtual device block (VDEVBLOK).

raddr

displays the following addresses associated with the real device address that you specify:

- The real channel block (RCHBLOK)
- The real control unit block (RCUBLOK)
- The real device block (RDEVBLOK).

Note: The *raddr* option is not valid for CP control blocks associated with SNA terminals that are connected with the VTAM Network Communications Application (VM/VCNA) product and with VTAM SNA Console Support (VSCS).

System Response

VMBLOK = xxxxxx
is the response for the LOCATE *userid* command.

VMBLOK	VCHBLOK	VCUBLOK	VDEVBLOK
xxxxxx	xxxxxx	xxxxxx	xxxxxx

is the response for the LOCATE *userid vaddr* command.

RCHBLOK	RCUBLOK	RDEVBLOK
xxxxxx	xxxxxx	xxxxxx

is the response for the LOCATE *raddr* command.

LOCK

LOCK

Privilege Class: A

When to Use LOCK

Use the LOCK command to permanently lock in real storage selected pages of:

- The pageable CP nucleus
- A user's virtual storage.

LOCK excludes the selected pages from future paging activity.

The LOCK command can make a virtual machine more efficient; LOCK keeps high activity pages, such as virtual page zero, in real storage if some other virtual machine is already using the reserved page frames option. If the amount of page frames available for paging is limited, do not issue LOCK without the approval of the system programmer.

Issue LOCK as many times as you need for one virtual machine to lock noncontiguous pages of storage. The remaining virtual machine storage blocks will stay pageable.

Note: If you lock too many pages of real storage, the remaining virtual machines may not have enough available page frames left to operate efficiently. Since the virtual machines will contend for the remaining available page frames, throughput may be severely degraded.

Other ways to unlock a locked page are:

- Use the SYSTEM CLEAR command, which unlocks the user's locked pages
- Issue DIAGNOSE 14, 30, 34, or 38 against the locked page.

LOCK Format and Operands

LOCK	$\left\{ \begin{array}{l} \text{userid} \\ \text{SYSTEM} \end{array} \right\} \text{ firstpage lastpage [MAP]}$
-------------	---

userid
identifies the logged-on user.

SYSTEM
locks one or more of the pageable CP pages. Once a page is locked, it remains locked until the user either logs off the system, or issues the

UNLOCK command for that page. If you have the locked pages option you can unlock pages by doing any of the following:

- Re-IPL the system by device address, and specify the clear option. The system unlocks the pages, and they become available to the system being loaded.
- Re-IPL the system by device address, and do not specify the clear option. All locked pages stay locked, except the page given to DMKVMI for IPL.
- Re-IPL the system by name (shared system). The system unlocks the pages only if:
 - The locked pages are not in the shared segment, or
 - The page is in the shared segment, and the user who is re-IPLing is the last user of the shared segment.

Example:

```
LOCK SYSTEM 2A 2A MAP
VIRTPG REALPG (response to the MAP operand)
02A000 04F000
DCP 4F038-4F048
etc.
```

```
UNLOCK SYSTEM 2A 2A
```

Locks a CP page for display purposes and unlocks it when finished.

firstpage

is the hexadecimal value of the first user page that you want the system to bring into storage and lock.

lastpage

is the hexadecimal value of the last user page that you want the system to bring into storage and lock. If you only want the system to bring one page into storage, *lastpage* must be the same as *firstpage*.

For *firstpage* and *lastpage*, specify only the page numbers.

Examples:

```
LOCK USERA 12 2C
locks USERA's virtual storage locations X'12000' to X'2C000'
in real storage.
```

```
LOCK USERA 0 2
locks the first three pages of USERA's storage.
```

LOCK

LOCK USERA BA BA

locks one page of USERA's storage.

MAP

prints a map of the virtual storage pages locked and the real page that is assigned for each page locked.

Note: If you omit the MAP operand from a LOCK command and you later want to find the real page, reissue LOCK with the MAP operand. You do not need to unlock the page first.

The SYSTEM and MAP operands help you look at (with DCP) or change (with STCP) a page that is normally not resident, such as console function routines. The MAP operand locates the real page for display purposes. Page numbers are defined as follows:

Storage Locations (in hexadecimal)	Page Number
000-0FFF	0
1000-1FFF	1
2000-2FFF	2
.	.
.	.
.	.
12000-12FFF	12
.	.
.	.
.	.

Things You Should Know about the LOCK Command

1. You cannot lock shared pages in a system generated for AP or MP operation. However, in uniprocessor mode, if the system was not generated for AP or MP operation, you can lock a shared page. If the shared page becomes nonshared, the system unlocks it. In addition, if you lock a protected shared page and then change the page, the system unlocks it.
2. The virtual pages locked in processor storage are blocks of 4K (4096) bytes. This block of storage does not need to represent all of the user's virtual storage. You can issue the LOCK command as many times as needed for one virtual machine to lock noncontiguous pages of storage. The remaining virtual machine storage blocks may remain pageable.

System Response

COMMAND COMPLETE

is the response when LOCK finishes processing.

VIRTPG	REALPG
virt-firstpage	real-firstpage
.	.
.	.
.	.
virt-lastpage	real-lastpage

is the response to the MAP operand.

MESSAGE

MESSAGE

Privilege Class: A or B

When to Use MESSAGE

Use the MESSAGE command to:

- Transmit message text to a specified userid
- Transmit message text to the primary system operator
- As the primary system operator, send message text to one or all logged-on users.

If you want the users to receive messages automatically when they log on, see the class B SET command.

MESSAGE Format and Operands

Message Msg	$\left\{ \begin{array}{l} \text{ALL} \\ \text{userid} \\ * \\ \text{OPerator} \end{array} \right\} \text{msgtext}$
----------------	--

ALL

is for the operator's use only. It lets you broadcast a message to all logged-on users.

userid

identifies the user to whom you want to send the message.

*

sends a message to yourself.

OPerator

sends a message to the primary system operator, without having to know his or her userid.

msgtext

is the message you want to transmit. You can enter as many characters as will fit on the rest of the input line.

Things You Should Know about the MESSAGE Command

1. A user will not receive a message if he or she is not logged on or has suppressed the receiving of messages. You, the sender, will receive a message that tells you so. If you send a message to ALL, you get a message for every user that has message typing suppressed.

The system does not save a message that a user does not receive. You must send the message at a later time when the user is receiving messages.

2. A typewriter terminal or display terminal that has AUTOREAD set ON, will receive the message when the user hits a carriage return.

System Response

hh:mm:ss MSG FROM OPERATOR: msgtext
is the message format that the user receives from the system operator.

hh:mm:ss MSG FROM userid: msgtext
is the message format that a user receives from another user (specified with userid).

hh:mm:ss MSG FROM LOGONxxx: msgtext
is the message format sent from a user on the VM/SP system to another user. The message shows that the user has not achieved logon status. xxx is the real line address from which the MESSAGE command was issued.

For all of the above responses:

- hh:mm:ss is the time in hours:minutes:seconds when the message was sent to the user.
- If the user receiving the message is the primary system operator, the alarm bell at the central computer console rings.
- If you are using a 3270, the message puts the screen in a HOLD status and rings the audible alarm, if present. An exception to this is during system IPL or restart operations, in which case the screen is put in MORE status. Press the cancel key to return to the RUNNING status.

MIGRATE

MIGRATE

Privilege Class: A

When to Use MIGRATE

Use the MIGRATE command to:

- Activate the normal page/swap table migration routines
- Force the pages of the particular user to the secondary device, even if that user is currently active.

You will rarely use this command for a general user.

MIGRATE Format and Operands

MIGRate	<i>[userid]</i>
----------------	-----------------

userid

identifies the virtual machine user for whom you want to invoke page/swap table migration.

When you issue MIGRATE with the *userid* operand, the system invokes page/swap table migration only for the specified virtual machine. In addition, the normal time-limit criteria for determining infrequently used segments is not active. The system migrates all pages that belong to the specified virtual machine if:

- The pages are not currently in real storage.
- The pages are not in the process of being brought in or written out.

The system migrates all swap tables, as long as the segment they belong to does not have any pages that are not migrated.

Note: If you issue the MIGRATE command without *userid*, the system will immediately invoke the page/swap table migration, regardless of when it was last invoked.

System Response

PAGE MIG DONE

The system migrates out the virtual machine's pages until none of its "working set" of pages is resident. If the user becomes active, he will build up his "working set" on his own.

MONITOR

MONITOR

Privilege Class: A or E

When to Use MONITOR

Use MONITOR to Trace and Record Events

Use MONITOR to override the system generated monitor function, and to initiate or end the recording of events that occur in the real machine. This recording is always active after a VM/SP IPL (manual or automatic). The CP internal trace table records the following events:

- External interruptions
- SVC interruptions
- Program interruptions
- Machine check interruptions
- I/O interruptions
- Free storage requests
- Release of free storage
- Entry into scheduler
- Queue drop
- Run user requests
- Start I/O
- Unstack I/O interruptions
- Storing a virtual CSW
- Test I/O
- Halt device
- Unstack IOBLOK or TRQBLOK
- NCP BTU (Network Control Program Basic Transmission Unit)
- Clear Channel operation (CLCH)
- Simulated I/O interrupts.

Use the trace table to determine what happened before a CP system failure. For information on finding and using the internal trace table see the *VM/SP System Programmer's Guide*.

Use MONITOR to Measure Performance

To measure performance, MONITOR:

- Displays the status of the following:

- Internal trace table
- Each implemented class of data collection
- Specifications for automatic monitoring using spool files.

- Enables various classes of **MONITOR CALL**
- Specifies the time intervals for timer-driven data collection
- Starts and stops data collection by **MONITOR** using tape or spool files
- Overrides the automatic monitoring assigned at system generation time
- Closes the monitor spool files
- Specifies device addresses to be included in or excluded from a selection list for DASD seeks analysis.

For more information about the **MONITOR** command, see the *VM/SP System Programmer's Guide*.

MONITOR

MONITOR Format and Operands

MONitor	<pre> AUTODisk { ON } { OFF } CLOSE Display [SPOOL TAPE ALL] ENable { PERForm }¹ { RESPonse { SCHedule { USER { INSTsim { DASTap { SEEKs { SYSprof INTERval <i>nmmmm</i> [SEC]<i>mm</i> [MIN] LIMIT <i>n</i> [NOSTOP STOP SAMPLE] SEEKS { INclude <i>raddr raddr...</i> { EXclude <i>raddr raddr...</i> { DELETE { DISPLAY START [SPOOL [TO <i>userid</i>] [BUFFS <i>n</i>] CPTRACE TAPE <i>raddr</i> [MODE { 800 { 1600 { 6250 { 38K }] [BUFFS <i>n</i>] STOP [SPOOL]² CPTRACE TAPE TIME { FROM <i>h1:m1</i> TO <i>h2:m2</i> { FOR <i>hh:mm</i> { ALL { NONE </pre>
----------------	---

¹Select one or more of the classes, subject to the restrictions listed with the ENable operand.

²See the operand description for the default value.

AUTOdisk { **ON** }
 { **OFF** }

overrides the specification in the SYSMON macro for automatic startup of monitoring. If automatic monitoring is already active, only a MONITOR STOP command will stop it manually.

Note: In general, when you try to override the definitions of the SYSMON macro the results are temporary. Because the system does not do monitor checkpointing, an IPL or system crash restores the automatic monitoring definitions from SYSMON.

CLOSE

closes the current spool file, so it is available to the reader of the recipient virtual machine. The system continues monitoring with a new spool file. You can use CLOSE when the VM/SP monitor is spooling the collected performance data and you want to reduce the data thus far.

Display [**SPOOL**]
 [**TAPE**]
 [**ALL**]

displays the status of VM/SP monitor variables and the internal trace table. For any of the operands, the system lists the class of monitor call and its current enabled/disabled state.

SPOOL

displays the automatic monitoring specifications. These include:

- If automatic monitoring has been requested, and its start and stop times
- The number of monitor buffers that the system will use
- The userid of the virtual machine to receive the spool file
- The spool file class and record limit
- The classes of monitoring that the system will enable.

If automatic monitoring is already in progress, the system lists the spool file number and the number of monitor buffer records that the system already wrote to it.

TAPE

displays the status of the monitor classes and CPTRACE table.

ALL

displays both SPOOL and TAPE responses.

MONITOR

ENable {
PERForm
RESPonse
SCHedule
USER
INSTsim
DAStap
SEEKs
SYSprof

enables the monitor call classes that you specify. Each time this command completes successfully the system creates a new value for control register 8.

The results from using the the MONitor ENable command differ, depending on if data collection is active or inactive when you issue the command. If data collection is active (MONitor STArt has been issued), the system moves the new mask directly into control register 8, replacing the previous mask. The new mask takes effect immediately. Collection then continues with the classes just entered.

If data collection is not active when you issue the command, the system saves the new mask and does not start collecting data until you issue MONitor STArt. If you issue MONitor STArt without first issuing MONitor ENable, the system uses the SYSMON class specifications. Any mask stays in effect until the next MONitor STOP command.

PERForm

periodically saves system counters, which provide you with system resource usage data (for example, information about how busy the CPU is and how much storage is being used).

RESPonse

collects data on terminal I/O. This data helps simplify your analysis of command usage, user response time, and system response time. The data can relate user activity to system performance. In order for RESPonse to be valid, the system programmer must set the TRACE bit (1) in the LOCAL COPY file to a 1 and reassemble DMKMCC.

SCHedule

does the following:

- Collects data about scheduler queue manipulation
- Monitors the flow of work through the system
- Indicates the resource allocation strategies of the scheduler, which is one of the key functions of the system.

USER

periodically scans the chain of VMBLOKs in the system. The result is user resource and status data.

INSTsim

records every virtual machine privileged instruction that the control program (CP) standard simulation routines (DMKPRV and DMKPRW) handle. Because INSTsim records the instructions, it may

help your installation improve performance. This command does not record privileged instructions that DMKFPS simulates.

If the VMA feature is active, CP handles less privileged instructions for the virtual machines that are running with the feature activated.

DAStap

periodically samples device I/O activity counts (SIO) for online tape and DASD devices.

Note: Specify DAStap before the MONitor STArt TAPE command. You can disable DAStap at any time by re-entering MONitor ENable without DAStap.

SEEKs

collects data for every I/O request to DASD devices. Displays channel, control unit, or device contention and arm movement interference problems.

Notes:

1. If SET NOTRANS is in effect, CP does not extract information for V=R regions.
2. The MN700CYL field of the output record is in cylinders for FBA devices and CKD devices.

SYSprof

adds data to the DAStap and SCHEDULE classes. This data helps build a more detailed profile of system performance, so you can study DASD utilization.

Note: The system cannot activate the SYSPROF class unless both the DAStap and SCHEDULE classes are also active.

Restrictions:

- Every MONitor ENable command yields a new mask. Thus, for example, if the system is currently collecting PERFORM and USER classes and you enter MONitor ENable INSTsim, the system stops the PERFORM and USER classes and starts the INSTsim.
- If the system is collecting data when you issue a MONitor ENable command and CP detects an error in the command line, CP does not change the monitoring status. Conflicting keywords, keywords that the system does not recognize, or missing operands cause error messages.

MONITOR

INTerval *nnnn*

SEC
MIN

mm

Use this operand if:

- The system has enabled or will enable a class that involves collecting data at periodic intervals, and
- If you want to override the default intervals of 60 seconds and 2 seconds.

Consider the 60 second interval as the primary sampling interval. This interval sets how often the system collects PERForm, USER and DAStap classes of monitor data. The two second interval applies to the high-frequency data sampling routine. This routine collects statistics about system use from the I/O subsystem (channels, control units, and devices) when the DAStap class is enabled.

You can issue MONitor INTerval at any time, if data collection is already in progress; the new interval does not take effect until the current interval has elapsed. The system resets MONitor INTerval to the defaults when:

- You issue MONitor STOP.
- The monitor stops automatically.
- The system stops the monitor because of an unrecoverable I/O error.
- The end-of-tape is reached.
- The spool record limit is reached.

nnnn

is the interval between data collections. You can follow *nnnn* with SEC or MIN to represent seconds or minutes. If you give an interval without SEC or MIN, the system defaults to SEC. Do not specify more than nine hours (540 minutes or 32,400 seconds) or less than five seconds. If you do not specify an interval, the system sends an error message.

mm

is the collection interval for the high-frequency sampler. This operand applies to the DAStap class of data collection only. The default value for *mm* is two seconds. You can specify from 1 to 99 seconds, but *mm* must be less than the value that you specify for *nnnn*.

LIMIT *n*

NOSTOP
STOP
SAMPLE

overrides the LIMIT options of the SYSMON macro.

n

is a decimal number from 10 to 50000 that represents the maximum buffer count in each spool file. If you want to change the automatic monitoring specification without changing the limit number, specify an asterisk (*) instead of a decimal number for *n*.

NOSTOP

changes the automatic monitoring specification after the system reaches its limit and closes the spool file. This option lets MONITOR start again, after you change the limit.

STOP

changes the automatic monitoring specification after the system reaches its limit and closes the spool file. This option stops MONITOR, after you change the limit.

SAMPLE

n becomes the number of samples of PERForm, USER, or DASTap data (or any combination of the three) that you want the system to collect before closing the spool file.

After the system has collected *n* samples of data, the system closes the monitor spool file and moves it to the virtual reader of the data reduction virtual machine. Monitoring continues uninterrupted using a new spool file. If you want to specify automatic monitoring, use the SYSMON macro instruction in DMKSYS at system generation.

SEEKS { **IN**clude *raddr raddr...*
EXclude *raddr raddr...*
DElete
DISplay }

provides seeks trace data on non-2305 DASD devices that you specify. You can specify a range of addresses; for example, 270-275. This command lets you form, display, or delete a list of devices for data collection. With a list of collected data, you can decrease both the overhead to collect the data and the processor time to reduce the data.

INclude

creates a list of non-2305 DASD device addresses for which the system collects seeks data. If a list already exists, the system erases the old list. Issue the appropriate MONitor ENable and STArt commands to invoke this function.

EXclude

excludes non-2305 devices from the existing list. The system creates a new list and erases the old list.

DElete

cancels the list, and frees the occupied real storage.

DISplay

lets you review the contents of the list.

Notes:

1. The system keeps the device list that you specify with the **IN**clude or **EX**clude option across separate monitoring sessions. Use the **DIS**play

MONITOR

and DElete options to revise the list when SEEKs analysis is completed. The system does not clear a list when you enter MONitor STOP.

2. You can enter as many device addresses as you can fit on a single line. However, for performance reasons, keep the list as short as possible.

```
STArt [SPOOL3 [TO userid] [BUFFS n]
      CPTRACE
      TAPE raddr [MODE { 800
                       1600
                       6250
                       38K } ] [BUFFS n]
```

- Starts VM/SP monitor data collection to the spool file or tape
- Starts the CP internal trace table.

SPOOL [TO *userid*] [BUFFS *n*]

starts the VM/SP monitor data collection using a spool file to store the data.

Close the monitor spool file by a MONitor STOP or MONitor CLOSE command when:

- The spool file reaches the record count limit (specified in the SYSMON macro).
- The system restarts or shuts down.

The system generates the filename and filetype of the monitor spool file. The filetype identifies the date and time of starting. The class of spool file is specified in the SYSMON macro and defaults to M. If you do not specify the classes of data collection with an ENABLE command, the system uses those specified in the SYSMON macro.

userid

identifies the virtual machine user whose virtual reader will receive the reader files. These reader files result from the system when it stops collecting data, closes the spool file, and adds the spool file to a the reader files.

If you are issuing the START command and want to be the recipient virtual machine, specify an asterisk (*) for *userid*. If you omit the *userid* option, the system uses the *userid* in the SYSMON macro instruction. The option that you specify stays in effect until the system is re-IPLed or until someone issues a new command that uses the TO *userid* option.

³ The default value is the active trace facility that is SPOOL or TAPE.

BUFFS *n*

overrides the number of monitor buffers that the system uses, as specified in the SYSMON macro. The BUFFS option overrides SYSMON for the length of the data collection session. Then the system returns to the SYSMON specification. If the number of buffers specified in the SYSMON macro has been defaulted, the system adopts the defaults described in the MONitor STArt TAPE command.

CPTRACE

starts the tracing of events that occur on the real machine. The system records the events on the CP internal trace table in chronological order. When the table fills, recording continues by overlaying data from the beginning of the table. You can tell what data was overlaid by reading the time stamp for each entry.

TAPE *raddr* MODE

starts data collection onto a tape mounted on a 9-track tape drive or a 3480 tape drive. The system starts collecting data for the classes of monitor call that you specified in a previous MONitor ENable command. The system moves the mask that the MONitor ENable command saved into control register 8. The system collects the data into buffer pages in real storage. These pages are separate from the internal trace table pages. As each data page is filled, the system writes it on the tape.

raddr

is the real hexadecimal address of the tape drive that you want to use.

BUFFS *n*

specifies the number of 4096-byte buffers that the system will use to monitor operations. *n* is a value from 1 to 99. If you don't specify a value for *n*, the default is based on the real storage sizes shown below.

Storage Size	Default
Less than 1 Meg	2 buffers
Between 1 and 2 Meg	3 buffers
Between 2 and 4 Meg	4 buffers
Between 4 and 8 Meg	5 buffers
Between 8 and 16 Meg	6 buffers
Over 16 Meg	7 buffers

BUFFS 1 is valid if the command entry is limited to ENable PERForm for data collection. Once monitoring is in progress with one buffer specified, you cannot invoke MONITOR with more ENable operands. Single buffer operation is useful for basic performance analysis in minimum main storage configurations.

MONITOR

MODE

requests a mode setting for the reset value of tape density when the system starts to monitor. You can specify mode values of 800, 1600, 6250, or 38K. If you specify a density mode that the tape drive cannot handle, the control unit does not return an error condition. In this case, the system ignores the mode setting and uses the default control unit setting.

A MODE option of 38K is valid only for a 3480 tape. Since the 3480 records only at a density of 38K, the default for the 3480 is 38K.

STOP $\left[\begin{array}{l} \text{SPOOL} \\ \text{CPTRACE} \\ \text{TAPE} \end{array} \right]$

SPOOL

stops data collection on a spool file if automatic VM/SP monitor data collection is active. Monitoring will not start again (even if the current time is within the bounds of the TIME operand of the SYSMON macro), unless the system fails or shuts down and your installation reloads it.

CPTRACE

stops the internal trace table event tracing. Event recording stops, but the system does not release the pages of storage that have the CP internal trace table. You can restart tracing at any time by issuing the MONitor STArt CPTRACE command.

TAPE

stops data collection on tape. The system stores a zero mask in control register 8, and, thus, disables monitor call interruptions. The system writes out the last partially filled page, writes two tape marks, and rewinds and unloads the tape. The system then releases the two buffer pages that it obtained at the time it processed the MONitor STArt TAPE command.

Note: If you do not specify an option and the VM/SP monitor is active, data collection stops whether or not a spool file or tape is in use. You must use the CPTRACE option to stop internal tracing. Note that the operands are separate functions, and commands that affect the status of one function have no effect on the other.

TIME $\left(\begin{array}{l} \text{FROM } h1:m1 \text{ TO } h2:m2 \\ \text{FOR } hh:mm \\ \text{ALL} \\ \text{NONE} \end{array} \right)$

overrides the automatic monitoring start and stop times defined by the SYSMON macro instruction.

FROM *h1:m1 TO h2:m2*

specifies a start and stop time in hours and minutes using a 24-hour clock. The default start time is 09:00 and end time is 17:00.

FOR

specifies the length of time that you want the system to collect data. Use FOR to collect performance data over short periods of time for testing or benchmarking. Automatic monitoring must be in effect before you specify this parameter (AUTO=YES in the SYSMON macro of DMKSYS or MONitor AUTODisk ON command). When you specify this option, the new time overrides the time defined in the SYSMON macro.

ALL

specifies 24-hour monitoring.

NONE

tells the system not to monitor performance.

Things You Should Know about the MONITOR Command

The MONitor command's response to unusual conditions are:

During Automatic Monitoring Using Spool Files

Shutdown: The system stops collecting data with a MONitor STOP SPOOL command. The system closes and moves the spool file for data reduction to the reader chain of the designated virtual machine.

System Failure: The system abandons the current monitor data collection buffer and closes the spool file that has all previous output buffers. The system then makes the spool file available to the virtual reader of the designated virtual machine for further data reduction.

Initial Program Load: The system automatically starts monitor data collection if:

- The SYSMON macro of DMKSYS specifies automatic monitoring. A system programmer at your installation is responsible for setting up DMKSYS.
- The IPL occurs within the time range that the macro specifies.

CP will inform you, but you do not need to do anything. You can, however, use the MONitor Display SPOOL command to check the specifications for automatic monitoring.

MONITOR

Spool Space: If automatic monitor data collection is regularly active and the system is issuing messages that show a serious loss of spool space, do the following checks:

- Be sure that the system reads and reduces the monitor spool files on a day-to-day basis. This is so that the system gets back the occupied spool space.
- Use MONitor DISPLAY to ensure that the data collection classes do not include trace classes, which produce large volumes of data (e.g. INSTsim or SEEKs). Keep in mind that the specifications in the DMKSYS macro SYSMON may be temporarily overridden by you or someone else using the MONitor command.
- If you have little spool space, the system can monitor to tape or use the LIMIT parameter. You can use the LIMIT parameter of the SYSMON macro or the LIMIT option of the MONITOR START SPOOL command to restrict the size of the monitor spool file.

Suspension: Data collection suspension may occur because of insufficient buffer space. You may need to increase the buffer allocations. Use the BUFFS n option of the MONitor STArt command until you can revise the SYSMON macro in DMKSYS.

During Manual Monitoring Using Tape

Suspension: Data collection suspension may occur because of insufficient buffer space. This usually occurs when:

- Classes USER and/or DASTap are enabled.
- Large numbers of users are logged on.
- The system I/O configuration is extensive.

Avoid losing data by specifying more buffers with the BUFFS option of the MONitor STArt TAPE command.

Unrecoverable Tape Error: Unrecoverable tape errors are errors that the I/O error handling routines cannot recover from by standard recovery procedures. When an unrecoverable error occurs, monitor tries to write two tape marks and rewind/unload the tape. The system abandons the use of the tape and data collection stops. The system informs the operator or takes action. Whether or not the write-tape-marks and rewind/unload is successful, the system releases the tape drive.

End-of-Tape Condition: When end-of-tape occurs, the system writes a tape mark on the tape and rewinds and unloads it. The system stops monitoring, and informs you of the action taken.

Initial Program Load: MONitor STArt CPTRACE is active after real system IPL (manual or automatic). The monitor tape data collection is off after IPL.

Shutdown: System shutdown implies a MONitor STOP TAPE command. The system performs normal command processing for the STOP TAPE function.

System Failure: If the VM/SP system fails and data collection is active, the system tries to write two tape marks and rewind/unload the tape. If the tape drive fails to rewind and unload, write a tape mark before you rewind and unload the tape. The system failure stops monitor data collection.

I/O Devices: You must dedicate a supported tape drive to the system for the duration of the monitoring.

System Response

COMMAND COMPLETE

for MONITOR commands, other than MONITOR DISPLAY, that run successfully.

CLS	KEYWORD	STATUS
nn	keyword	ENABLED
nn	keyword	DISABLED

.	.	.
.	.	.
.	.	.

-- CPTRACE ENABLED

if you issue the MONITOR DISPLAY command and the data collection is on tape.

If monitor is using spool files, the automatic monitoring specifications and status are listed as described in the MONITOR DISPLAY section.

MONITOR HAS BEEN MANUALLY STARTED

when you issue the MONitor STArt CPTRACE command. Only the primary system operator gets this message.

SELECTIVE SEEKS NOT CURRENTLY ACTIVE

occurs if SEEKS is not active at the time you issued the DISPLAY option.

SEEK, STOP, OR CLOSE CMD IS IN PROCESS, RETRY

if you issue a MONITOR command when the monitor is suspended.

MONITOR HAS BEEN MANUALLY STOPPED

when you issue the MONITOR STOP command. Only the primary system operator gets this message.

MSGNOH

MSGNOH

Privilege Class: A or B

When to Use MSGNOH

Use the MSGNOH command to:

- Transmit message text to a specified userid
- Transmit message text to the primary system operator
- As the primary system operator, send message text to one or all logged-on users

without the standard header associated with the MESSAGE command.

MSGNOH Format and Operands

MSGNOH	$\left. \begin{array}{l} \textit{userid} \quad \textit{msgtext} \\ \text{ALL} \\ * \end{array} \right\}$
---------------	--

userid

identifies the virtual machine to whom you are sending the message.

msgtext

is the message text that you want to transmit. You can enter as many characters as will fit on the rest of the input line.

ALL

sends the message to all users.

sends a message to yourself.

Note: MSGNOH also accepts a keyword of OPERATOR.

Things You Should Know about the MSGNOH Command

You can have your TELL EXEC use MSGNOH to send messages to users on the same system as you. Change the default set up in the GLOBALV file by issuing:

```
DEFAULTS set tell msgcmd msgnoh
```

When you set the default, it remains that way until you reset it. To change default back to MSG, issue:

```
DEFAULTS set tell msgcmd msg
```

System Response

```
msgtext  
the specified virtual machine receives the message text only.
```

NETWORK

NETWORK

Privilege Class: A or B

When to Use NETWORK

Use the NETWORK command to:

- Control the operation of a 3704, 3705, or 3725 (collectively referred to as 37xx) control program operating in 270x emulation mode (EP).
- Load or dump the contents of a 3704 or 3705 communications controller.
- Control remote 3270 devices attached to VM/SP with binary synchronous lines.

Be aware of the different classes of resources that are defined at generation time for the 37xx control program and the 3270 remote support.

Examine the output from the first stage of the 370x control program generation to tell which resources are lines and which are terminals. The installation system programmer (or whoever performs the 370x control program generation) should prepare a cross-reference list of resource IDs and their characteristics, such as line or terminal, type of line, and location. You will use most of the NETWORK commands for terminal resources.

- NETWORK subcommands for terminals only are:

NETWORK ENable
NETWORK DISable
NETWORK Query ACTive
NETWORK Query FREe
NETWORK Query OFFline
NETWORK Query ALL

- NETWORK subcommands for either lines or terminals are:

NETWORK Query resid
NETWORK VARY ONline
NETWORK VARY OFFline

Using the NETWORK Command

For 37xx Functions

The NETWORK command does the following:

- Causes 3704/3705 dump operations
- Starts 3704/3705 load operations
- Enables or disables terminal resources
- Varies resources online or offline
- Stops all 37xx operations
- Queries 37xx resource status.

When operating with a 270x or an Emulation Program there is only one address for each logon device, and that is the physical subchannel address for the telecommunications line. For a simple leased line configuration, there is one resource ID for each line and one resource ID for each terminal (one terminal per line). These resource IDs alternate in numeric value.

For 3270 Devices

Use the NETWORK command to:

- Control remote 3270 resources
- Vary the polling delay on the binary synchronous lines that are connected to these resources.

You can use the NETWORK command if the remote 3270 resources are connected to a 2701, 2703, or to a 37xx in Emulation Program (EP) mode.

The operands that specifically affect remote 3270s are:

- SHUTDOWN (Class A)
- POLLdelay (Class A,B)
- ENable (Class A,B)
- VARY (Class A,B)
- DISAble (Class A,B)
- Query (Class A,B)

Referring to a Remote 3270 Resource: You will specify a four-digit hexadecimal number to refer to a remote 3270 resource (display station, printer, or control unit (lines) in the 3270 remote system). For example, in the number 0001, the two low order characters, 01, identify the resource for a display station or printer. This value indicates the relative position of the TERMINAL macro from the first TERMINAL statement within the CLUSTER/TERMINAL grouping. This value begins at 01. The two high-order characters, 00, are a relative line code associated with a physical Binary Synchronous Communication (BSC) line. This value indicates the relative position of the CLUSTER macro from the first CLUSTER statement within the DMKRIO module. This value begins at 00.

NETWORK

NETWORK Formats

Class A NETWORK Format

NETWORK	<p>ATTach <i>resid</i> [To] <i>userid</i> [As] <i>cuu</i></p> <p>DETach <i>resid</i> [From] <i>userid</i></p> <p>DISAble [<u>ALL</u> <i>resid</i> [<i>resid...</i>]]</p> <p>DISPlay <i>raddr</i> <i>hexloc1</i> { - } [<i>hexloc2</i> :] [<u>END</u>] { . } [<i>bytecount</i> :] [<u>END</u>]</p> <p>DUMP <i>raddr</i> [<u>IMMED</u> <u>OFF</u> <u>AUTO</u>]</p> <p>ENable [<u>ALL</u> <i>resid</i> [<i>resid...</i>]]</p> <p>LOAD <i>raddr</i> <i>ncpname</i></p> <p>POLLdlay <i>nnnn</i> [<u>ALL</u> <i>raddr</i>]</p> <p>Query [<u>ACTive</u> <u>OFFline</u> <u>FR</u>Ee <u>ALL</u> <i>resid</i> [<i>resid...</i>]]</p> <p>SHUTDOWN [<i>raddr</i> <u>ALL</u>]</p> <p>VARY { <u>ONline</u> } <i>resid</i> [<i>resid...</i>] { <u>OFFline</u> }</p>
----------------	---

Class B NETWORK Format

NETWORK	<p>ATTach <i>resid</i> [To] <i>userid</i> [As] <i>vaddr</i></p> <p>DETach <i>resid</i> [From] <i>userid</i></p> <p>DISAble [<u>ALL</u> <i>resid</i> [<i>resid...</i>]]</p> <p>DISPlay <i>raddr</i> <i>hexloc1</i> { - } [<i>hexloc2</i> :] [<u>END</u>] { . } [<i>bytecount</i> END]</p> <p>DUMP <i>raddr</i> [<u>IMMED</u> OFF AUTO]</p> <p>ENable [<u>ALL</u> <i>resid</i> [<i>resid...</i>]]</p> <p>LOAD <i>raddr</i> <i>ncpname</i></p> <p>POLLdlay <i>nnnn</i> [<u>ALL</u> <i>raddr</i>]</p> <p>Query [<u>ACTive</u> OFFline FREe ALL <i>resid</i> [<i>resid...</i>]]</p> <p>VARY { <u>ONline</u> } <i>resid</i> [<i>resid...</i>] { OFFline }</p>
----------------	--

NETWORK Operands (Class A, B)

ATTach *resid* [**TO**] *userid* [**AS**] *vaddr*

attaches the remote printer device (*resid*) to a virtual machine (*vaddr*). Before you issue this command, be sure the device is powered on and enabled; otherwise, the advanced features of the device will not be operational.

NETWORK

resid

identifies the device. *resid* is the four-digit resource ID as specified in the real I/O configuration module (DMKRIO). For a complete description of *resid*, read about the four-digit hexadecimal number under "Referring to a Remote 3270 Resource" on page 3-103.

userid

identifies the user to whom you want to attach the device.

vaddr

is the virtual device address of the virtual machine where you want to attach the printer. A network resource is represented as a graphic device in the user's virtual machine configuration. Therefore, you should specify an address on a SHARED virtual control unit (VCU).

When you issue NETWORK ATTACH, be aware that VM/SP does not support a mixture of SHARED and NONSHARED device types on the same virtual control unit. If you do not specify *vaddr* correctly, the system does not attach the device. Refer to Appendix B in the *VM/SP Planning Guide and Reference*, SC19-6201 for a complete list of the virtual device characteristics.

Note: NETWORK treats the remote printer as a shared device in the virtual machine configuration. When you are attaching a remote printer to a virtual machine, the virtual machine may only access the 3270 Information Display printer with an SIO instruction.

DETach *resid* [**From**] *userid*

detaches a remote printer device that you request from the virtual machine.

resid

identifies the device. *resid* is the four-digit resource ID as specified in the real I/O configuration module (DMKRIO). For a complete description of *resid*, read about the four-digit hexadecimal number under "Referring to a Remote 3270 Resource" on page 3-103.

userid

identifies the user from whom you want to detach the device.

DISAble [**ALL**
[*resid* [*resid*...]]

disables 37xx resources (terminals only) and remote 3270 resources.

If you specify a resource that the system is currently using, CP does not immediately disable it. CP disables the resource when it becomes free (usually after the user logs off).

ALL

disables all 37xx terminals.

resid

disables resources that you select. *resid* is the four-digit resource ID of each terminal resource that you want to disable. For a complete description of *resid*, read about the four-digit hexadecimal number under “Referring to a Remote 3270 Resource” on page 3-103.

DISP *raddr hexloc1* [{ - } [*hexloc2*]]
 [{ : } [**END**]]
 [{ . } [*bytecount*]]
 [**END**]]

displays the contents of 37xx storage in fullwords. This operand does not provide EBCDIC translation.

raddr

is the real address of the 37xx whose storage you want CP to display.

hexloc1

specifies the hexadecimal address of the start of the display. This operand is required. Specify a colon (:), dash (-) or period (.) to display more than one fullword.

hexloc2

specifies the hexadecimal location of the end of the display.

DUMP *raddr* [**IMMED**]
 [**OFF**]
 [**AUTO**]

dumps the contents of 370x storage for EP 370x control programs.

raddr

is the real address of the 370x that you want CP to dump.

IMMED

dumps the 370x immediately. See “Network Dump Operations” on page 4-41 for additional information.

OFF

automatically dumps the 370x if the 370x control program abnormally ends.

AUTO

automatically dumps and reloads the 370x if the 370x control program abnormally ends (ABENDS).

ENable [**ALL**]
 [*resid* [*resid...*]]

activates 37xx resources (terminals only) and remote 3270 resources for VM/SP to use. The resource that you specify must be a terminal device. The NETWORK ENABLE command:

1. Ensures that the associated line resource is activated

2. Enables the terminal device or, in the case of remote 3270s, formats the screen.

Note: If you enable a 3275 remote display unit, and the unit is equipped with a printer to handle the display's hard copy output (via PFnn COPY), you must also enable the printer.

ALL

enables all 370X resources.

resid

enables resources that you select. *resid* is the four-digit resource ID of each terminal resource that you want CP to enable. For a complete description of *resid*, read about the four-digit hexadecimal number under "Referring to a Remote 3270 Resource" on page 3-103.

LOAD *raddr ncpname*

loads an EP 3704/3705 control program.

raddr

is the real address of the 3704/3705 that you want CP to load.

ncpname

is the name of the 3704/3705 control program image that you want CP to load into the specified 3704/3705. This name is previously defined by a NAMENCP macro and saved on a CP volume.

POLL *dlay nnnn* **[ALL]**
raddr

changes the duration of the polling delay interval for the binary synchronous line to *nnnn*.

The polling delay interval, defined at system generation, is a half second. The polling delay interval minimizes unproductive polling and processor meter time. In general, if no data or other communications is being received from the stations on the binary synchronous line, the system starts the polling delay interval and gives control to the dispatcher.

Note: The polling delay interval is the amount of time from when a binary synchronous line receives a negative response from a general polling sequence until the polling delay interval expires, or CP sends a message to the station on the binary synchronous line.

nnnn

is the polling delay interval. *nnnn* is a decimal number in tenths of a second (not to exceed 9999).

raddr

is the address of the binary synchronous line.

ALL

sets the polling delay interval for all the 3270 remote lines.

Query [**ACTive**
OFFline
FRee
ALL
resid [*resid...*]]

displays the status of 37xx resources (lines or terminals) and remote 3270 resources. You can use NETWORK QUERY to display the status of a line resource, but only when you use the NETWORK QUERY resource command format.

ACTive

displays the resources (terminals, remote 3270 display and printer stations) that are being used by VM/SP users.

OFFline

displays the resources that are not available to VM/SP users.

FRee

displays the resources that are not offline and also not currently in use.

ALL

displays the active, offline, or free status of each resource attached to all 37xx and 3270 remote control units on the VM/SP system.

resid

displays only the resources that you specify with the four-digit hexadecimal identifier. For a complete description of *resid*, read about the four-digit hexadecimal number under "Referring to a Remote 3270 Resource" on page 3-103.

SHUTDOWN [*raddr*]
[**ALL**]

controls 3270 remote device applications.

Use NETWORK SHUTDOWN to detach a binary synchronous line that is coupled to a 3270 remote console unit cluster or work station. CP places either of the following in a disconnected state:

- A virtual machine dependent on the binary synchronous line(s)
- A virtual machine dependent on the 37xx for which you issued NETWORK SHUTDOWN.

raddr

identifies the 3270 remote binary synchronous line on which you want all telecommunications to stop.

NETWORK

ALL

stops telecommunications on all binary synchronous lines in the 3270 remote system configuration.

CP does not attempt to preserve line status or messages in the 37xx.

VARY {**ONline** } *resid* [*resid...*]
{**OFFline**}

varies the status of the 37xx resource that you specify. You can use VARY ONLINE and VARY OFFLINE for line resources. However, use these commands primarily for terminal resources, because the state of the line changes automatically if you enable or disable the terminal.

Examine the output from the first stage of the 37xx control program generation to tell which resources are lines and which are terminals.

ONline

places a resource (line or terminal) online.

OFFline

places a resource (line or terminal) offline.

Note: Only the ONLINE and OFFLINE operands are valid for remote 3270 resources.

resid

identifies the resource. *resid* is the four-digit resource ID of each terminal resource that you want CP to vary. For a complete description of *resid*, read about the four-digit hexadecimal number under "Referring to a Remote 3270 Resource" on page 3-103.

System Response

NETWORK ATTACH

```
resid ATTACHED AS cuu
CP successfully attached the resource.
```

NETWORK DETACH

```
resid DETACHED
CP successfully detached the resource.
```

NETWORK DISABLE

```
COMMAND COMPLETE
```

NETWORK DUMP

CTLR raddr DUMP COMPLETE
CP successfully dumped the 370x "raddr."

NETWORK ENABLE

COMMAND COMPLETE

NETWORK LOAD

CTLR raddr ncpname LOAD COMPLETE
CP successfully loaded the 370x "raddr" with the control program ncpname.

NETWORK QUERY

DEV resid LOGON AS userid
DEV resid DISABLE
DEV resid ENABLED
DEV resid OFFLINE

LINE resid ACTIVE
LINE resid OFFLINE

DEV resid1 ENABLED, DEV resid2 ENABLED,...
DEV resid1 DISABLE, DEV resid2 DISABLE,...
DEV resid1 OFFLINE, DEV resid2 OFFLINE,...

ACTIVE resid NOT FOUND
FREE resid NOT FOUND
OFFLINE resid NOT FOUND
resid NOT FOUND

Note: For any of the "..... NOT FOUND" responses, a problem exists with the Network Control Program and/or the 37xx.

ACTIVE

indicates that the line resource is online and has been activated by the system. Users may or may not be using the terminals on the line.

DISABLE

indicates that the resource is online, but VM/SP users cannot access it.

ENABLED

indicates that VM/SP users can access the resource.

LOGON

indicates that "userid" is using the resource as a virtual machine operator console.

OFFLINE

indicates that the resource is inactive and unavailable for use.

NETWORK

res id
identifies the real resource.

user id
identifies the user.

NETWORK VARY

COMMAND COMPLETE

ORDER

Privilege Class: D

When to Use ORDER

Use the ORDER command to reorder closed spool files of a device type that you specify. To determine the filename, filetype, originating userid, and other attributes of all of your files, use the QUERY command.

ORDER Format and Operands

ORDER	$\left[\begin{array}{l} \textit{userid} \\ \text{SYSTEM} \\ * \\ - \end{array} \right] \left\{ \begin{array}{l} \text{Reader RDR} \\ \text{Printer PRT} \\ \text{PUnch PCH} \end{array} \right\} \left\{ \begin{array}{l} \text{Class } c1 \text{ Class } c2... \\ \textit{spoolid1 } \textit{spoolid2} \\ \text{FORM } \textit{form1} \text{ FORM } \textit{form2}... \end{array} \right\}$
--------------	---

userid

identifies the user whose spool files the system is to purge.

SYSTEM

orders all spool files in the system.

*

orders your own spool files.

Reader

RDR

orders the reader files.

Printer

PRT

orders the printer files.

PUnch

PCH

orders the punch files.

Class c

orders the files by their class. *c* is a one-character field (0 through 9 or A through Z) that specifies classes for input and output spool files. This lets you place files in the order of classes that you specify with **Class c1**, **Class c2**, etc.

ORDER

spoolid

identifies the spool files and lets you order the files for processing.

FORM *form1* **FORM** *form2*

orders the files by their form. *form1* and *form2* are one to eight character form numbers. The form number is the user form if you specify the *_** option or let it default. The form number is the operator form if you specify the *userid/SYSTEM* option.

You can do sequencing with the ORDER command using a combination of CLASS *c*, FORM, and *spoolid* specifications.

Example:

ORDER PRINTER FORM XYZ CLASS A 1963 CLASS C

queues the printer files in the following order: all files with the form XYZ, all Class A files, the file with the spoolid 1963, all Class C files.

System Response

{ nnnn } FILES ORDERED
{ NO }

is the response to the user who issues the ORDER command.

PURGE

Privilege Class: D

When to Use PURGE

Use the PURGE command to remove closed spool files from the system either:

- Before the spooling device prints or punches the files, or
- Before the user reads the files.

You can purge any file in the system, regardless of its status, as long as it has not been selected for processing.

PURGE Format and Operands

PURge	$\left[\begin{array}{l} \textit{userid} \\ \text{SYSTEM} \\ * \\ - \end{array} \right]$	$\left\{ \begin{array}{l} \text{Reader RDR} \\ \text{Printer PRT} \\ \text{PUch PCH} \\ \text{ALL} \end{array} \right.$	$\left[\begin{array}{l} \underline{\text{ALL}} \\ \text{CLass } c1 \text{ CLass } c2 \dots \\ \textit{spoolid1} \textit{ spoolid2} \dots \\ \text{FORM } \textit{form1} \text{ FORM } \textit{form2} \end{array} \right]$
--------------	--	---	--

userid

identifies the user whose spool files the system is to purge.

SYSTEM

purges all files on the system, regardless of the *userid*.

*

lets you manipulate your spool files.

Reader

RDR

purges reader files.

Printer

PRT

purges printer files.

PURGE

PUnch
PCH

purges punch files.

ALL

purges all the files. The system ignores any operands that follow ALL. If you specify ALL after Reader, Printer, or PUnch, the system only purges the files for that device type.

Class *c*

purges the files for the class that you specify and for the device type that you specify. *c* is a one-character field (0 through 9 or A through Z) that specifies classes for input and output spool files. This lets you purge files in the order of classes that you specify.

spoolid

purges the files of the spool identification that you specify and for the device type that you specify.

FORM *form1* FORM *form2*

tells the system to purge all files that have the form numbers that you specify. *form1* and *form2* are one to eight characters. The form number is the user form if you specify the *** option or let it default. The form number is the operator form if you specify the *userid/SYSTEM* option.

You can use PURGE using a combination of CLASS *c*, FORM and *spoolid* specifications.

Example:

```
PURGE PRINTER CLASS A FORM XYZ CLASS D 619
```

purges the files in the following order: all class A files, all files with form XYZ, all class D files, and the file with spoolid 619.

System Response

```
{ nnnn } FILES PURGED  
{ NO }
```

shows the number of files that the system purged.

Note: You will not get this response if you have issued the SET IMSG OFF command. Therefore, if you are monitoring the RSCS virtual machine for your installation, do not specify this option. You need to see informational messages to accurately monitor the RSCS system activity.

QUERY

Privilege Classes: A, B, C, D, E, or F

When to Use QUERY

Use the QUERY command to request the following types of information:

- System status
- Machine configuration
- Paging and scheduling information.

(For 3704 or 3705 Communication Controllers and remote 3270 resources, see the Class A and B NETWORK command.)

QUERY

QUERY Formats

Class A QUERY Formats

Query	<p>AFFinity [<i>userid</i>] CPAssist ¹ CPLEVEL JOurnal ² LOGmsg Names PAGing PRIORity <i>userid</i> PROCEssr QDROP S370E SASSist ¹ SPMODE SRM { APAGes DSPSlice IB MAXDrum MAXWss MHFULL PB PCI PGMStat PGMTlim } <i>userid</i> Users [<i>userid</i>]</p>
-------	--

¹Use both QUERY CPASSIST and QUERY SASSIST to determine the current status of the expanded virtual machine assist portion of the Extended Control-Program Support:VM/370.

²The JOURNAL operand is valid only if STQUERY=YES is specified in the SYSJRL macro instruction in DMKSYS.

Class B QUERY Format

Query	<table style="border: none; width: 100%;"> <tr> <td style="border: none;">DAsd</td> <td style="border: none;">[Sysvirt Virtual]</td> <td style="border: none;">[<u>ACTive</u> ATTach FREe OFFline ALL]</td> <td style="border: none;">[PATHS]</td> </tr> <tr> <td style="border: none;">GRaf LINES UR TApes ALL</td> <td colspan="3" style="border: none;">}</td> </tr> <tr> <td style="border: none;">CPLEVEL DAsd <i>valid</i> DUMP Lnn LOGmsg MTime Names PROCEssr <i>raddr1 (-raddr2)</i> STATUS <i>raddr</i> STORAge SYStem <i>raddr</i> TDsk <i>userid</i> Users [<i>userid</i>]</td> <td colspan="3" style="border: none;">}</td> </tr> </table>	DAsd	[Sysvirt Virtual]	[<u>ACTive</u> ATTach FREe OFFline ALL]	[PATHS]	GRaf LINES UR TApes ALL	}			CPLEVEL DAsd <i>valid</i> DUMP Lnn LOGmsg MTime Names PROCEssr <i>raddr1 (-raddr2)</i> STATUS <i>raddr</i> STORAge SYStem <i>raddr</i> TDsk <i>userid</i> Users [<i>userid</i>]	}		
DAsd	[Sysvirt Virtual]	[<u>ACTive</u> ATTach FREe OFFline ALL]	[PATHS]										
GRaf LINES UR TApes ALL	}												
CPLEVEL DAsd <i>valid</i> DUMP Lnn LOGmsg MTime Names PROCEssr <i>raddr1 (-raddr2)</i> STATUS <i>raddr</i> STORAge SYStem <i>raddr</i> TDsk <i>userid</i> Users [<i>userid</i>]	}												

Class C QUERY Format

Query	<table style="border: none; width: 100%;"> <tr> <td style="border: none;">CPLEVEL LOGmsg Names PROCEssr <i>userid</i> Users [<i>userid</i>]</td> <td style="border: none;">}</td> </tr> </table>	CPLEVEL LOGmsg Names PROCEssr <i>userid</i> Users [<i>userid</i>]	}
CPLEVEL LOGmsg Names PROCEssr <i>userid</i> Users [<i>userid</i>]	}		

QUERY

Class D QUERY Format

Query	<pre> CPLEVEL Files [Class c[[FORM form] [userid ²] [*] [HOLD ¹ NOHold SYShold USERhold] Hold Printer { [Class c[[FORM form] [userid ²] [*] [HOLD NOHold SYShold USERhold] } [null ¹] PUnch { Reader { <i>spoolid</i> LOGmsg Names userid Users [userid] </pre>
-------	---

¹Specify the options in any order.

²Using a 1-4 digit, all numeric userid will cause unpredictable results for the QUERY command, which also has a 1-4 digit all numeric spoolid parameter.

Class E QUERY Format

Query	<pre> AFFinity [userid] CPAssist ¹ CPLEVEL JOurnal ² LOGmsg Names PAGing PRIORity <i>userid</i> PROCessr QDROP SASsist ¹ SRM { APAGes DSPSlice IB MAXDrum MAXWss MHFULL PB PCI PGMStat PGMTim } <i>userid</i> Users [<i>userid</i>] </pre>
-------	---

¹Use both QUERY CPASSIST and QUERY SASSIST to determine the current status of the expanded virtual machine assist portion of the Extended Control-Program Support:VM/370.

²The JOURNAL operand is valid only if STQUERY=YES is specified in the SYSJRL macro instruction in DMKSYS.

Class F QUERY Format

Query	<pre> CPLEVEL LOGmsg Names <i>userid</i> Users [<i>userid</i>] </pre>
-------	---

QUERY

QUERY Operands (Class A,B,C,D,E,F)

AFFinity [*userid*]

(For attached processor applications only) displays the affinity setting of a virtual machine that you specify or all virtual machines that are logged onto the system.

Set the affinity in the VM/SP directory OPTION statement or with the CP SET command. The affinity setting of a virtual machine indicates the processor (main or attached) that the system will use to run the programs of a virtual machine. If you do not set an affinity for a virtual machine, the system dispatches the virtual machine for execution on the first available processor.

Response:

USER	PROCESSOR AFFINITY
<i>user id</i>	<i>affinity</i>
.	.
.	.
.	.

user id

identifies the virtual machine user.

affinity

is the address of the processor (main or attached) that will execute the virtual machine instructions. *affinity* can be any hexadecimal value from 0 to 3F.

If NONE is in the response, the system did not set an affinity, and the virtual machine is dispatched for execution on the first available processor.

Note: In System/370 Model 168 installations, you can reverse the role of the attached processor and main processor with a console switch setting.

CPAssist

displays the current status of the CP assist function for the VM/SP system.

Response:

CPASSIST	{ ON }
	{ OFF }

ON

shows that the CP Assist portion of the Extended Control-Program Support:VM/370 is active.

OFF

shows that the CP Assist portion of the Extended Control-Program Support:VM/370 is inactive or not available on the System/370 processor.

For Query CPAssist (attached processor applications):

CPASSIST { ON } PROC nn, { ON } PROC nn
 { OFF } { OFF }

ON

shows that the CP assist portion of the ECPS:VM/370 is active for the processor, PROC nn.

OFF

shows that the CP assist portion of the ECPS:VM/370 is inactive or not available for the processor, PROC nn.

nn

is the processor address assigned to the main processor or the attached processor by the installation when the system hardware is installed. nn can be any hexadecimal value from 00 to 3F.

Note: To determine the active status of the Expanded Virtual Machine Assist portion of ECPS you need a positive response to QUERY CPASSIST and QUERY SASSIST.

CPLEVEL

displays the current level of the system, most recent generation, and the most recent IPL date.

Response:

VM/SP RELEASE xx, SERVICE LEVEL xxxx
GENERATED AT date time zone
IPL AT date time zone

}	DAsd	[Sysvirt _Virtual]	[ACTIVE ATTAch FREe OFFline _ALL]	[PATHS]
	GRaf			
	LINES			
	UR			
	TApes ALL			

ACTIVE

displays the status of the active devices in the group that you specify. An active device is one that a user or the system is using. Active devices do not include devices that are free or offline.

QUERY

ALL

as the first operand, displays all devices and the size of real storage.

Response:

The response is the same as if you issued the following commands:

```
QUERY STORAGE
QUERY UR
QUERY LINES
QUERY DASD
QUERY TAPES
QUERY GRAF
```

ALL

as the second operand, displays the status of all devices within the group that you specify.

Response

The response is the same as if you issued the following commands:

```
QUERY type ACTIVE
QUERY type ATTACH
QUERY type FREE
QUERY type OFFLINE
```

ATTach

displays all the devices that are dedicated to any user on the system. An attached device is also an active device.

DASd

displays the real addresses of disk or drum devices.

Response:

For Query DASd:

```
DASD raddr ATTACH TO userid vaddr
PATH raddr {OFFLINE
            {ONLINE [PROC nn[,PROC nn]]} 4
```

indicates that the real device specified by raddr is attached to a user's (user id) virtual machine at virtual address vaddr.

⁴ Additional lines (PATH...) will only appear if you specify the PATHS operand and the device has alternate paths. Note that in systems generated for MP mode, all devices have at least a second logical channel path.

DASD raddr CP SYSTEM volid nnn

PATH raddr { OFFLINE
ONLINE [PROC nn[,PROC nn]] }⁴

indicates that the real device specified by raddr is allocated to the system for use as user's minidisks. nnn is the number of active user's minidisks on the physical disk. volid is the volume serial number of the real disk.

DASD raddr CP OWNED volid nnn

PATH raddr { OFFLINE
ONLINE [PROC nn[,PROC nn]] }⁴

indicates that the real device specified by raddr is used by the system for paging and spooling activity. nnn is the number of active user's minidisks and T-disks (if any) on the physical disk. volid is the volume serial number of the real disk.

FREE

displays all the devices that neither a user nor the system is currently using. Free devices do not include offline devices.

Response

The system produces a response for each device that is not active or offline in the following format:

type raddr FREE

type raddr [PROC n1[,PROC n2]] DRAINED⁵
is the response for unit record devices.

Note: This response implies that no spool files are queued for this device.

type raddr [PROC n1[,PROC n2]] { ENABLED }⁵
DISABLED }
is the response for communication devices.

type raddr { FREE }
volid }
is the response for DASD devices with mounted volumes.

GRaf

displays the real address of each locally and remotely attached display device and graphics devices dedicated to a virtual machine.

For VTAM Service Machine (VSM) supported terminals, GRaf displays:

⁵ The system will display the processor address(es) if the system is generated for MP mode.

QUERY

- The userid of the VSM that controls the terminals from which users have DIALED a guest virtual machine
- The userid and luname (logical unit name) associated with each terminal dialed to the VSM
- The userid of each terminal's owner.

Response:

GRAF raddr [PROC n1[,PROC n2]] LOGON AS userid ⁶
shows that userid is currently logged on the terminal located at real address, raddr.

GRAF raddr [PROC n1[,PROC n2]] ATTACH TO userid vaddr ⁶
shows that the display device at real address, raddr, is attached to the virtual machine of user id at virtual address, vaddr.

GRAF raddr [PROC n1[,PROC n2]] DIALED TO userid vaddr ⁶
shows that the display device at real address, raddr, is dialed to the virtual machine of user id at virtual address, vaddr.

PRT raddr ATTACHED TO userid
shows that the remote printer device at real address raddr is dedicated to the virtual machine of user id.

GRAF raddr { OFFLINE
 ENABLED
 DISABLED
 DISABLE PND }
shows that the remote graphics device at real address, raddr, is either offline, enabled, disabled or has a disable pending.

vvvvvvvv CONTROLS THE FOLLOWING LU'S

GRAF luname LOGON AS userid

GRAF luname DIALED TO userid

shows that the VTAM service machine (VSM) whose userid is vvvvvvvv is controlling the graphics devices identified by the SNA logical unit names (luname). This response also identifies the userid of each user logged on through one of these devices, or the userid of the virtual machine that each device is DIALED to.

LINES

displays the real address of each communication line.

⁶ If you specified the PATHS operand or if the command was QUERY raddr, and the system is generated for MP mode, the system will add the processor addresses to the response line as shown.

Response:

```
{LINE} raddr [PROC n1[,PROC n2]] LOGON AS userid 7
{CONS}
```

shows that user id is currently logged on at the terminal located at real address, raddr.

```
LINE raddr [PROC n1[,PROC n2]] ATTACH TO userid vaddr 7
```

shows that the communication line at raddr is attached to the virtual machine represented by user id at virtual address, vaddr.

```
LINE raddr [PROC n1[,PROC n2]] DIALED TO userid vaddr 7
```

shows that the communication line at raddr is dialed to the virtual machine represented by user id at virtual address, vaddr.

Offline

displays the devices in offline status in the group that you specify. An offline device is not available for access by any user or the system.

Response

This command produces a response for each offline device in the following format:

```
type raddr OFFLINE [PATH raddr, ...] 8
```

Multiple responses are displayed in the following format:

```
type raddr OFFLINE, ...
:      :      :
:      :      :
:      :      :
```

Note: In the preceding responses the term type refers to one or more of the following device types:

Type	Meaning
DASD	Direct access device
TAPE	Magnetic tape units
LINE	Communication line
RDR	Card reader
PRT	Line printer

⁷ If you specified the PATHS operand or used the QUERY raddr command, and the system is generated for MP mode, the system will add the processor addresses to the response line as indicated.

⁸ The second line will only appear if the device is an alternate path device and if you specify the PATHs operand or the QUERY raddr command.

QUERY

PUN Card punch
GRAF Graphics device
CONS Console
DEV Any other device

PATHS

displays the path status for all alternate path devices. In multi-processor mode, the system will also display the processor address for each device.

Sysvirt

displays the addresses and status of 3330V devices that have the SYSVIRT feature.

TApes

displays the real addresses of magnetic tape units.

Response:

```
TAPE raddr CP SYSTEM
PATH raddr { OFFLINE
            { ONLINE [PROC n1[,PROC n2]] }9
```

indicates that the real tape device (raddr) is attached to CP for its exclusive use.

```
TAPE raddr ATTACH TO user id vaddr
PATH raddr { OFFLINE
            { ONLINE [PROC n1[,PROC n2]] }9
```

The system displays this response if the real tape device (raddr) is attached to a user's (user id) virtual machine at virtual address vaddr.

```
TAPE raddr status
the system displays this response for conditions that test the status of the tape device.
```

raddr
is the real device address (cuu).

status
is any available status for the tape device. The status conditions can include:

- INT REQ
- ATTACH
- FREE
- OFFLINE

⁹ Additional lines (PATH...) will only appear if you specify the PATHS operand and the device has alternate paths. Note that in systems generated for MP mode, all devices have at least a second logical channel path.

UR

displays the real addresses of unit record devices (card reader, card punches, printers).

Response:

The response that you get depends on how the UR device is used. The UR device can be owned by CP or attached to a virtual machine.

UR Devices Attached to a Virtual Machine

```

{ PRT }
{ PUN } raddr [PROC nn] ATTACH TO userid vaddr
{ RDR }
    
```

raddr
is the real device address (cuu).

userid
is the name of the spool file owner.

vaddr
is the user's virtual device address.

UR Devices that CP Owns

```

PRT raddr [PROC n1[,PROC n2]] 10
raddr status CLASS a... FORM form
raddr xsetup xxsep xxx3800 xxxxxFCB
is the response for a 3211-type printer.
    
```

```

PRT raddr [PROC n1[,PROC n2]] 10
raddr status CLASS a... FORM form
raddr xsetup xxsep xxx3800
raddr FLASH ffff IMAGE imagelib CHARS char FCB nn
xhold
is the response for a 3800 printer.
    
```

```

PRT raddr [PROC n1[,PROC n2]] 10
raddr status CLASS a... FORM form
raddr xsetup xxsep xxx3800
is the response for all other printers.
    
```

```

PUN raddr [PROC n1[,PROC n2]] 10
raddr status CLASS a... FORM form
raddr xsetup xxsep
is the response for a punch device.
    
```

¹⁰ If you specified the PATHS operand or used the QUERY raddr command, and the system is generated for MP mode, the system will add the processor address(es) to the response line as indicated.

QUERY

RDR raddr [PROC n1[,PROC n2]] ¹⁰ status
is the response for a reader device.

The operands are:

a...

specifies the classes that the output device services. An output class can service up to four classes. Do not put blanks or commas between classes.

char

is the name of the character arrangement table used for the separator page.

ffff

is the forms overlay sheet that the device contains.

form

is the operator form that the output device is processing.

image lib

is the named system image library that the system uses for tables.

nn

is the value of the forms control buffer that the system uses for the file.

raddr

is the real device address (cuu).

status

is the status of the device and can be one of the following:

DRAINED

the device is not currently available for processing. Issue a START command to activate the device.

INT REQ

the device is not ready. You must manually make it ready before spool processing continues.

MOUNT REQ

the output device is waiting for a forms mount.

PRINTING

the printer is currently printing a file.

PUNCHING

the punch is currently punching a file.

SETUP REQ
the printer is waiting for a forms mount and alignment (setup).

START REQ
tells you to start the punch if you gave the **START** command with the **NOSEP** option. This lets you remove the last punch deck before starting to punch the new job.

STARTED
is the response to the **START** command.

WAITING
the device is available for spooling activity.

xhold
can be **HOLD** or **PURGE**.

HOLD
indicates a save status for spool files that have load errors.

PURGE
indicates a deleted status for spool files that have load errors.

xsetup
is one of the following:

MANUAL
the system started the device with the **MANUAL** option.

AUTO
the system started the device with the **AUTO** option.

SETUP
the system started the device with the **SETUP** option.

xxsep
is one of the following:

SEP
the system started the device without the **NOSEP** option.

NOSEP
the system started the device with the **NOSEP** option.

xxx3800
is **NO3800**, **BEG3800**, or **ANY3800**.

QUERY

xxxxxFCB

is one of the following:

DEFFCB

the system is using the default FCB that was loaded in the printer.

FILEFCB

tells you that you can send File FCBs to the printer. File FCBs are LOAD FCB CCWs imbedded in the spool file.

CFILEFCB

tells you that you can send File FCBs to the printer. File FCBs are LOAD FCB CCWs imbedded in the spool file. After the system has finished printing each copy of the spool file containing the file FCB, you will get this message:

```
DMKRSP239I PRT cuu DRAINED. POSSIBLE
FCB-FORMS MISMATCH.
```

You will also get one of the following responses if the printer, punch, or reader is currently active with a spool file.

```
{PRT} raddr {PRINTING} USERID FILE CLASS RECORDS RECLEFT CPY DIST SEQ
{PUN}       {PUNCHING}  userid file a typ norecs noleft  nnn code seq

RDR  raddr {READING }  USERID FILE CLASS RECORDS
        {HAS READ}  userid file a RDR norecs
```

user id

is the name of the spool file owner.

file

is the spool file spoolid number.

a

is the spool file class.

typ

is the originating device type (PRT, PUN, CON).

norecs

is the total file logical record count.

no left

is the number of records left for the system to process.

nnn
is the number of copies remaining for output, where 001 indicates the last copy.

code
is the distribution code of the file.

seq
is the sequence number of the file (printed on the separator page).

Virtual
displays the addresses and status of 3330V devices that have the VIRTUAL feature.

DAsd valid
displays the active or free status of the DASD volume that you specify with *valid*.

Response:

The command response is in either the active or free format depending on the device status.

Note: If you query any real device or group of devices (such as QUERY DASD), the system will display the following message for all devices in a not-ready status:

```
type raddr INT REQ
```

DUMP
displays at your terminal the type of device and device address. If the device is DASD, DUMP displays the type of space on which is the abnormal termination dump. Allocate DASD space for dumps as either 'DUMP' or 'TEMP' space.

Response:

```
type raddr DUMP UNIT { CP } [ TEMP ]
                     { ALL } [ DUMP ]
```

The system response shows that the device of type located at raddr is the system dump unit. If the system dump is allocated on DASD, DUMP, or TEMP, the message will indicate space allocation.

```
{ Files Class c FORM form [userid] [HOLD
                             * ] [NOHold
                             SYShold
                             USERhold] }
```

displays the number of spooled input and output files. The system does not include files that it is currently processing.

Class c
selects spool files of class c for counting.

QUERY

FORM *form*

selects spool files of the form that you specify for counting. *form* is one to eight characters.

- The form number is the user form if you specify the * option.
- The form number is the operator form if you specify the *userid* option, or neither *userid* nor *.

HOLD

selects spool files that are in hold status for counting. These are the files held by system hold or user hold.

NOHold

selects spool files that are not in hold status for counting. These are the files not held by system hold or user hold.

SYShold

selects spool files that are in system hold status for counting.

USERhold

selects spool files that are in user hold status for counting.

userid

displays the spool files of the user that you specify. If you don't specify a *userid*, CP examines the spool files that all users own.

*

displays only your (or the one who issued QUERY) spool files.

Response:

FILES: { NO } RDR, { NO } PRT, { NO } PUN
 { nnnn } { nnnn } { nnnn }

displays the total number of spool files in the system, based on the options you select.

Hold

displays a list of users whose output is being held as a result of issuing the HOLD command.

Response:

HOLD : { NO } RDR, { NO } PRT, { NO } PUN
 { nnn } { nnn } { nnn }

displays the total number of files within the system that are in system hold status.

user id - $\left. \begin{array}{l} \text{ALL} \\ \text{RDR} \\ \text{PRT} \\ \text{PUN} \end{array} \right\} , \dots$

shows the type of hold (if any) for any user in the system for which HOLD is in effect. When you issue QUERY HOLD you may receive either or both of these responses, depending on the status of your spooled files.

JOurnal

indicates the status of the logon/link journaling facility as active or inactive.

Response:

JOURNAL LOGON $\left. \begin{array}{l} \text{ON} \\ \text{OFF} \end{array} \right\}$ LINK $\left. \begin{array}{l} \text{ON} \\ \text{OFF} \end{array} \right\}$

ON

shows that the relevant journaling function is active.

OFF

shows that the relevant journaling function is inactive.

Lnnn

displays the status of the logical device with address *nnn*.

Response:

LDEV nnn LOGON AS user id

shows that the logical device with address *nnn* is the virtual console for the virtual machine of user *id*.

LDEV nnn ATTACHED TO user id vaddr

shows that the logical device with address *nnn* is attached to the virtual machine of user *id* at virtual address *vaddr*.

LDEV nnn ENABLED

shows that the system created a logical device *nnn*, but no one has entered the CP DIAL or CP LOGON command through it.

Lnnn NOT LOGGED on

shows that the logical device *nnn* does not exist. The system interpreted *Lnnn* entered on the QUERY command line as a *userid*.

LOGmsg

displays the system log messages of the day.

QUERY

Response:

```
* logmsg text line 1
.
.
* logmsg text line n
logmsg additional text lines
.
.
.
```

The system displays all lines (both those with and without an asterisk) in the log message file.

MITime

displays the current time interval settings for missing interruption monitoring.

Response:

```
DASD=mm:ss GRAF=mm:ss TAPE=mm:ss UR=mm:ss MISC=mm:ss
shows the current time interval settings for DASD (FBA and direct
access), GRAF (graphics), TAPE (magnetic tape units), and MISC
(miscellaneous) devices.
```

MITIME OFF

shows that no monitoring for missing interruptions is taking place.

MISSING INTERRUPTION MONITORING NOT AVAILABLE

shows that DMKDID is not in the system.

INVALID OPTION - option

shows that you have specified an invalid parameter on the QUERY MITIME command.

Names

displays a list of all the users logged on and the real address of the line to which each is connected. The system displays the DSC, instead of the line address, of any user that is disconnected.

Response:

```
userid - {DSC }, ...
.
.
userid - {DSC }, ...
.
.
VSM userid
userid-LUNAME, userid-LUNAME,...
```

lists all logged-on users. If the user is currently connected, the system displays the real address to which he is connected (raddr). If the user is not connected, the system displays DSC.

If the user is logged on to the system with a SNA virtual console, CP is not aware of the real address of the LU (logical unit) device. In this case, CP displays the LUNAME for the user *id* instead of the real address. The system lists the userid of the VTAM Service Machine (VSM) that supports the SNA console before the console listing. The VSM and LUNAME displays follow all userid and real address listings.

Note: The logical unit is the port through which an end user can:

- Access the SNA network to communicate with another end user
- Access the facilities of the system services control point.

PAGing

displays the current system paging activity. In Basic System Extensions, this operand displays the current SET PAGING value.

Response:

PAGING SET: nn

nn

is the number the system will use in the working set size estimate control algorithm.

$$\left. \begin{array}{l} \text{Printer} \\ \text{PUch} \\ \text{Reader} \end{array} \right\} \left(\begin{array}{l} [\text{Class } c] [\text{FORM } form] \\ [userid] \left[\begin{array}{l} \text{HOld} \\ \text{NOHold} \\ \text{SYShold} \\ \text{USERhold} \end{array} \right] \end{array} \right) \left[\begin{array}{l} \text{null} \\ \text{ALL} \\ \text{TBL} \end{array} \right]$$

spoolid

displays the following types of information about printer, punch, or reader spool files:

- Userid of the spool file owner. If you are looking at files for a specific user (*userid* option), the userid indicates the originator of the spool file.
- Spool file *spoolid* number.
- Class and originating device type.
- Number of logical records in the file.

You can enter Printer as PRT, PUch as PCH, and Reader as RDR.

You can display more information using the null, ALL, or TBL operand as follows:

null

displays the following information for the requested device:

- Number of copies the user specified for the file
- File hold status
- Form name that the user assigned
- Operator form that corresponds to the user form.

QUERY

ALL

displays the following information for all spool files examined:

- Number of copies the user specified for the file
- File hold status
- Date and time that the system created the file
- Filename and filetype of the file (if any)
- Distribution code of the file.

TBL

displays the following table information for 3800 spool files:

- Flash name
- Forms control buffer (FCB)
- Name of the copy modification module
- Number of copies to be flashed (flash count)
- Where the 3800 LOAD commands are within the spool file
- Name of the character arrangement table.

Class *c*

selects spool files of class *c* for displaying.

FORM *form*

selects spool files of the form that you specify for displaying. *form* is one to eight characters.

- If you specify the * option, the form number is the user form.
- If you specify the *userid* or specify neither the *userid* or * option, the form number is the operator form.

HOLD

selects spool files that are held (either by system hold or user hold) for displaying.

NOHold

selects spool files that are not held (either by system hold or user hold) for displaying.

SYShold

selects spool files that are in system hold for displaying.

USERhold

displays spool files that are in user hold status.

userid

displays only the spool files of *userid*. If you don't specify *userid*, the system examines the spool files owned by all users.

displays your (or the person who issued QUERY) spool files.

Response:

If you specify the spoolid operand, the system only lists one file for Query Reader, Query Printer, or Query PUnch.

- If you enter Query Reader null, Query Printer null, or Query PUnch null, the system displays the following information: ownerid, file, class, records, cpy, hold, userform, and operform.
- If you enter Query Reader ALL, Query Printer ALL, or Query PUnch ALL, the system displays the following information: ownerid, file, class, records, cpy, hold, date, time, name, type, and dist.
- If you enter Query Reader TBL, Query Printer TBL, or Query PUnch TBL the system displays the following information: ownerid, file, class, records, flash, fcb, mdfy, flshc, load, and chars.

OWNERID ¹¹	FILE	CLASS	RECORDS	CPY	HOLD	USERFORM	OPERFORM			
userid	file	a typ	norecs	*nnn	stat	userform	operform			
				CPY	HOLD	DATE	TIME	NAME	TYPE	DIST
				*nnn	stat	mm/dd	hh:mm:ss	name	type	code
				FLASH	FCB	MDFY	FLSHC	LOAD	CHARS	
				ovly	plpi	cmod	ccc	{ NO } { BEG } { ANY }	ctab

userid
identifies the user who owns the file.

file
is a unique, system-assigned number that VM/SP uses to identify the file.

a
is the spool file class.

typ
is the originating device type (PRT, PUN, CON, or RDR).

norecs
is the number of logical records in the file.

***nnn**
is the number of copies for the file that you specify. (Has no effect for reader files.)

¹¹ The system alters the OWNERID heading in the title line for the spool file data to ORIGINID when you use the userid operand. In that event, ORIGINID represents the originator of the file.

QUERY

stat

is the file hold status and can be any of the following:

NONE - no hold
USER - user hold
SYS - system hold
USYS - user and system hold.

user form

is the 1-8 character form name that the user assigns.

oper form

is the 1-8 character operator form that corresponds to the user form.

mm/dd

is the date that the system creates the file in month/day.

hh:mm:ss

is the actual time when the system creates the file in
hours:minutes:seconds.

name

is the filename assigned to the file (if any). If the file has a
24-character data set name (dsname), the system only displays 18
characters. These characters extend from the name field through the
type field.

type

is the filetype assigned to the file (if any).

code

is the distribution code of the file.

ovly

is the 1-4 character name of the flash overlay sheet loaded into the
3800 printer.

p lpi

is the print lines/inch value that the system will use for the separator
page.

cmod

is the 1-4 character name of the copy modification module that the
system will use to print the file.

ccc

is the flash count for the file.

{ NO }
 { BEG }
 { ANY }

is where the 3800 LOAD commands are positioned within the spool file:

- NO - not at all
- BEG - at the beginning
- ANY - interspersed throughout the spool file.

ctab

is the 1-4 character name of the character arrangement table that the system will use for the separator page.

PRIORity *userid*

displays the current priority of *userid*. The VM/SP directory establishes this priority. However, you can override it with a SET PRIORITY *nn* command.

Response:

user id PRIORITY = nn

nn

is the the assigned priority of the user that you specify. The lower the value, the higher the priority.

PROCEssr

displays the online processors in the system.

Response:

PROCESSOR nn ONLINE, [PROCESSOR nn ONLINE, [SYSTEM IN AP/MP MODE]]

is a display of the processors that are online. In an attached processor environment, the first processor is the main processor. In an MP system, the first online processor that the response specifies is the IPL processor.

QDROP

displays the userids that have QDROP set off. See the class A SET command for a description of the QDROP performance option.

Response:

QDROP SET OFF FOR: user id1 [+(USERS)], user id2 [+(USERS)],

...

The values, user id1 and user id2, are userids that have the QDROP OFF option set for them. You may get more than one line of output. +(USERS) shows that the system is extending the option to the virtual machines communicating with the displayed userid.

QDROP SET OFF FOR: NO USERS

This is the response if no userids have QDROP set off.

QUERY

raddr1 [-*raddr2*]

displays the device status at the address you specify. If you specify a range of addresses, the system displays the status of all devices in the range. If *raddr1* or any address in the range *raddr1-raddr2* is the address of a 3800 printer, the system displays the table information unique to the 3800, in addition to the current status of the device.

Response:

This response depends on the type of device located at the *raddr*. See the QUERY DASD, TAPES, UR, GRAF, and LINES responses.

When you issue QUERY *raddr* and *raddr* is a 37xx, the response is in the form:

```
CTLR raddr DEV xxoo PGM = {nnnnnnnn} DUMP = {OFF }
                          {EP NAME}      {AUTO }
```

CTLR

is the keyword for the Communications Controller.

raddr

is the real address of the controller in the cuu form.

DEV

is the keyword DEVICE.

xxoo

tells which device *raddr* is relative to the CLUSTER macro.

PGM=

is the program loaded into the 37xx (nnnnnnnn) by VM. This area will be blank if VM didn't load the 37xx. If VM loads the 37xx, this area will contain the name of the EP that VM loaded.

DUMP=

tells the user if an automatic dump (AUTO) was requested or if no dump will be performed (OFF) in case of a failure.

S370E

displays the status of MVS/System Extensions support. In attached processor mode, the system reports the state (enabled or disabled) of MVS/System Extensions support for each processor.

Response:

```
S370E {ON }
      {OFF }
```

ON or OFF

shows whether MVS/System Extensions support is active or inactive (not available) on the system.

S370E { ON } PROC nn, { ON } PROC nn
 { OFF } { OFF }

ON or OFF

shows whether MVS/System Extensions support is enabled or disabled (not available) on the specified processor, PROC nn.

nn

can be any value from 0-63.

SASsist

displays the following:

- The current status of the virtual machine assist for the VM/SP system
- The current status of the virtual interval timer assist portion of ECPS:VM/370.

Response:

SASSIST { ON }
 { OFF }

ON or OFF

shows if the virtual machine assist (and the virtual interval timer assist portion of ECPS:VM/370 - if the processor is equipped this way) is active or inactive (or not available on the system).

For Query SASsist (attached processor and multiprocessor applications)

SASSIST { ON } PROC nn, { ON } PROC nn
 { OFF } { OFF }

ON or OFF

shows if the virtual machine assist (and the virtual interval timer assist portion of ECPS:VM/370 - if the main processor or the attached processor is equipped this way) is active or inactive for the designated processor, PROC nn.

nn

can be any hexadecimal value from 00 to 3F. Your installation assigns values to the processors when the system hardware is installed.

Note: If you get a positive response to both the QUERY CPASSIST and QUERY SASSIST commands, the expanded virtual machine assist portion of the Extended Control-Program Support:VM/370 is active.

SPMODE

displays the status of the single processor mode of operation (on or off).

QUERY

Response:

SPMODE { ON }
 { OFF }

ON or OFF

shows if the single processor mode of operation is on or off.

SRM { APAGES
 DSPSLICE
 IB
 MAXDRUM
 MAXWSS
 MHFULL
 PB
 PCI
 PGMSTAT
 PGMTLIM }

displays the current status of the system's internal parameters.

APAGES

displays the number of available pageable pages.

Response:

AVAIL. PAGES = nnnn

nnnn

is the number of available paging pages.

DSPSLICE

displays the duration of the dispatching time slice.

Response:

TIME SLICE = nnn MILS

nnn

is the size of the dispatching time slice in milliseconds.

IB

displays the current interactive shift bias value. The system uses this value for Q1 virtual machines that are using less than their allocated share of the processor.

Response:

INTACT. SHIFT = nnnn

nnnn

is the current interactive shift bias value.

MAXDRUM

tells if the maximum drum page allocation limit has been set and what the limit value is.

Response:

DRUM MAX = nnnn x

nnnn
is the maximum drum page allocation.

x
is D if the system set it dynamically and C if you set it with a command.

DRUM MAX NOT SET

shows that the system has not set a value for the maximum drum page allocation.

MAXWSS

tells if the maximum page allocation set estimate has been set and what the limit value is.

Response

WSS MAX = nnnn x

nnnn
is the maximum working set estimate.

x
is D if the system set it dynamically and C if you set it with a command.

WSS MAX NOT SET

shows that the system has not set a value for the working set estimate maximum.

MHFULL

displays either the moveable head page migration limits (nnn%) or that the function is shut off (OFF).

Response:

MHFULL { nnn }
 { OFF }

nnn
is a value from 001 to 100 that shows the percent of available MH page space to be used before migration takes place for pages in the preferred paging area.

PGMTLIM

displays the unused segment elapsed time criteria for the page migration routine to use.

Response:

TIME LIMIT = nnnn

nnnn

is the unused segment of elapsed time criteria for the page migration routine to use.

STATUS raddr

displays the status of the DASD that has the specified real address.

Note: When you issue the Query STATUS command, be sure the Missing Interrupt Handler (MIH) is enabled and monitoring for missing interrupts. If the MIH is disabled, the system will not detect the missing interrupt condition if it occurs upon executing the Query STATUS command. Refer to the *VM/SP System Programmer's Guide* under "Interruption Handling" for information about the Missing Interrupt Handler.

Response:

DEVICE raddr OPERATIONAL
device is online and ready.

DASD raddr NOT OPERATIONAL
shows one of the following:

- The DASD at address raddr is powered off.
- The control unit that the device is attached to is powered off.
- The control unit is logically removed from the channel.

DEVICE raddr OFFLINE
when the system checks the device status and finds that the device is disabled or offline.

UNABLE TO COMMUNICATE WITH THE DEVICE raddr
reports that the specified device or control unit is busy for a prolonged period.

DEVICE raddr EQUIPMENT CHECK, SENSE=sense data
the system has done an equipment check and shows the 24 or 32 bytes of sense data. Although this is an "informational" type of message, you may need to intervene.

QUERY

`raddr CHANNEL ERROR`
is sent to you (the class B user) when one of the following occurs:

- A missing interrupt
- A channel error (channel data check, channel control check, or interface control check).

`DEVICE raddr INTERVENTION REQUIRED`
when the device is not in a “ready” condition. For example, one of the following could occur:

- The device is not mounted properly.
- The device is logically removed from an operational control unit.

`DASD raddr NOT READY`
when the system checks the device status and finds that the device is not ready due to a previous hardware check that has not been fixed.

STORage

displays the size of real storage.

Response:

`STORAGE = xxxxxK`
displays the size of real storage (xxxxx) in multiples of 1024 bytes.

SYStem raddr

displays the following:

- The userid, virtual address, and access mode of virtual disks that reside on the specified channel for the particular device at `raddr`
- The control unit address `raddr` of logged-on users.

Response:

You use this command to request the number of user minidisks residing on the physical disk located at `raddr`. The system gives the response for each minidisk in the following format.

```
userid vaddr mode, ...
```

```
·   ·   ·  
·   ·   ·  
·   ·   ·
```

`userid`

identifies the users who have links to minidisks on `raddr`.

`vaddr`

is the virtual address by which the user refers to the minidisk.

mode

is the type of access the user has:

- R/O
- R/W
- nnn for the number of cyls/blks of TDSK space defined by users with the DEFINE command. nnn is also the number of cylinders of TDSK space defined on a count-key-data device.
- nnnnnn for the number of blocks of TDSK space defined on an FBA device.

TDsk

displays the currently defined temporary disk space (TDSK) from all available system-owned volumes assigned to virtual machine users.

Response:

This command displays all the currently defined user TDSK space from all available system-owned volumes. The system produces an entry of the following format for each TDSK:

```
user id vaddr {nnn|nnnnnn}
```

user id

identifies the user.

vaddr

is the user's virtual device address.

nnn

is the number of cylinders that the user has defined. This operand is also the number of cylinders allocated for count-key-data devices.

nnnnnn

is the number of blocks defined for FBA devices.

Note: If no user TDSK space is currently defined, you will get this message:

```
DASD TDSK NOT FOUND
```

userid

if the user is logged on, *userid* displays the userid and device address of the user's terminal. If the user is not logged on, you will get a message.

Response:

```
user id - raddr
```

displays the real address (raddr) to which the specified user is connected.

QUERY

`userid - luname`

when you specify `QUERY USERS userid` and the `userid` is logged on with a SNA virtual console. Since CP is not aware of the real device address (`raddr`) of the LU device, it supplies the `LUNAME` in place of the real address.

Users

displays the number of logged on users and the number of users dialed to other virtual machines. If you specify `userid` and the user is logged on, the system will display the `userid` and device address of his or her terminal. If the user that you specify is not logged on, you will get a message.

Use the `USERS` operand if the `userid` you wish to specify is the same as an operand (or its minimum truncation) of the `QUERY` command.

Note: The number of users logged on as shown by the `NAMES` operand may differ from the number logged on as shown by the `USERS` operand. The number of users in the process of logging on and logging off accounts for this difference.

Response:

`nnn USERS, mmm DIALED, xxx NET`

`nnn`

is the total number of logged on users.

`mmm`

is the total number of users logically attached with the `DIAL` command to virtual machines.

`xxx`

is the number of users currently logged onto the SNA network.

Note: `DIALED` means that the line is not available to CP because it is logically attached to a logged-on user and is a part of that user's virtual machine operation.

QVM

Privilege Class: A

When to Use QVM

Use this command to change a V=R machine to run in native mode, instead of under VM/SP. Native mode refers to running an operating system stand alone on the real machine.

QVM Format and Operands

QVM	<i>userid</i> [NORETURN]
-----	--------------------------

userid

identifies the V=R machine that you want to run in native mode.

When you transfer control of the operating system from under VM/SP to native mode, and VM/SP is running in uniprocessor mode, you can issue this operand to give the V=R machine control of the processor. Refer to *Running Guest Operating Systems* for more information.

NORETURN

indicates that you do not want to return from native mode to the VM/SP environment. If you do not specify NORETURN, the system will let you change from native mode back to VM/SP.

Things You Should Know about the QVM Command

1. Before you pass control to the V=R guest machine to run in native mode, vary offline all 3480 tape drives that the guest machine will use. After the Control Program gains control again, vary back online the tape drives that you want to use.
2. When an SCP transfers control to the V=R virtual machine in native mode, error records for the SCP are in two locations. For details on finding all the error records that pertain to the SCP, see *VM/SP OLTSEP and Error Recording Guide*.

REPEAT

REPEAT

Privilege Class: D

When to Use REPEAT

The REPEAT command applies to files that the system is currently writing to a real punch or printer. Use this command to:

1. Request more copies of an output file than that requested by the file's creator.
2. Place the current output file in a HOLD status, with or without increasing the number of copies that you want the system to create.

REPEAT Format and Operands

REPEAT	$raddr \left[\begin{array}{l} [nnn] \\ [1] \\ \{nnn\} \end{array} \right] \text{ HOLD}$
---------------	--

raddr

is the address of the output device (printer or punch) whose current active spool file you want the system to repeat or hold.

nnn

is the number of extra copies of the current file that you want the printer or punch to make. If you do not specify *nnn* or the HOLD operand, the number of extra copies defaults to 1. The maximum value that you can specify for *nnn* is 255.

HOLD

holds the file that the system is currently printing or punching. When the system finishes the the last copy, it places the file in system hold status. The file will remain held until you explicitly release it.

Note: Before the HOLD operand takes effect, the system prints all copies that a SPOOL command or a previous REPEAT command indicated.

System Response

```
{ PRT } raddr { REPEATED } USERID FILE CLASS RECORDS RECLEFT CPY DIST SEQ  
{ PUN }      { FILE HELD } userid file a typ norecs  noleft  nnn dist-seq  
                                         code
```

raddr

is the real address of the device whose current output the system is repeating.

userid

identifies the user who will receive the output.

file

is the unique, spoolid number that the system assigns to refer to the file.

a

is the spool file class.

typ

is the originating device type (PRT, CON, or PUN).

norecs

is the total number of logical records in the file.

noleft

is the number of records that the system has left to process.

nnn

is the number of copies of the file that the system has left to create. 001 is the last copy.

distcode

is the distribution code of the file.

seq

is the sequence number of the file (printed on the separator page).

SAVESYS

SAVESYS

Privilege Class: E

When to Use SAVESYS

Use this command to save named systems that your installation has defined in the system name table (DMKSNT). Invoke SAVESYS immediately after loading the system that you want to save (while the system to be saved is in a stopped state). This will cause CP to save the virtual machine storage with register contents, PSW, and storage keys as they currently exist.

SAVESYS Format and Operands

SAVESYS	<i>systemname</i>
----------------	-------------------

systemname

is a predefined name in the system name table (DMKSNT) that defines the installation requirements for the named system. The definition indicates:

- The number of pages to be saved
- The DASD volume on which the system is to be saved
- The shared segments (if any).

For more information about saved systems, refer to the discussion of named systems in the *VM/SP Planning Guide and Reference*.

System Response

SYSTEM SAVED

SET

Privilege Classes: A, B, E, or F

When to Use SET

Use the SET command to establish system parameters. The SET command performs functions to control the CP system and virtual machine options.

SET Formats

Class A SET Format

SET	AFFinity	userid	[ON OFF nn]
	CPAssist		{ON OFF} [PROC [nn]]
	FAVORed	userid	[nnn OFF]
	JOURNAL ¹	{LOgon LInk}	[ON OFF]
	PRIORity	userid	nn
	QDROP	userid	{ON OFF} [USERS]
	REServe	userid	{nnn OFF}
	SASsist		{ON OFF} [PROC [nn]]
	S370E		{ON OFF} [[PROC] addr]

¹The JOURNAL operand is valid only if STQUERY=YES is specified in the SYSJRL macro instruction DMKSYS.

SET

Class B SET Format

SET	$\left(\begin{array}{l} \text{DUMP } \left\{ \begin{array}{l} \text{AUTO} \\ \text{raddr} \end{array} \right\} \left[\begin{array}{l} \text{CP} \\ \text{ALL} \end{array} \right] \\ \text{LOGmsg } \left[\begin{array}{l} \text{nn} \quad \text{text} \\ \text{NULL} \end{array} \right] \\ \text{MITime } \left\{ \begin{array}{l} \text{class} \quad \left\{ \begin{array}{l} \text{mm:ss} \\ \text{OFF} \end{array} \right\} \\ \text{OFF} \end{array} \right\} \left[\begin{array}{l} \text{class} \quad \left\{ \begin{array}{l} \text{mm:ss} \\ \text{OFF} \end{array} \right\} \dots \end{array} \right] \end{array} \right)$
------------	---

Class E Command Format

SET	$\left(\begin{array}{l} \text{PAGing } \text{nn} \\ \text{SRM } \left(\begin{array}{l} \text{APAGES } \text{nnnn} \\ \text{DSPSlice } \text{nnn} \\ \text{IB } \text{n} \\ \text{MAXDrum } \left\{ \begin{array}{l} \text{nnnn} \\ \text{OFF} \end{array} \right\} \\ \text{MAXWss } \left\{ \begin{array}{l} \text{nnnn} \\ \text{OFF} \end{array} \right\} \\ \text{MHFULL } \left\{ \begin{array}{l} \text{nnn} \\ \text{OFF} \end{array} \right\} \\ \text{PB } \text{nn} \\ \text{PCI } \left\{ \begin{array}{l} \text{DRUM} \\ \text{DISK} \end{array} \right\} \\ \text{PGMTlim} \end{array} \right) \end{array} \right)$
------------	--

Class F Command Format

SET	RECORD { OFF ON <i>raddr</i> LIMIT <i>nn</i> BYTE <i>nn</i> BIT <i>n</i> [{AND} BYTE <i>nn</i> BIT <i>n</i>] } MODE { RETRY } { Quiet } [<i>cpuid</i>] { MAIN } { Record }
------------	--

SET Operands (Class A,B,E,F)

AFFinity *userid* [ON
OFF
nn]

sets affinity on or off for the *userid* that you specify (attached processor and multiprocessor systems only). AFFINITY lets the virtual machine execute only on the processor that you specify; however, AFFINITY does let CP code execute for a virtual machine on any processor.

userid

is the user for whom you want the system to set or reset affinity. If you don't specify a *userid*, the system sets or resets your affinity (the user who issued the command).

ON

sets affinity on, using the processor address in the user's directory entry.

OFF

sets affinity off.

nn

the hexadecimal (00-3F) processor address of an installed processor of an attached processor system. If you specify this option, the system uses this processor address to set affinity on.

CPAssist { ON } [PROC [*nn*]]
{ OFF }

sets CP assist to the active or inactive state.

If CP assist is active, functions that CP routines normally execute are executed by processor hardware logic. This function made active on VM/SP initialization, a part of Extended Control-Program Support:VM/370, is available on most VM/SP-supported processors.

SET

ON

activates CP assist for the processor. If you want the same setting for main and attached processors, use this option without the PROC *nn* option.

This option also controls the expanded virtual machine assist.¹² Use this option along with SET SASsist ON to enable this function.

OFF

deactivates CP assist for the processor. If you want the same setting for main and attached processors, use this option without the PROC *nn* option.

This option also controls the expanded virtual machine assist.¹² Use this option or SET SASsist OFF to set this function to the inactive state.

PROC *nn*

activates CP assist in 3031 attached processor applications. PROC *nn* activates CP assist on the desired main or attached processor. The value *nn* is the hexadecimal processor address (00-3F) expressed in decimal (0-63).

DUMP { AUTO } { CP }
 { *raddr* } { ALL }

identifies the unit to receive the VM/SP system ABEND dump.

AUTO

specifies that the system dump unit is disk.

raddr

is the address of a real printer, 9-track tape address, or a 3480 address. If the dump unit is a tape drive, the dump data must fit on one reel because VM/SP does not support dumping to multiple tape volumes.

ALL

dumps all storage.

CP

dumps only storage that the control program is occupying.

Note: If the dump unit is a tape or a printer, all 3480 tape drives online at the time of the dump will remain assigned to the system. To unassign the drives, IPL VM/SP after the dump completes, and vary the devices offline.

¹² The expanded virtual machine assist is another part of ECPS:VM/370. The system sets this virtual machine assist to the active state when VM/SP is initialized.

FAVORed *userid* [*nnn*]
[OFF]

sets the favored execution performance option.

userid

is the user for whom you specified the option. You can favor more than one user; however, you can only designate one *userid* in a single SET command.

nnn

is a percentage value from 1 to 100. When you specify this option, VM/SP tries to provide that percentage of processor time to the virtual machine, if the virtual machine can fully utilize the processor time.

At regular time intervals, the VM/SP dispatcher checks the processor time used by the particular virtual machine; if the percentage has been exceeded, the virtual machine is given his normal priority usage for the remainder of the interval. If the percentage is low, the virtual machine is assigned highest priority for the remainder of the interval.

Notes:

1. If you specify 100% for a virtual machine, the system places the virtual machine at the top of the dispatch queue and holds it there until it logs off. Because there can be only one user at the head of the queue, do not specify 100% for more than one virtual machine.
2. If the total amount of percentage times that you assign to a number of virtual machines exceeds 100%, the effect it will have on interactive (Q1) response is unpredictable.

OFF

the system removes the specified virtual machine from favored status.

Note: You can only use this option with percentage *xxx* for one virtual machine at any time.

If you don't specify *xxx* or OFF, the system will give preference to the specified virtual machine over other users with the same priority anytime the virtual machine is ready to use processor time.

Note: SET FAVORed *userid xxx* does not designate the *userid* as always dispatchable; to do this, issue another SET FAVORed command without a percentage (*xxx*) value.

JOurnal { LOGON } [ON]
 { LINK } [OFF]

sets the status of the LOGON and LINK journaling functions.

SET

LOGmsg [*nn* *text*]
[NULL]

nn

specifies the line number of the log message that you want to add or change. You can supply the text of the log message line on the command line after this line number.

If you do not supply the text, the system will respond with:

LOGMSG:

Enter the text here. If you do not enter a line, CP will delete log message line *nn*. If you do enter text, CP will replace log message line *nn* with the new text. If log message line *nn* does not exist, CP will add your text as a new log message line after the current last line in the log message.

text

specifies the contents of the *n*th log message line. If you specify *text*, CP will not perform a console read.

NULL

deletes the entire existing LOGMSG. If you enter more SET LOGMSG commands, CP will start to create a new LOGMSG.

Notes:

1. To stop the LOGMSG entry, enter a null line.
2. The maximum log message length per line is 76 characters, including spaces.
3. If you precede the message text with an asterisk, CP will automatically display the message on the user's terminal as soon as he/she logs on.

MITime { *class* {*mm:ss*} [*class* {*mm:ss*} ...] }
{OFF}

alters the current missing interrupt monitoring status in one of the following ways:

- Changes the missing interrupt monitoring time interval in DMKSYS for a specific device class (*class*)
- Stops missing interrupt monitoring for the device class that you specify (*class* OFF)
- Stops all monitoring of missing interrupts for all device classes (MITIME OFF).

CP only changes the time intervals for device classes that you specify on the command line. All other time intervals remain unchanged.

class

refers to one or more of the following:

Class	Meaning
DASD	Direct access and FBA devices
GRAF	Graphics devices (local only)
TAPE	Magnetic tape devices
UR	Unit record devices
MISC	Miscellaneous device

Notes:

1. Do not specify the same device class more than once on the same command line.
2. GRAF refers to devices generated by CLASGRAF, except for TYP328X and TYP1053 printers.
3. UR refers to devices generated as CLASURI and CLASURO, except for TYP3800 and TYP3289E printers
4. MISC refers to Mass Storage devices and 1053, 328X, 3289E, and 3800 printers.

mm:ss

is the time interval in minutes and seconds. The maximum value for *mm* is 99 and the maximum for *ss* is 59.

Note: If you set the time interval for a device class below its default value, you may cause unnecessary missing interrupt handler processing for devices that are functioning properly. The device class default time interval values are in the "Interruption Handling" section of the *VM/SP System Programmer's Guide*.

OFF

If you issue SET MITime *class* OFF, the system stops monitoring missing interrupts for the specified device class for the duration of the IPL or until you issue another SET MITIME command. SET MITime OFF stops monitoring missing interrupts for ALL device classes.

MODE {RETRY} {Quiet} [*cpuid*]
 {MAIN} {Record}

sets the error recording mode for soft errors. Soft errors are recoverable processor storage or instruction errors corrected by hardware or software logic.

RETRY QUIET

turns off error recording for soft machine check errors successfully corrected by instruction retry.

RETRY RECORD

resets the error threshold count to zero and records up to 12 soft machine check errors, successfully corrected by instruction retry. The system records the errors on the error recording cylinder.

MAIN QUIET

turns off error recording for single bit main storage errors successfully corrected by Error Correction Code logic.

MAIN RECORD

resets the error threshold count to zero and records up to 12 successfully corrected main storage errors. The system records these errors on the error recording cylinder before entering MAIN QUIET mode.

Note: SET MODE MAIN is invalid for 3031, 3032, and 3033 processors.

cpuid

in attached processor mode, *cpuid* is the address of the processor for which you executed the command. Valid hexadecimal values are 00 to 3F. If you don't specify an address, the system processes the command for both processors.

PAGing *nn*

is the number that you want the system to use in the working set size estimate control algorithm.

PRIORity *userid nn*

sets a priority of *nn* to the user that you specify.

userid

identifies the user for whom you want to set a priority.

nn

sets the user's dispatching priority in relation to other users in the system. Generally, the lower the value of *nn*, the more favorable the user's position is in relation to other users in VM/SP's dispatch queues.

**QDROP *userid* { ON }
 { OFF } [USERS]**

controls the QDROP elimination option. QDROP ON is the initial setting for all users.

A virtual machine may enter an idle wait state with no active high speed I/O outstanding for it. In this case, VM/SP scans the virtual machine's page and segment tables, and puts resident pages on the flush list. Pages on the flush list are the first candidates for page-out or removal from main storage.

userid

identifies the user for whom you want to set the QDROP option.

ON

sets QDROP elimination on for the user that you specify (*userid*).

OFF

eliminates the scanning of the virtual machine's page and segment tables, and the placing of resident pages on flush list. This can improve performance on some virtual machines.

For example, the VM/VTAM service machine waits for the system to send IUCV messages to it. Each time the machine processes a message and enters the wait state, the system must scan the page and segment tables. If the message rate is moderate to high, the system does not need to scan, because the VM/VTAM service machine is not idle for an extended period of time. Setting QDROP OFF for the VM/VTAM service machine eliminates the overhead. If you set QDROP OFF and set the redisplay timer for the VTAM service machine to zero, the system may delay the command output for the VTAM service machine user until the user starts the next I/O operation.

If you set QDROP OFF for a virtual machine that is idle for extended periods of time, you may degrade system performance. This is because the system does not remove the virtual machine's resident pages from storage as quickly as if you set QDROP ON.

USERS

QDROP OFF USERS provides a temporary extension of the QDROP OFF status to any virtual machine communicating with VMCF or IUCV to the (server) virtual machine for which QDROP OFF is being specified. USERS eliminates overhead with a virtual machine being queue dropped during communications with a server machine for which the QDROP OFF specification is in effect. This can occur in small systems where there is a high degree of inter-virtual machine communications.

The QDROP status for the served virtual machine remains in effect only while messages are outstanding between it and the server machine. Thus, performance gains can be realized in systems with heavy usage of products such as IFS or PVM (invoked with the CMS PASSTHRU command). A system where PVM is invoked with CP DIAL or with the SNA VTAM service machine will not gain in performance, since the machine communicates with CP rather than another virtual machine.

RECORD

$$\left\{ \begin{array}{l} \text{OFF} \\ \text{ON } raddr \text{ LIMIT } nn \text{ BYTE } nn \text{ BIT } n \left[\begin{array}{l} \text{AND} \\ \text{OR} \end{array} \right] \text{ BYTE } nn \text{ BIT } n \end{array} \right\}$$

sets the recording mode for a device.

SET

OFF

turns off intensive recording mode. If you do not enter SET RECORD OFF, the system stops intensive recording after it accumulates ten error records for that device.

ON

turns on intensive recording mode for the device *raddr*.

LIMIT *nn*

specifies the number of test conditions that must occur before the system will record it. (Define the test condition with the BYTE or BIT operands.)

BYTE *nn* or BIT *nn*

defines the test condition in the sense bytes.

AND or OR

lets you test two bits. Note that you can only have one device at a time specified with intensive recording.

**REServe *userid* { *nnn* }
 { OFF }**

sets the reserved page performance option for the user that you specify (*userid*).

userid

identifies the user for whom you want set the reserved page performance option. You can only apply this option to one virtual machine at a time.

nnn

is the number of page frames that you reserve for the virtual machine's exclusive use. If the number of page frames you assign does not equal or surpass the most active pages, performance of the virtual machine will degrade.

The virtual machine can use other available page frames. The page frames that you reserve are not locked, but no other virtual machine can use a reserved page frame unless the system has exhausted all other available page frames. The size of the virtual machine determines the maximum number of page frames that you can reserve.

OFF

returns the reserved pages to the system.

SASsist {ON } [[PROC] *nn*]
{OFF }

ON

activates the virtual machine assist ¹³, a facility available on most VM/SP supported System/370 processors, for the entire system. If you want the same setting for main and attached processors, use this option without the PROC *nn* option.

This option along with CPAssist also helps activate the Extended Control-Program Support:VM/370.¹⁴ Refer to the *VM/SP System Programmer's Guide* for more information.

At system initialization, the default value SET SASsist is ON for VM/SP processors except the System/370 Model 155 II and the 165 II.

OFF

deactivates the virtual machine assist. ¹³ If you want the same setting for main and attached processors, use this option without the PROC *nn* option.

This option along with CPAssist also helps deactivate the Extended Control-Program Support:VM/370. ¹⁴

Note: SET SASsist ON or OFF also controls if the virtual interval timer assist function of ECPS:VM/370. This assist function is a hardware updating facility for the interval timers of virtual machines, (virtual location X'50') is available to users. Use of this timer hardware update facility by the virtual machine is controlled by the class G command SET ASSIST TMR or SET ASSIST NOTMR. For details, see the *VM/SP CP Command Reference for General Users*.

PROC *nn*

activates SASsist in the processor that you define with *nn* (decimal value 00 to 63). If *nn* is a value that is not the address of either processor, the system issues an error message.

SRM

sets the system resource management function.

¹³ Virtual machine assist increases the efficiency of CP processing for certain instructions, privileged instructions, and interrupts.

¹⁴ Extended Control-Program Support:VM/370 simulates the CP processing of additional privileged instructions and extends the level of handling of other privileged instructions not covered by virtual machine assist. For details on the virtual machine assist and the Extended Control-Program Support:VM/370, refer to the *VM/SP System Programmer's Guide*.

APAGES *nnnn*

the current number of pageable pages available. *nnnn* can be from 0000 to 9999.

DSPSlice *nnn*

the duration, in milliseconds (*nnn*), of the dispatching time slice.

IB *n*

is the interactive shift bias that the system will use for Q1 virtual machines that are using less than their allocated share of the processor. *n* can be from 0 to 9.

MAXDrum { *nnnn* }
 { OFF }

is the maximum number of drum pages that you want the system to allocate to any one virtual machine (*nnnn*). If you specify OFF, the system will shut off the MAXDrum. *nnnn* can be from 0000 to 9999.

MAXWss { *nnnn* }
 { OFF }

is the maximum value for a virtual machine's working set (*nnnn*). If you specify OFF, the system will shut off the MAXWss. *nnnn* can be from 0000 to 9999.

MHFULL { *nnn* }
 { OFF }

is the percentage of the preferred moveable head paging area that the system must be using before migration from this area takes place. If you specify OFF, the migration of pages will take place from fixed head areas only (OFF).

PB *nn*

indicates the maximum page bias that the system can calculate. *nn* can be from 00 to 99.

PCI { DRUM }
 { DISK }

indicates the PCI flag for 2305 and fixed head page requests. DRUM is the standard default mode; the system only flags requests for sector 1. DISK flags all chained page requests.

PGMTlim

is the unused segment elapsed time criteria that the page migration routine will use.

S370E { ON } [[PROC] [*addr*]]
 { OFF }

enables or disables MVS/System Extensions support for virtual machines; thus allowing an MVS system, running in a virtual machine, to use the MVS/System Extensions Program Product (Program No. 5740-XE1).

ON

enables the MVS/System Extensions support for virtual machines. In attached processor mode, the system enables both processors unless you specify a processor address.

OFF

disables the MVS/System Extensions support for virtual machines. When you specify OFF, the system issues a message and enters the console function mode.

PROC nn

is the address of the processor (0-63). In attached processor or uniprocessor mode, the system issues a message if the processor that you specify is unavailable or invalid.

SET S370E is invalid if you are running a VM system in a virtual machine.

Things You Should Know about the SET Command

1. In attached processor mode, if you specify one processor, the change will not affect MVS/System Extensions support users with affinity established on the other processor.
2. If the dump allocation has defaulted to the printer, when dumps are purged from DUMP DASD allocated space, CP will reallocate the dump to DUMP or TEMP space.

System Response

When you issue SET, the system displays informational messages relating to the operands that you specify. These messages are described in the *VM/SP System Messages and Codes*.

SYSTEM AUTO DUMP RE-ALLOCATED TO DASD raddr

indicates that the system reallocated the CP dump DASD space described in "Things You Should Know about the SET Command."

UNABLE TO ALLOCATE SYSTEM AUTO DUMP

indicates that CP could not find the DASD space that it needs to allocate the system.

INVALID OPTION - option

or

OPERAND MISSING OR INVALID

means that you specified an invalid parameter on the SET MITIME command.

MISSING INTERRUPTION MONITORING NOT AVAILABLE
indicates that DMKDID is not in the system.

Note: Your installation may remove the missing interruption handler module (DMKDID) from the load list if you do not want missing interruption monitoring.

SHUTDOWN

Privilege Class: A

When to Use SHUTDOWN

Use the SHUTDOWN command to:

- End all VM/SP functions, disable communication lines, checkpoint the system for an eventual warm start, and save virtual machines that are enabled to be saved.
- Automatically do a warm start, when you specify the REIPL option.

Using SHUTDOWN is the normal method to shut the system down. However, if you want to stop communications on 370x or remote 3270 binary synchronous lines, use the NETWORK SHUTDOWN command.

SHUTDOWN Format and Operands

SHUTDOWN	[REIPL POWEROFF]
----------	-----------------------

stops the system, and the CPU enters a wait state of 8.

REIPL

stops the system and does an automatic warm start. All online 3480 tape devices remain assigned to the system.

When you don't specify the REIPL option, the system rewinds, unloads, and unassigns all online 3480 tape devices from the system. The system then stops and enters a wait state.

POWEROFF

is only allowed on 4361 processors. When you specify this parameter, a normal shutdown will take place and the processor will power off. If the processor has no poweroff feature, the system will load a wait state 007 PSW after it has saved the system warm start data and virtual machines.

Upon power on of the processor, you will not need to intervene during IPL if the 4361 has the Auto Start feature installed and the warm start data is valid. During IPL, the system uses the battery operated TOD clock setting and a warm start will take place. (The TOD clock provides an accurate measure of

SHUTDOWN

time, independent of system events or activities.) If the 4361 does not have the Auto Start feature installed, you will have to re-IPL the system using the steps in "System Initialization" on page 2-3.

Things You Should Know about the SHUTDOWN Command

1. Because the SHUTDOWN command stops all virtual machine operation, use the appropriate LOGMSG information, messages and warnings to tell all virtual machine users of the impending shutdown.
2. In installations that use the 3850 Mass Storage System, the VM/SP shutdown process issues a SUSPEND order to all 3851 devices. This suspends the MSC interface. The SUSPEND order may cause the shutdown process to take longer than usual, because it causes the destaging of data on 3330V volumes.
3. During processing, if the device has an outstanding intervention required condition, the system will not log a Miscellaneous Data Record (MDR) for a device that maintains hardware environmental counters.

System Response

SYSTEM SHUTDOWN COMPLETE
indicates that the system completes the shutdown.

You won't get a response when you issue SHUTDOWN REIPL.

For a normal SHUTDOWN, you will enter a wait state of 8. When you issue SHUTDOWN POWEROFF for a 4361 that does not have the battery TOD clock, it will shutdown and poweroff. However, when you IPL the 4361 again, you will need to use the traditional IPL procedures.

SPACE

Privilege Class: D

When to Use SPACE

Use the SPACE command to force a printer to single space the output for the current active spool file. The command will override any carriage control commands that are in the actual file.

SPACE Format and Operands

SPAcE	<i>raddr</i>
-------	--------------

raddr

is the real address (cuu) of the printer whose output you want to single space.

Note: For buffered printers, such as the 3800 printer, the SPACE command does not affect files that have already been sent to the printer.

System Response

None.

SPMODE

SPMODE

Privilege Class: A

When to Use SPMODE

Use SPMODE when you want to restrict VM/SP to run in a uniprocessor environment in a multiprocessor configuration, while letting an MVS V=R virtual machine control the other processor. SPMODE can establish or reset the single processor mode environment.

Note: Before entering the SPMODE command, review its operating procedures in *Running Guest Operating Systems*.

SPMODE Format and Operands

SPMode	{ ON } { OFF }
--------	-------------------

ON

initializes VM/SP in single processor mode if all the proper conditions are met. SPMODE ON will only execute when VM/SP is running in uniprocessor environment.

Note: Before issuing SPMODE ON, you may need to VARY OFFLINE PROCESSOR *nn* if the system is operating in attached processor mode.

OFF

resets VM/SP to the normal uniprocessor mode of operation.

Things You Should Know about the SPMODE Command

1. When in single processor mode, error records for the MVS V=R virtual machine are in two locations. For details concerning the collection of all error records see *VM/SP OLTSEP and Error Recording Guide*.
2. Before the MVS native processor can use a 3480 tape drive in SPMODE, vary the drive offline from the CP system. After SPMODE is turned off, vary back online the tape drives that you want to use.

SPTAPE

Privilege Class: D

When to Use SPTAPE

When the spooling system is overloaded with spool files, you can temporarily store files on tape to free up those resources. SPTAPE does the following:

- Stores to tape (**DUMP**) the unit record files that you want to schedule later on the VM/SP system
- Retrieves from tape (**LOAD**) the unit record files
- Reads the files back into the system at your selection
- Scans the tape for priority files.

SPTAPE

SPTAPE Format and Operands

SPTape	<pre> STOP <i>raddr</i> CANCEL <i>raddr</i> SCAN <i>raddr</i> SADump option2 LOAD <i>raddr</i> SADump option2 LOAD <i>raddr</i> { Printer } <i>spoolid1</i> [<i>spoolid2</i>] option2 option3 { PUnch } { Reader } } END <i>c1</i>[<i>c2</i>[<i>c3</i>[<i>c4</i>]]] { Class { FORM <i>form</i> { ALL } } DUMP <i>raddr</i> { Printer } <i>spoolid1</i> [<i>spoolid2</i>] option1 option2 option3 option4 { PUnch } { Reader } } END <i>c1</i>[<i>c2</i>[<i>c3</i>[<i>c4</i>]]] { Class { FORM <i>form</i> { ALL } } </pre>
	— OPTIONS —
	<pre> option 1 [MODE(800)] [1600] [6250] [38K] </pre>
	<pre> option 2 option 3 option 4 [LEAVE] [SYSHOLD] [PURGE] [REWInd] [USERHOLD] [RUN] [NOHOLD] </pre>

STOP *raddr*

stops the operation after CP completes the current file. CP positions the tape (*raddr*) at the beginning of the next file. You can continue with a new command at the current position.

CANCEL *raddr*

stops the operation immediately. CP then rewinds and unloads the tape (*raddr*), without regard to the previous tape positioning options.

SCAN *raddr*

scans the tape (*raddr*) from the current point to the first double tape mark it finds. Use one of the tape handling options to position the tape.

LOAD *raddr*

specifies the device address of the tape (*raddr*) from which CP will read the spool files. Use one of the tape handling options to position the tape.

Note: When you use this operand with the SADUMP operand, do not attach the tape drive to the virtual machine.

DUMP *raddr*

specifies the device address of the tape (*raddr*) where CP will write the spool files. Use one of the tape handling options to position the tape.

SADUMP

indicates that the data on the tape is output from the stand-alone dump facility.

Note: When you use this operand with the LOAD *raddr* operand, do not attach the tape drive to the virtual machine.

Printer**PRT**

is a printer-type spool file.

PUnch**PCH**

is a punch-type spool file.

Reader**RDR**

is a reader-type spool file. This includes MONITOR and CP system dump spool files.

$$\left. \begin{array}{l} \text{spoolid1} \left[\begin{array}{l} \text{spoolid2} \\ \text{END} \end{array} \right] \\ \text{CLASS } c1[c2[c3[c4]]] \\ \text{FORM } form \\ \text{ALL} \end{array} \right\}$$
spoolid

identifies the spool file or the start of a range of spool files ending with *spoolid2*.

Example:

SPTAPE spoolid 1 50

causes the operation to start at spoolid 1 and operate on all spoolids from 1 up until 50.

SPTAPE spoolid 1 55

operates on spoolids 1, 7, 5, 55, if a reader chain had spoolids in the following order: 50, 6, 1, 7, 5, 55.

END

dumps or loads all spool files after *spoolid1* on the output chain.

CLASS *c1[c2[c3[c4]]]*

indicates the output classes. You can specify up to four classes. If you are loading and dumping files with the same class, dump and load them synchronously to avoid any possible overlap of functions.

SPTAPE

FORM *form*

dumps or loads all files with the specified operator form.

ALL

dumps or loads all spool file classes.

[**MODE** (**800**)
 (**1600**)
 (**6250**)
 (**38K**)]

is the reset value of tape density for reel-to-reel tape systems. You may specify mode values of 800, 1600, 6250, or 38K. 1600 is the default, except for the 3480 tapes. For the 3480 tapes, the default is 38K. MODE also specifies the recording density for the 3480 magnetic tape subsystem.

Notes:

1. If a user specifies a density mode that the tape cannot handle, the control unit may not return an error condition. In this case, CP ignores the mode setting and uses the default control unit setting.
2. The following combinations of modes and tape drives are invalid:
 - Mode 800 with 3430 tape drives
 - Mode 38K with any tape drive other than the 3480
 - Modes 800, 1600, or 6250 with the 3480 tape drive.

[**LEAVE**
 REWind
 RUN]

LEAVE

does not move the tape after CP performs the tape operation.

REWInd

rewinds the tape after CP performs the tape operation.

RUN

rewinds and unloads the tape after CP performs the tape operation.

[**SYSHOLD**
 USERHOLD
 NOHOLD]

SYSHOLD

includes the system-held files in the tape operation.

USERHOLD

includes the user-held files in the tape operation.

NOHOLD

includes the non-held files in the tape operation.

[PURGE]

tells CP to delete the spool files from the spooling system after successfully writing them to tape.

Things You Should Know about the SPTAPE Command

1. The SPTAPE command does not handle multi-volume tapes.
2. While SPTAPE is processing spool files, other virtual machines cannot access or process the spool files.
3. If the format of the input spool file is different from the current spool file format, CP changes (or leaves) the restored spool file in system hold status. This happens when you move a spool file from:
 - A VM/SP Release 1 or 2 system to a VM/SP Release 3 system
 - A VM/SP Release 3 system to a VM/SP Release 1 or 2 system.
4. When you are using SPTAPE, do not attach the tape drive to another virtual machine.

System Response

The restored files keep the same characteristics as the original file, but CP assigns a new spoolid to the restored files. The response shows the old and new spoolids. It also shows that the SPTAPE command is currently processing the designated spool file on the indicated tape.

```
{ DUMPING }
{ LOADING } raddr { PRT }
{ SCANNING }      { PUN } userid spoolid1 class form stat [NOW spoolid2]
                  { RDR }
```

raddr

is the tape drive address where CP is performing the operation (dumping, loading, or scanning).

userid

identifies the user that originally created the file.

spoolid1

is the spoolid that CP first gave to the spool file.

SPTAPE

`class`
is the spool class of the file.

`form`
is the operator form number of the file.

`stat`
is the status of the files: USER, SYS, USYS (for USER and SYS), or NONE.

`spoolid2`
is the new spoolid that CP gave to the restored file.

If an I/O error occurs:

- Or CP detects an end-of tape, while writing files (SPTAPE DUMP) or reading files from the disk, you will receive a message stating that the tape is backed-up to the last complete spool file. CP writes a tape mark, rewinds and unloads the tape, and ends the command.
- While trying to read or write files for the SPTAPE LOAD option, you will receive a message. CP rewinds and unloads the tape, and ends the command. Spool files that CP has successfully read remain in the spooling system, but the system purges any partial file.

START

Privilege Class: D

When to Use START

Use the START command to:

- Restart a spooling device after it has been drained.
- Change the output class that the spooling device may service.

If the device is busy when you issue the command, the changed classes take effect after the current file is complete.

START Format and Operands

STArt	<p> <u>ALL</u> Printer PUnch Reader <i>raddr</i>[Class c...] </p> <p> [<i>FOrm form</i>] NOSep [AUto] [NO3800] [FLash name] [<i>FOrm *</i>] [SETup] [BEG3800] [CHars name] [MAAnual] [ANY3800] [FCB <i>plpi</i>] [IMAge <i>imagelib</i>] [PURge] </p> <p> [DEffcb] [FILEfcb] [CFIlefcb] </p>
-------	---

ALL
starts all readers, printers, and punches.

Printer
PRT
starts all printers. Printers that are in ready status, with files available for processing, immediately start to produce output when you issue this command.

START

PUnch **PCH**

starts all punches. Punches that are in ready status, with files available for processing, immediately start to produce output when you issue this command.

Reader **RDR**

starts all readers. Readers do not start until you supply a card deck and make the device ready.

raddr

is the address (cuu) of the spooling device that you want CP to start. You can specify more than one device. If the device is a reader, you cannot use any options.

This is the only operand with which you can use options.

Class c...

specifies the output classes that the device can service. You can specify up to four classes (single characters separated with blanks). If you specify more than one class, the device services the output files in the order that you specify the classes.

Example:

CLASS A D 1

specified for a printer, spools all class A files before class D files, and all class D before class 1.

Specify a class of * for printers or punches if you want the device to service any spool file class.

Form *form*

specifies the operator form that the printer or punch can process. This becomes the active form on the printer or punch. *form* is one to eight characters.

Form *

specifies that the printer or punch processes spool files with any form number. If you do not specify FOrM on the STArT command, the same form that you specified on the last STArT command remains in effect. CP keeps the last form across system warm and checkpoint starts, but not on cold starts.

If you do not issue the FOrM option with STArT since the last cold start, CP uses the default operator form specified in the SYSFORM macro.

NOSep

does not use a file separator for the output files. If you specify **NOSep** for the punch, the system does not punch separator cards. Instead, you will receive the following message:

```
PUN raddr START REQ CLASS c... FORM form status  
NOSEP
```

Issue **CP STAr** *raddr* or physically activate the **START** or **STOP** key on the device to start the punch for the next file.

If you do not specify this option, CP produces a separator.

AUto

processes files of any form, with automatic prompting messages for forms changes. CP processes the currently active form first.

SETup

processes files of any form, with automatic prompting messages for forms changes. You can check and adjust forms alignment when changing forms.

This option is valid only for printers.

MAnnual

tells CP to process only spool files with the currently active form. If the form is *, CP processes spool files with any form.

NO3800

restricts all 3800 LOAD CCWs. The device can print a file from any virtual device only if it has no 3800 LOAD CCWs. After a cold start, **NO3800** is the default value for the 3800 operands (**NO3800**, **BEG3800**, **ANY3800**).

The minimum abbreviation is **NO3**. This option is valid for all real spooling printers, not just 3800s.

BEG3800

restricts some 3800 LOAD CCWs. The device can print a file from any virtual device if it has no 3800 LOAD CCWs or if it has 3800 LOAD CCWs only at the beginning of the file.

The minimum abbreviation is **BEG**. This option is valid for all real spooling printers, not just 3800s.

ANY3800

does not restrict 3800 LOAD CCWs. The device can print any file.

The minimum abbreviation is **ANY**. This option is valid for all real spooling printers, not just 3800s.

START

FLash name

is the flash overlay sheet loaded into the device. *name* may be one to four characters in length. If you do not specify this operand, the previous value remains in effect.

This option is only valid for the 3800 printer.

CHars name

is the character arrangement table that CP will use for the separator page. *name* may be from one to four characters in length. If you do not specify this operand, the previous value remains in effect.

This option is valid for only the 3800 printer.

FCB *plpi*

specifies the forms control buffer for the separator page and any spool file for which you have not specified an FCB (using the SPOOL or CHANGE command). *plpi* (print lines per inch) must be a unique number 6, 8, or 12, and for the 3800 Model 3 device, 10. These numbers mean that you want the system to print the entire spool file at 6, 8, 10 or 12 lines per inch, regardless of the size of the paper that is currently loaded. If you do not specify this operand, the previous value will remain in effect.

This option is only valid for the 3800 printer.

IMage *imagelib*

specifies the named system image library that CP uses for tables. *imagelib* may be from one to eight characters and must be a valid named system. If you do not specify this option, the previous value will remain in effect.

This option is only valid for the 3800 printer.

PUrge

purges all files with load errors. If you do not specify this option, CP will hold all files.

This option is only valid for the 3800 printer.

DEffcb

uses the default FCB that was loaded in the printer. CP will not send imbedded LOAD FCB CCWs to the printer. The LOADBUF command specifies the default FCB.

This option is only valid for 3211-type printers.

FILEfcb

sends file FCBs to the real printer. File FCBs are LOAD FCB CCWs imbedded in the spool file. This is the default when you start the printer following a non-warm or non-checkpoint start.

This option is only valid for 3211-type printers.

Note: When a virtual machine loads a 3211 with a new FCB, the operator must make sure that the proper form (paper) is loaded and that the paper is aligned properly.

CFilefcb

checks the file FCBs. File FCBs are LOAD FCB CCWs imbedded in the spool file. You can send file FCBs to the real printer. After CP prints each copy of the spool file, CP sends a message saying that forms misalignment may have occurred and that it drained the printer. Check the forms alignment within the printer.

This option is only valid for 3211-type printers.

Things You Should Know about the START Command

1. If you do not specify either **MANual**, **AUto** or **SETup**, the previous mode setting remains in effect. CP keeps the mode across system warm and check-point starts. If a cold start occurs, the mode defaults to **MANual**.
2. If you do not specify either **NO3800**, **BEG3800** or **ANY3800**, the previous setting remains in effect.
3. If you want to print or punch without separator pages, specify **NOSEP** every time you enter the **START** command. Otherwise, **START** defaults to the **SEP** option.
4. **VM/SP** will assure that the FCB used for separator pages contains enough print lines. An FCB with more lines per inch than specified will be loaded, if necessary, for printing separator pages. For paper:
 - 5 1/2 inches long and smaller, a 12 lpi FCB is loaded.
 - 7 1/2 inches long and smaller (down to 5 1/2 inches), a 10 lpi FCB is loaded (12 lpi for the 3800 Model 1), unless 12 lpi was specified.
 - 8 1/2 inches long and smaller (down to 7 1/2 inches), an 8 lpi FCB is loaded, unless 10 or 12 lpi was specified.

Select a character set that matches the FCB so that the tops of characters will print.

System Response

You may receive more than one response line for this command. The format of the response depends on the device type. The response is identical to that of the "QUERY UR" command. In each case the status is given as **STARTED**. Look in "QUERY" on page 3-117 for a description of the response to Query UR.

STCP

STCP

Privilege Class: C

When to Use STCP

Use the STCP command to change the contents of real storage.

Note: You cannot change the real PSW or real registers with this command. Also, in a system running in attached processor mode, this command cannot modify data in shared pages.

STCP Format and Operands

STCP	$\left. \begin{array}{l} \{ \text{MLhexloc} \\ \text{NLhexloc} \\ \text{Mhexloc} \\ \text{Nhexloc} \\ \text{Lhexloc} \\ \text{hexloc} \} \end{array} \right\} \text{hexword1 [hexword2...]}$ $\left. \begin{array}{l} \{ \text{MShexloc} \\ \text{NShexloc} \\ \text{Shexloc} \} \end{array} \right\} \text{hexdata}$
-------------	--

MLhexloc
NLhexloc
Mhexloc
Nhexloc
Lhexloc
hexloc

stores the data given in *hexword1* [*hexword2...*] in successive fullword locations starting at the address specified by *hexloc*. The smallest group of hexadecimal values that you can store using *hexloc* is one fullword. The system aligns data to the nearest fullword boundary. If the data that the system is storing is less than a fullword (eight hexadecimal digits), CP right-adjusts the word, and fills the high order bytes of the word with zeros.

In attached processor systems, *hexloc* and *Lhexloc* values are absolute addresses. The M prefix causes the system to interpret that address through the prefix register of the main processor. In MP systems, M corresponds to addresses on the IPL processor.

In MP systems, the N prefix causes the system to interpret addresses as if they were on the non-IPL processor. In attached processor systems, the N prefix causes the system to interpret the address through the prefix register for the attached processor.

Example:

STCP N64 20000

stores X'00020000' into the second word of the SVC new PSW in the PSA (prefix storage area) of the attached processor.

If you don't specify M or N, the system will interpret the address as an absolute address on both AP and MP systems. N is valid while the second processor is in operation. M is valid if the system was generated for the attached processor or multiprocessor mode of operation.

MShexloc

NShexloc

Shexloc

stores the data, given in hexdata, in the address that *hexloc* specifies without word alignment. The shortest string that the system can store is one byte (two hexadecimal digits). If the string has an odd number of characters, the system does not store the last character. An error message occurs and the function ends.

Attached processor VM/SP applications use *MShexloc* and *NShexloc* operands. If you prefix *Shexloc* with M or N, the system interprets the specified hexadecimal addresses through the associated prefix registers of the main or attached processor.

hexword

specifies up to eight hexadecimal digits. If you specify fewer than eight digits, the system right-justifies the string in a fullword and left-fills it with zeros. If you specify two or more hexwords, you must separate them with at least one blank.

hexdata

specifies a string of two or more hexadecimal digits with no embedded blanks.

System Response

STORE COMPLETE

TRANSFER

TRANSFER

Privilege Class: D

When to Use TRANSFER

Use the TRANSFER command to:

- Direct one or more spool files to a specified user.
- Reclaim reader spool files that CP previously sent to one or more users.

TRANSFER Format and Operands

TRANSfer	$\left[\begin{array}{l} \textit{userid} \\ \text{SYSTEM} \\ * \\ - \end{array} \right] \left[\begin{array}{l} \text{Printer} \\ \text{PUnch} \\ \text{Reader} \end{array} \right] \left\{ \begin{array}{l} \textit{spoolid} \\ \text{Class } c \\ \text{FORM } \textit{form} \\ \text{ALL} \end{array} \right\} \left\{ \begin{array}{l} \text{[To]} \\ \text{From} \end{array} \right\} \left\{ \begin{array}{l} * \\ \textit{userid} \\ \text{ALL} \\ \textit{userid} \end{array} \right\} \left[\begin{array}{l} \text{Printer} \\ \text{PUnch} \\ \text{Reader} \end{array} \right]$
----------	---

userid

identifies the user to whom CP will transfer or reclaim the spool files.

SYSTEM

lets you manipulate all spool files in the system, regardless of the *userid*.

*

lets you manipulate your own spool files.

Printer

PRT

is a printer-type queue to which or from which CP will transfer the files.

PUnch

PCH

is a punch-type queue to which or from which CP will transfer the files.

Reader

RDR

is a reader-type queue to which or from which CP will transfer the files.

spoolid

identifies the file that you are transferring.

Class *c*
transfers all files of the class that you specify.

FORM *form*
transfers all files of the form that you specify. *form* is one to eight characters. If you specify the * option or let it default, the form number is the user form. If you specify the *userid*/SYSTEM option, the form number is the operator form.

ALL
transfers all spool files of the queue that you specify, regardless of *class*, *form*, or *spoolid*.

To { * }
 { *userid* }

is the user to whom you are directing the files. If you omit the optional keyword TO, *userid* cannot be T or TO. The asterisk (*) means that you want CP to send the files to you.

From { ALL }
 { *userid* }

is the user from whom CP will reclaim the spool files. Specify ALL to reclaim files from all users, starting with the *userid* that you specified first.

Note: If you use the TRANSfer SYSTEM From command to reclaim files, CP directs the files back to their originators.

Things You Should Know about the TRANSFER Command

The command will only transfer files to queues of a valid type. If you try to transfer printer to punch or punch to printer, CP indicates an error. You can transfer reader files to the printer/punch, but CP will only do valid transfers. For example, CP will not transfer real reader, punch or Dump/Monitor files to the printer. The following chart shows the valid queues for each spool file origin.

O r i g i n	Spool Files	Queue		
		reader	printer	punch
Real Reader	*			
Printer	*		*	
Punch	*			*
Dump/ Monitor	*			

Note: Dump/Monitor includes CP system dumps, user dumps (VMDUMP) command, NCP dumps, or VM Monitor spool files.

TRANSFER

System Response

{ RDR } FILE spoolid TRANSFERRED TO userid { RDR }
{ PRT } { PRT }
{ PUN } { PUN }

tells what your input was when you issued the TRANSFER command. This response identifies the pool file and the user to whom CP transferred the file.

Note: If you used the asterisk (*), the user will be yourself (the person who issued the command).

{ nnnn } FILES TRANSFERRED
{ NO }

is the response to the user who issued the TRANSFER command.

Note: Do not specify the SET IMSG OFF command if you are monitoring the RSCS virtual machine for your installation; you need to see informational messages to accurately monitor RSCS system activity.

UNLOCK

Privilege Class: A

When to Use UNLOCK

Use the UNLOCK command to unlock page frames that a previous LOCK command locked.

UNLOCK Format and Operands

UNLOCK	$\left. \begin{array}{l} \left\{ \begin{array}{l} \textit{userid} \\ \text{SYSTEM} \end{array} \right\} \textit{firstpage lastpage} \\ \left\{ \begin{array}{l} \text{VIRT=REAL} \\ \text{V=R} \end{array} \right\} \end{array} \right\}$
---------------	---

userid

identifies the virtual user that you previously specified in a LOCK command.

SYSTEM

unlocks a previously locked pageable CP nucleus page.

firstpage

is the hexadecimal value of the first page that you want to unlock.

lastpage

is the hexadecimal value of the last page to be unlocked.

Note: For *firstpage* and *lastpage*, specify only the page numbers, not the displacement. For example, to unlock USERA's virtual storage locations '12000' to '2CFFF', issue the following command:

```
UNLOCK USERA 12 2C
```

VIRT=REAL

V=R

releases, for normal paging use, all the page frames normally for the virtual=real machine. However, the area must not be occupied by a virtual=real virtual machine.

Note: Once the system releases the virtual=real area, it cannot reactivate the area to run another virtual=real machine until VM/SP is loaded again.

UNLOCK

System Response

COMMAND COMPLETE

occurs even if some (or all) of the pages unlocked were not previously locked.

VARY

Privilege Class: B

When to Use VARY

Use the VARY command to mark a device available or unavailable for a user or CP to use. (For 370x resources running in NCP or PEP mode, or remote 3270 resources, use the NETWORK VARY command.)

VARY Format and Operands

VARY	$\left. \begin{array}{l} \{\text{ONLine}\} \\ \{\text{OFFline}\} \end{array} \right\} \left. \begin{array}{l} \{raddr-raddr\} \\ \{raddr\dots\} \\ \{\text{PROcessr}\} \end{array} \right\}$ <p style="text-align: center;">Offline PROcessr <i>nn</i> [<u>VPHY</u>] [FORCE] [<u>VLOG</u>]</p>
-------------	--

ONLine

makes an offline device(s) available or makes offline paths to an online device available.

After a WARM IPL or CKPT IPL, the first time you VARY a printer ONLINE it will have the same status (for example, DRAINED, WAITING) that was present at the last checkpoint before the system shut down.

OFFline

makes the device(s) unavailable.

VM/SP remembers spooled unit record devices marked offline through any system failures. A shutdown and subsequent cold start assumes that all devices are available.

raddr

is the real address of the device whose status you want to change. You can specify more than one device or a range of device addresses. CP also changes any alternate paths to the device(s) that you specify online or offline.

When you enter more than one device or a range of devices, CP continues to process the VARY command even if it finds an error while trying to vary one of the devices online/offline. CP issues an error message for every

device that causes an error and an informational message for every device that it could vary online/offline.

Note: If the device is a 2305, the address must be for the first of 8 exposures, that is, address xx0 or xx8.

PROcEssr nn

is the processor that CP will make available or unavailable. nn is the processor address (hexadecimal value from 00 through 3F) assigned when your installation installed the processor.

For VARY ONLINE PROcEssr, CP checks to see if the processor is already online. If it is not, and if the processor is available and operational, CP does the following:

- Establishes prefixing
- Initializes the control registers and prefix areas
- Sets the clock comparator and CPU timer.

The system can then function in attached processor or multiprocessor mode.

For VARY OFFLINE PROcEssr, CP checks to see if the processor is online and operational. CP also ensures that, with the specified processor offline, an online path will still remain to all system-owned devices. If the processor is operating in multiprocessor mode, CP ensures that any device that would lose its last online path is not currently in use. CP then does the following:

- Stops work flow (quiesces) to the specified processor
- Clears the control registers and prefix register
- Notifies any users having affinity to the specified processor that the affinity is set off; CP puts these users in console function mode.

Offline PROcEssr nn | | |-------------| | VPHY | | VLOG |

VPHY

physically varies the processor offline by the MSSF in a 3081 processor configuration.

VLOG

causes the processor to be logically offline to VM/SP but not physically offline as when MSSF is called. This option is for shared system environments.

If you want to run a program on a 3081 in single processor mode, enter the command as follows:

VARY OFF PROC nn VLOG

FORCE

abruptly stops the processor that you specify. Use this option only when all other means of varying the processor offline have been unsuccessful.

Use **FORCE** when you need to stop the processor immediately or when the system cannot stop all activity on devices attached to the processor. When you issue **VARY OFFLINE PROC FORCE**, CP does not check for device activity before marking devices offline. This may result in a device being hung, or it may preclude a device's being varied online again.

Things You Should Know about the VARY Command

1. If system performance monitoring is taking place when you want to vary a device, use the **MON STOP** command to stop monitoring. Restart monitoring after the system makes the configuration change.
2. If the system is only enabled for the **UP** mode, do the following:
 - a. Stop the online processor
 - b. Enable the **MP** features for the system
 - c. Restart the online processor
 - d. Vary the offline processor online.

System Response

$\left. \begin{matrix} \{raddr\dots\} \\ \{raddr-raddr\} \end{matrix} \right\} \text{ VARIOUS } \left\{ \begin{matrix} \text{ONLINE} \\ \text{OFFLINE} \end{matrix} \right\}$

is the response when CP successfully varies the device (*raddr*) online or offline.

RADDR MISSING OR INVALID

is the response if you entered the **PROCESSR** operand without a processor number or with any invalid number.

PROCESSOR nn DOES NOT EXIST

is the response if the processor that you specified in **VARY PROCESSR nn** is not defined to the system.

VARY PROCESSOR COMMAND FAILED

For **VARY OFFLINE PROCESSR**, this response means one of the following:

- The processor that you specified is in uniprocessor mode or not online.
- The processor that you specified is the main processor in an attached processor system.

VARY

- In a multiprocessor system, the system-owned device would lose its last online path, or a device that would lose its last online path is currently in use.

For VARY ONLINE PROCESSR, this response means one of the following:

- The system is not generated for AP mode.
- The MP function is not installed.
- The system is already in AP mode.

PROCESSOR nn ONLINE
the specified processor is now online and operational.

PROCESSOR nn OFFLINE
the specified processor is now offline and not operational.

LAST PATH TO DEVICE raddr ON PROC nn
the system-owned device that you specified by raddr will lose its last online path if CP varied processor nn offline.

WARNING

Privilege Classes: A or B

When to Use WARNING

Use the WARNING command to transmit high-priority messages to a user that you specify or to all users.

WARNING does not wait for input operations at the terminal to complete. Instead, WARNING breaks in on the line that the terminal is currently displaying and sends the message immediately (unless you or someone else previously issued the SET WNG OFF command for the virtual machine).

WARNING Format and Operands

Warning Wng	$\left\{ \begin{array}{l} \textit{userid} \\ \text{OPerator} \\ \text{ALL} \end{array} \right\} \textit{msgtext}$
----------------	---

userid
identifies the user who will receive the message.

OPerator
sends the message to the primary system operator regardless of his *userid*.

ALL
sends the message to all users able to receive warning messages.

msgtext
is the text of the warning.

Things You Should Know about the WARNING Command

You can have your TELL EXEC use WARNING to send warnings to users on the same system as you.

DEFAULTS set tell msgcmd warning
changes the default set up in the GLOBALV file.

DEFAULTS set tell warning msg
changes default back to MSG.

You must reset the default again if you want to change it.

WARNING

System Response

hh:mm:ss WNG FROM userid: msgtext
is the response to the users who receive warning messages. hh:mm:ss is the time in hours, minutes, and seconds when the system sends the message to the user.

If a user is using a 3270, WARNING puts the screen in a HOLDING status and rings the alarm, if present. Press the cancel key to return to the RUNNING status. If you send the WARNING during system IPL or restart operations, WARNING will put the screen in a MORE status.

Section 4. VM/SP Service Programs

Controlling the 3704/3705/3725 Communications Control Program

The *EP/3725 Installation and Resource Definition Guide and Reference*, SC30-3172, provides the information necessary to:

- Define and generate an Emulator Program for the IBM 3725
- Load the control program into the controller
- Dump the contents of the controller storage.

VM/SP does not provide loading or dumping facilities for the 3725. ACF/NCP-SSP provides these facilities for both 3705 and 3725 communications controllers. Refer to the publications listed in the preface for more information.

Operational control of the 3704/3705 Communications Controller (hereinafter referred to as 3705) resides with the system operator via the NETWORK command. In addition, the behavior of the 3705 device is, for a large part, dependent upon the communications control program with which it was loaded. The communication control programs are created, assembled, or modified by a set of CMS commands specifically designed for this purpose. These commands are:

ASM3705
GEN3705
LKED
SAVENCP
ZAP

The cited commands, described in greater detail in the *VM/SP Installation Guide*, are the system programmer's responsibility. The 3705 NETWORK and NCPDUMP commands, however, are the responsibility of the VM/SP system operator and are described elsewhere in this publication.

The 3704/3705 Dynamic Trace Facility is supported for virtual machine operation. However, if the 3704/3705 in emulator mode is not dedicated to the user invoking the trace, the user is required to have in his directory a privilege class other than (or in addition to) G.

Service Programs

Using CMS To Alter VM/SP Files

The VM/SP system operator is usually assigned a virtual machine with a large DASD storage allotment. This DASD storage can be assigned to other users as a temporary work area or for short-term DASD storage. In addition, the operator's virtual machine can record system statistical data, or create and maintain directory files and other system files.

For whatever reason the operator uses the system, it is probable that some files will need to be modified at some time. One of the easiest ways to modify such system files is to use the VM/SP System Product Editor (XEDIT) facilities.

VM/SP XEDIT facilities provide ways to create and modify all types of CMS files. For a complete description of VM/SP XEDIT facilities and commands, see *VM/SP System Product Editor User's Guide* and *VM/SP System Product Editor Command and Macro Reference*.

Other CMS Commands

There may be occasions, caused by user demands, when other facilities of CMS may be invoked. For example, to create punched output of a file for use by another system, or to create a history file. The commands likely to be needed by the system operator are the following:

CMSBATCH	LISTFILE	STATE
COMPARE	MOVEFILE	TAPE
COPYFILE	PRINT	TAPEMAC
ERASE	PUNCH	TAPPDS
FORMAT	QUERY	TYPE
LABELDEF	READCARD	UPDATE
LISTDS	SET	

A complete description of these and other commands and their usage is contained in *VM/SP CMS User's Guide* and *VM/SP CMS Command and Macro Reference*.

DASD Dump Restore (DDR) Service Program and How to Use It

Use the DASD Dump Restore (DDR) program to dump, restore, copy, or print VM/SP user minidisks. The DDR program may run as a standalone program, or under CMS via the DDR command.

The DDR program has five functions:

1. Dumps part or all of the data from a DASD device to tape.

Service Programs

2. Be aware that DDR when run as a standalone program does not have error recovery support. However, when DDR is invoked in CMS, in a virtual machine environment, the I/O operation is performed by CP (CP has built-in error recovery facilities).
3. When running standalone, DDR will search for a console at address 009 or 01F. If these consoles are not operational, the program will enter a wait state, waiting for an interrupt to identify the console. If any nonconsole type device is physically connected to address 009 or 01F, it must be made nonoperational or the results will be unpredictable.

Invoking DDR as a Standalone Program

To use DDR as a standalone program, the operator should IPL it from a real or virtual IPL device as he would any other standalone program. Then indicate where the DDR program is to obtain its control statements by responding to prompting messages at the console.

Notes:

1. Be aware that DDR when run as a standalone program has only the most elementary error recovery support. However, when DDR is invoked in CMS, in a virtual machine environment, the I/O operation is performed by CP which has better error recovery facilities.
2. The standalone DDR utility will not support cylinder faults for MSS virtual volumes when performing the DUMP or COPY functions.
3. It may be necessary to disable channel-to-channel devices in order for the DDR utility to run as a standalone program.
4. The DDR utility does not assign 3480 tape drives to itself. So, when a 3480 is in use during standalone DDR processing, do not IPL any other processor that has a path to the device. This is because the other processor might steal the 3480 drive by assigning it to itself. This would immediately stop DDR's access to the device. For the same reason, do not vary the 3480 device online to any other processor until DDR processing completes.

DDR Control Statements

DDR control statements describe the intended processing and the needed I/O devices. I/O definition statements must be specified first.

All control statements may be entered from either the console or the card reader. Only columns 1 to 71 are inspected by the program. All data after the last operand in a statement is ignored. An output tape must have the DASD cylinder header records in ascending sequence; therefore, the extents must be entered in sequence by cylinder or by extent. Only one type of function — dump, restore, or copy — may be performed in one execution, but up to 20 statements describing DASD cylinder extents may be entered. The function statements are delimited by (preceded by) an INPUT and OUTPUT statement. If additional functions are to be per-

formed, the sequence of control cards must be repeated. If you do not use INPUT or OUTPUT control statements to separate the functions you specify when the input is read from a card reader or CMS file, an error message is displayed. However, the remainder of the input stream will be checked for proper syntax, but no further DDR operations will be performed. Only those I/O devices defined by the INPUT statement and the OUTPUT statement must be redefined in subsequent steps. The SYSPRINT I/O definition remains the same.

To return to CMS, enter a null line (carriage return) in response to the prompting message (ENTER:). To return directly to CP, key in #CP.

The PRINT and TYPE statements work differently from other DDR control statements in that they operate on only one data extent at a time and it is not necessary to respecify the INPUT statement ahead of each PRINT or TYPE statement. If the input is from a tape created by the dump function, the tape must be positioned at the header record for each step. The PRINT and TYPE statements have an implied output of either the console (TYPE) or system printer (PRINT), so no OUTPUT statement is required.

I/O Definition Statements

The I/O definition statements describe the tape, DASD, and printer devices used while executing the DASD Dump Restore program.

INPUT/OUTPUT Control Statement

An INPUT or OUTPUT statement describes each tape and DASD unit used. The format of the INPUT/OUTPUT statement is:

INput OUTput	<i>cuu type</i> [<i>volser</i>] [(Options...)] [<i>altape</i>]
	Options: [SKip <i>nn</i>] [MOde 6250] [REWind] COmpact [SKip 0] [MOde 1600] [UNload] [MOde 800] [LEave] [MOde 38K]

INPUT

indicates that the device described is an input device.

OUTPUT

indicates that the device described is an output device.

Note: If the output device is a DASD device and DDR is running under CMS, the device is released using the CMS RELEASE command function and DDR processing continues.

Service Programs

cuu

is the unit address of the device.

type

is the device type (2311, 2314, 2319, 3330, 3330-11, 3340-35, 3340-70, 3350, 2305-1, 2305-2, 2400, 2401, 2415, 2420, 3375, 3380, 8809, 3410, 3411, 3420, 3430, 3480, or FB-512). There is no 7-track support for any tape devices. Specify a 3340-70F as a 3340-70, and a 3333 as a 3330. Specify a 3350 that is in 3330-1 or 3330-11 compatibility mode as a 3330 or 3330-11. Specify a 3344 as a 3340-70, and specify 3350 for a 3350 operating in native mode (as opposed to compatibility mode).

The DASD Dump Restore (DDR) program, executing in a virtual machine, uses I/O DIAGNOSE 20 to perform I/O operations on tape and DASD devices. DDR under CMS requires that the device type entered agree with the device type of the real device as recognized by VM/SP. If there is a conflict with device types, the following message is issued:

```
INVALID OPTION
```

However, if DDR executes standalone in a virtual machine, DDR uses DIAGNOSE 20 to perform the I/O operation if the device types agree. If the device types do not agree, an error message is issued.

volser

is the volume serial number of a DASD device. If the keyword "SCRATCH" is specified instead of the volume serial number, no label verification is performed.

altape

is the address of an alternate tape drive.

If multiple reels of tape are required and "altape" is not specified, DDR types the following at the end of the reel:

```
END OF VOLUME CYL xxx HD xxx, MOUNT NEXT TAPE
```

After the new tape is mounted, DDR continues automatically.

SKIP $\left[\begin{array}{l} nn \\ 0 \end{array} \right]$

forward spaces nn files on the tape. nn is any number up to 255. The SKIP option is reset to zero after the tape has been positioned.

MODE $\left[\begin{array}{l} 6250 \\ 1600 \\ 800 \\ 38K \end{array} \right]$

Mode 38K, 6250, 1600, or 800 causes all output tapes that are opened for the first time and at the load point to be written or read in the specified density. All subsequent tapes mounted are also set to the specified density.

If no mode option is specified, then no mode set is performed and the density setting remains as it previously was unless the tape is positioned at the load point. When this occurs, the density setting resets to 1600 (the default value).

MODE also specifies the recording density for the 3480 magnetic tape subsystem. Since the 3480 records only at a density of 38K, the default for the MODE option of the 3480 is 38K.

REWIND

rewinds the tape at the end of a function.

UNLOAD

rewinds and unloads the tape at the end of a function.

LEAVE

leaves the tape positioned at the end of the file at the end of a function.

COMPACT

causes the output tape to be in a compact format, which uses less tape space than standard format. DDR stores data in a compact format by compressing strings of duplicate data into a smaller amount of space and reducing the amount of space necessary to represent the characters in the data. This option is valid only on the OUTPUT control statement for the DUMP functions.

You can use tapes in the compact format as input to the RESTORE, COPY, PRINT, and TYPE functions without any changes. For more information, refer to the following sections about function statements.

Notes:

1. When the wrong input tape is mounted, the message DMKDDR709E is displayed and the tape will rewind and unload regardless of options REWIND, UNLOAD, or LEAVE being specified.
2. If DDR is executed from CMS, failure to attach the tape drive or the disk device (or both) to your virtual machine prior to invoking the input/output statement causes the following response to be displayed:

```
INVALID INPUT OR OUTPUT DEFINITION
```

SYSPRINT Control Statement

The SYSPRINT control statement describes the device that output is to be sent to. If the SYSPRINT CONS option is specified, the output is directed to the console for both the CMS environment and the standalone DDR virtual machine.

In the CMS environment, all output is directed (by default) to 00E, unless the SYSPRINT CONS option is specified. Any SYSPRINT cuu option specification is ignored.

Service Programs

In the standalone DDR virtual machine, the output is directed to the output device specified by the **SYSPRINT** *cuu* option. If the **SYSPRINT** **CONS** option is specified, all output is directed to the console. If no options are specified, the output is directed (by default) to 00E.

SYsprint	{ <i>cuu</i> CONS }
-----------------	-------------------------------

cuu

specifies the unit address of the device.

CONS

specifies the console as the output device.

Function Statements

The function statements tell the DDR program what action to perform. The function commands also describe the extents to be dumped, copied, or restored. The format of the **DUMP/COPY/RESTORE** control statement is:

DUmp COpy REstore	[FTr ¹] [<i>block1</i> [To] [<i>block2</i> [Reorder] [To] [<i>block3</i>]] <i>cyl1</i> [To] [<i>cyl2</i> [Reorder] [To] [<i>cyl3</i>]] { CPvol ALL NUcleus }
--	---

¹The FTr operand is valid with only the **DUMP** control statement.

DUMP

requests the program to move data from a direct access volume onto a magnetic tape or tapes. After the system dumps the data, if you specified the **COMPACT** option on the **OUTPUT** control statement, you will receive these messages:

```
' BYTES IN _____ BYTES OUT _____ '
' TRACKS NOT COMPACTED ON TAPE _____ '
' BLOCKS NOT COMPACTED ON TAPE _____ '
```

The format of the tape depends on the type of the direct access volume.

Note: If you specify dump with the **COmpact** option, the system dumps the data in an **FTR** format.

FTr

requests an output tape format of variable unblocked records. The size of the records and the number of records per track written to the tape depend on the density of the tape.

The option can be used for those devices supporting the full-track-read (FTR) feature (3330, 3340, 3350, 3375, and 3380) and for FBA devices (FTR is the default for 3375 and 3380 and, therefore, need not be specified).

If FTr is specified on the DUMP control statement for a count-key-data DASD but the control unit does not support the feature, a message is written and the operation proceeds with data written in the old format.

For count-key-data direct access volumes, the data is moved cylinder by cylinder.

The format of the resulting tape is:

Non-FTR format:

Record 1

A volume header record, consisting of data describing the volumes.

Record 2

A track header record, consisting of a list of count fields to restore the track, and the number of data records written on tape. After the last count field, the record contains key and data records to fill the 4K buffer.

Record 3

Track data records, consisting of key and data records packed into 4K blocks, with the last record truncated.

Record 4

Either the end-of-volume (EOV) or end-of-job (EOJ) trailer label. The end-of-volume label contains the same information as the next volume header record, except that the ID field contains EOJ. The end-of-job trailer label contains the same information as record 1 except that the cylinder number field contains the disk address of the last record on tape and the ID field contains EOJ.

FTR format:

Record 1

The same as described for the non-FTR format.

Record 2

A track header record, consisting of fields containing the length of the track, the density of the tape, and the number of count fields in the track followed by the track contents to fill the record, making it the same length as Record 3.

Service Programs

Record 3

Track data records, consisting of count-key-data records in 8K, 12K, or 48K blocks for 800, 1600, or 6250 BPI respectively, or 48K blocks for the 3480 Tape Subsystem. The last block, in all cases, is a short block.

Record 4

The same as described for the non-FTR format.

For FBA devices, the data is moved in sets of blocks. Each set contains 95 (or less for the last set) blocks of data. Any number of blocks can be moved with one DUMP statement. The format of the resulting tape depends on whether the FTR option is used or not, and/or the density of the output tape.

Record 1

The same as described for the non-FTR format of CKD devices.

Record 2

A data header record. This consists of data that describes the set of blocks that follow (such as block numbers and the number of tape records required to hold these FB-512 blocks). Following the control data are the actual FB-512 blocks filling out the tape record.

Record 3

FB-512 data records. These contain the rest of the blocks making up the set.

Record 4

The same as described for the non-FTR format of CKD devices.

In non-FTR format the record length of record 2 and record 3 is 4K bytes.

For FTR or compact formatted tapes, record length is 8K, 12K, or 48K blocks for 800, 1600, or 6250 BPI respectively, or 48K blocks for the 3480 Tape Subsystem. The last block, in all cases, is a short one.

COPY

requests the program to copy data from one device to another device of the same or equivalent type. Data may be recorded on a cylinder or block basis from input device to output device. A tape-to-tape copy can be accomplished only with data dumped by this program.

You may use a tape in compact format as input. For a tape-to-tape copy, the output tape will be in the same format (compact or standard) as the input tape. The COMPACT option on the OUTPUT control statement is not valid for the COPY function. If it is specified, the system displays the following message:

```
'COMPACT OPTION IGNORED FOR COPY OPERATIONS'
```

Note: You cannot copy between FBA and count-key-data devices.

RESTORE

requests the program to return data that has been dumped by this program. Data can be restored only to a DASD volume of the same or equivalent device type from which it was dumped. It is possible to dump from a real disk and restore to a minidisk as long as the device types are the same.

You can use a tape in compact format as input. DDR checks if the input is in compact format, and expands the data back to standard format, if needed. You do not need to specify anything to the program about the tape format. After the system restores the data, you will receive the following message:

```
'BYTES RESTORED _____'
```

cyl1 [TO] [*cyl2* [REORDER] [TO] [*cyl3*]

Only those cylinders specified are moved, starting with the first track of the first cylinder (*cyl1*), and ending with the last track of the second cylinder (*cyl2*). The REORDER operand causes the output to be reordered, that is, moved to different cylinders, starting at the specified cylinder (*cyl3*) or at the starting cylinder (*cyl1*) if *cyl3* is not specified. The REORDER operand must not be specified unless specified limits are defined for the operation; the starting and, if required, ending cylinders (*cyl1* and *cyl2*) must be specified. If the input device cylinder extents exceed the number of cylinders specified on the output device, an error message results.

block1[TO] [*block2* [REORDER] [TO] [*block3*]

only the specified blocks are moved, starting with the first block, up to and including the last block. The REORDER operand causes the data to be moved to a different DASD location. The REORDER operand must not be specified unless limits are defined for the operation. If the input block extents exceed the capacity of the output device, an error message results.

CAUTION:

The REORDER operand is intended to be used to move minidisks to new locations; it is not intended for re-locating non-minidisk cylinders. To understand the difference, consider a 10 cylinder minidisk. Its cylinders are numbered 0-9 and the count fields of its records refer to cylinders 0-9. Furthermore, if the minidisk contains location-dependent data, then references to cylinders 0-9 will be hidden within that data. When REORDER is used to move the minidisk to a new real location, the minidisk cylinders are nevertheless still regarded as being cylinders 0-9 and there is no need to change the cylinder numbers in the count field of the records. On the other hand, when moving non-minidisk cylinders, one would generally want the count fields of the moved records to reflect the new cylinder addresses; but REORDER keeps the original cylinder numbers in the count fields.

CPVOL

specifies that cylinder 0 for count-key-data devices, or blocks 0-16 for FBA devices including all active directory, override space, and permanent disk space are to be copied, dumped, or restored. This indicates that both source and target disk must be in CP format, that is, the CP Format/Allocate program must have formatted them.

Service Programs

ALL

specifies that the operation is to be performed on the entire DASD volume, either cylinders or blocks. This operation is not valid for alternate track cylinder assignments on some devices. (See "Restrictions".)

Note: The occurrence of message DMKDDR705E (issued upon completion of the copy restore or dump operation) indicates that an attempt was made to copy, restore, or dump the contents of cylinders beyond the extents of the designated minidisk.

NUCLEUS

specifies that record 2 on cylinder 0, track 0 and the nucleus cylinders for count-key-data devices are dumped, copied, or restored. These are blocks 5-12 for FBA devices.

Note: To do a 'DDR RESTORE NUC', the tape must be created by 'DDR DUMP NUC'. Otherwise, the system would issue message DMKDDR723E.

Restrictions

1. Each track processed by this utility must have a valid home address on it containing the real cylinder and track location. Even when restoring and copying data to a track it must have a pre-existing home address on it.
2. Each track on an input DASD device must have a valid record zero on it, with no more than eight bytes in the key and data fields of the record. Each track on an output DASD device must also have a valid record zero on it unless that device is 2314, 2319, 2305-1, or 2305-2.
3. Flagged tracks are treated just as any other track for all 2314, 2319, and 2305 devices. That is, no attempt is made to substitute the alternate track data when a defective primary track is read. In addition, tracks are not inspected to determine whether they were previously flagged when written. Therefore, volumes containing flagged tracks should be restored to the same cylinders of the volume from which they were dumped. The message DMKDDR715E occurs each time a defective track is dumped, copied or restored, and the operation continues. When ALL is specified for these device types, both the primary cylinders and the high-order cylinders normally reserved for alternate tracks are dumped, copied, or restored.
4. Flagged tracks on 3330, 3340, 3350, 3375, and 3380 devices are handled so that data is transferred to or from the assigned alternate track in place of the defective track. (For 3330, 3375, 3380 and 3350 this is accomplished automatically by the hardware of the control unit while for the 3340 and 3344 it is accomplished through software.)

Note: Alternate track recovery for overflow records is not provided by VM/SP for 3340s and 3344s.

The tape created by dumping one of these types of DASD devices will appear as if it had been dumped from a defect-free device and the tape can be restored to any device of the same type, even though that device might not have the

same tracks flagged defective as the original device had. (The COPY function works this way also.) If a track is flagged as defective, but has no alternate assigned, a warning message is issued and the only data transferred is the home address record and record zero. When ALL is specified for these device types, only the primary cylinders are processed; the cylinders reserved for alternate tracks are not processed except that an assigned alternate track is processed whenever the corresponding defective track is processed. However, by specifying the cylinder range explicitly (cyl1 to cyl2 format), all cylinders, including cylinders in the alternate track area, can be dumped or copied from. But these same cylinders cannot be restored to or copied to explicitly. It is intended that explicitly dumped cylinders in the alternate track area will be restored to another area via the REORDER operand. The only reason the explicit dumping and copying of cylinders from the alternate track area is allowed at all is to facilitate conversion of 3340 and 3344 disks that were written using early releases of VM/SP.

5. For DASD devices other than the 3375 or 3380 (which will default to FTR mode) that support the full track read (FTR) processing, you must specify the option. Otherwise, the tape will be produced in the current non-FTR DDR format of 4096 blocks. The 3330/3340 DASD devices can only take advantage of the full track read feature only when the 3830 or 3880 has microcode supporting either 3344 or 3350.
6. The system cannot use tapes created by the DDR DUMP function, which are in compact format, as input to earlier levels of DDR.
7. The 8809 tape drive may not operate efficiently in streaming mode while DDR is processing data in the compact format.

Example:

```
INPUT 191 3330 SYSRES
OUTPUT 180 2400 181 (MODE 800
SYSPRINT 00F
DUMP FTR CPVOL
INPUT 130 3330 MINI01
DUMP 1 TO 50 REORDER 51
60 70 101
```

This example sets the density to 800 BPI, then dumps all pertinent data from the volume labeled SYSRES onto the tape that is mounted on unit 180. If the program runs out of space on the first tape, it continues dumping onto the alternate device (181). A map of the dumped cylinders is printed on unit 00F while the program is dumping. When the first function is complete, the volume labeled MINI01 is dumped onto a new tape. Its cylinder header records are labeled 51 to 100. A map of the dumped cylinders is printed on unit 00F. Next, cylinders 60 to 70 are dumped and labeled 101 to 111. This extent is added to the cylinder map on unit 00F. When the DDR processing is complete, the tapes are unloaded and the program stops.

If cylinder extents are being defined from the console, the user need only enter DUMP, COPY or RESTORE on the command line. The following is displayed:

Service Programs

ENTER CYLINDER EXTENTS
ENTER:

For any extent after the first extent, the message

ENTER NEXT EXTENT OR NULL LINE
ENTER:

is displayed. The user may then enter additional extents to be dumped, restored, or copied. A null line causes the job step to start.

Notes:

1. When a cylinder map is printed on the virtual printer (00F as in the previous example) a heading precedes the map information. Module DMKDDR controls the disk, time and zone printed in the heading. Your installation must apply a local modification to DMKDDR to ensure that local time, rather than GMT (Greenwich Mean Time), is printed in the heading.
2. Attempts to restore cylinders or blocks beyond the capacity that had been recorded on the tape produces a successful EOJ, but the printout only indicates the last cylinder or block found on the tape.

PRINT/TYPE Function Statement

Use the PRINT and TYPE function statement to print or type (display) a hexadecimal and EBCDIC translation of each record specified. The first of a group of PRINT or TYPE statements must be preceded by an INPUT statement defining either a direct access device or a tape. The output is directed to the system console for the TYPE function, or to the SYSPRINT device for the PRINT function. (This does not cause redefinition of the output unit definition.) PRINT and TYPE may be used to display the contents of any track including those in the alternate track cylinders. For 3330, 3340, and 3350 devices, the following is displayed when alternate tracks are involved:

- When displaying a defective track that has a properly assigned alternate, the home address record displayed is taken from the defective track while record zero and all other records are taken from the alternate. The “defective” flag, visible in the displayed home address, is the only hint that this is not a normal track.
- When displaying a flagged defective track which does not have a proper alternate, only the home address record and record zero are displayed, and they are both taken from the defective track.
- When displaying an alternate track explicitly, all data displayed is from that track.

You can use a tape in compact format as input to the PRINT or TYPE function. The input format does not affect the printed or displayed output. You do not need to specify anything to the program about the format of the input tape.

The format of the PRINT/TYPE control statement is:

PRint TYpe	<i>cyl1</i> [<i>hh1</i> [<i>rr1</i> .]] [To <i>cyl2</i> [<i>hh2</i> [<i>rr2</i>]]] [(Options... [])]] <i>block1</i> [To <i>block2</i>] Options [Hex] [Graphic] [Count]
-----------------------	---

cyl1

is the starting cylinder.

hh1

is the starting track. If present, it must follow the *cyl1* operand. The default is track zero.

rr1

is the starting record. If present, it must follow the *hh1* operand. The default is home address and record zero.

TO *cyl2*

is the ending cylinder. If more than one cylinder is to be printed or typed, "TO *cyl2*" must be specified.

hh2

is the ending track. If present, it must follow the *cyl2* operand. The default is the last track on the ending cylinder.

rr2

is the record ID of the last record to print. The default is the last record on the ending track.

block1

is the starting FB-512 block number.

To *block2*

is the ending block number. If more than one block is to be printed or typed, "To *block2*" must be specified.

Options:

HEX

prints or displays a hexadecimal representation of each record specified.

GRAPHIC

prints or displays an EBCDIC translation of each record specified.

Service Programs

COUNT

prints or displays only the count field for each record specified. The COUNT option is ignored for FBA data.

Usage Note

If the TYPE statement follows the occurrence of error message DMKDDR705E and specifies the same cylinder, track, and record extents indicated in the error message, the contents of the printed record must be interpreted in the context of the I/O error information given in the initial message.

Examples:

PRINT 0 TO 3

Prints all of the records from cylinders 0, 1, 2, and 3.

PRINT 0 1 3

Prints only one record, from cylinder 0, track 1, record 3. Count-key-data devices print only one record; from cylinder 0, track 1, record 3; FBA devices print blocks 0 to 3.

PRINT 1 10 3 TO 1 15 4

Prints all records starting with cylinder 1, track 10, record 3, and ending with cylinder 1, track 15, record 4.

The example in Figure 4-1 on page 4-18 shows the information displayed at the console (TYPE function) or system printer (PRINT function) by the DDR program. The listing is annotated to describe some of the data fields.

The printed output for FBA data is also provided by the DDR program. The program first prints a heading that lists the block number, and then prints the 512 bytes of data in the block.

Responses

ENTER CYLINDER EXTENTS

ENTER:

-- or --

ENTER BLOCK EXTENTS

ENTER:

These messages are received only if you are entering input from your terminal:

END OF VOLUME CYL xxx HD xx, MOUNT NEXT TAPE

-- or --

END OF VOLUME BLOCK xxxxxxxx, MOUNT NEXT TAPE

DDR continues processing, after the mounting of the next tape reel.

RESTORING volser

volser

is the volume serial number of the disk dumped. The RESTORE operation has begun.

COPYING volser

volser

is the volume serial number described by the input unit. The COPY operation has begun.

DUMPING volser

volser

is the volume serial number described by the input unit. The dumping operation has begun.

PRINTING volser

volser

is the volume serial number described by the input unit. The PRINT operation has begun.

END OF DUMP

The DUMP operation has ended.

END OF RESTORE

The RESTORE operation has ended.

END OF COPY

The COPY operation has ended.

END OF PRINT

The PRINT operation has ended.

END OF JOB

All specified operations have completed.

Enter :

Prompts input from the terminal. A null line (Press the Enter key or equivalent) causes control to return to CMS, if the virtual machine is in the CMS environment.

In addition to the above responses, other informational messages that call for a response by the operator to continue, terminate, or reinitiate the current operation are described in *VM/SP System Messages and Codes*.

For return codes associated with DDR messages, see the *VM/SP CMS Command and Macro Reference*.

Service Programs

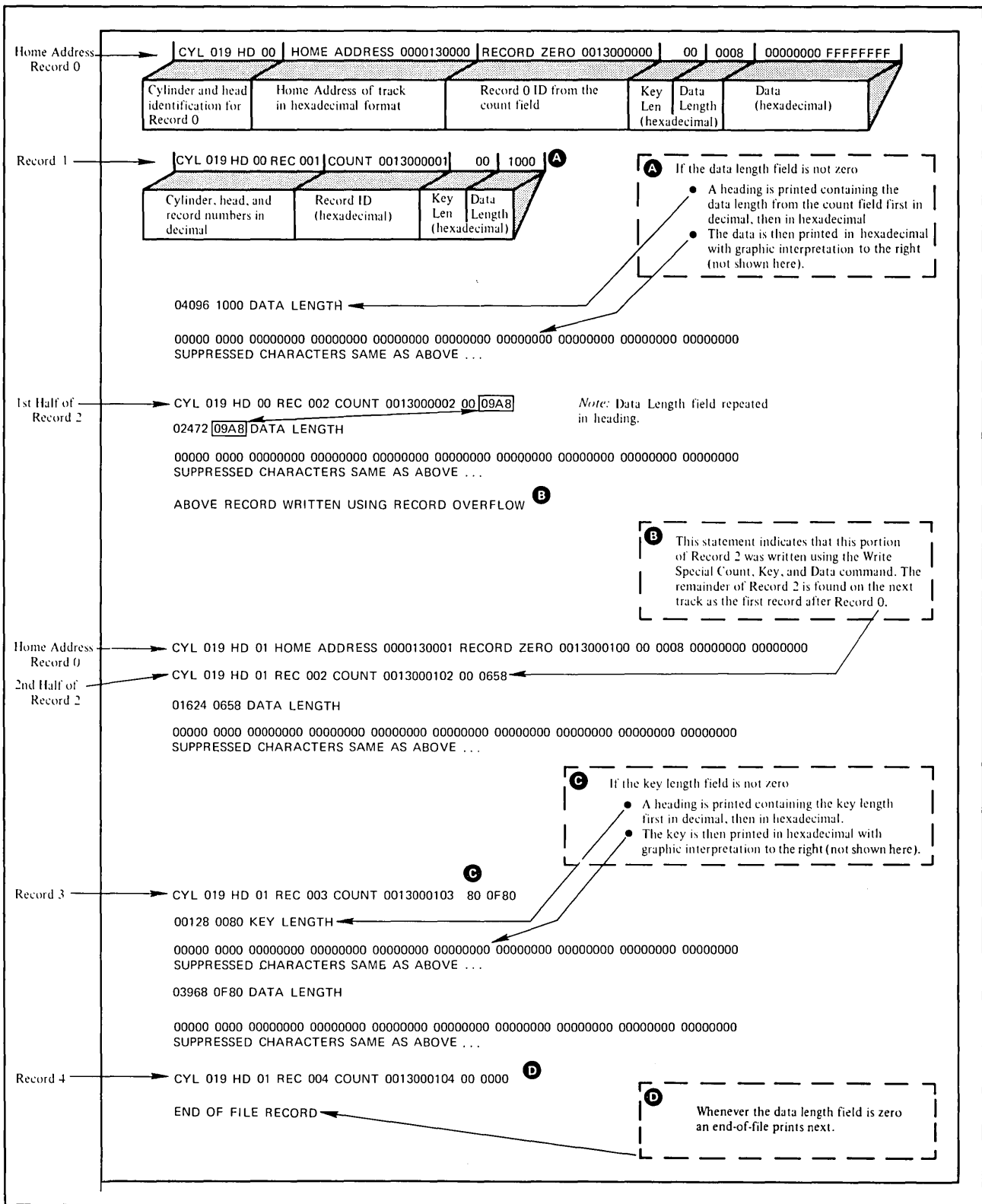


Figure 4-1. Annotated Sample of Output from the TYPE and PRINT Functions of the DDR Program

Maintaining the 3800 Image Library

The Image Library

The 3800 image library contains library character sets (LCSs), graphic character modification modules (GRAPHMODs), character arrangement tables (CATs), forms control buffers (FCBs), and copy modification modules (COPYMODs). These modules are selected and loaded into the 3800 printer based on the parameters specified on the CP START command and control information associated with each spool file. The 3800 modules are created by the GENIMAGE command and are stored in the 3800 image library by the IMAGELIB and IMAGEMOD commands.

Both 3800 model 1 and 3800 model 3 modules may be stored in the same 3800 image library. The 3800 image library must contain the CAT and any GRAPHMODs referenced by that CAT for each character set that is to be used. For the 3800 model 3 only, any LCSs referenced by the CAT must also be loaded in the 3800 image library.

The GENIMAGE Service Routine

The GENIMAGE command invokes the IEBIMAGE program to create the CATs, LCSs, GRAPHMODs, FCBs, and COPYMODs. These are created on a CMS disk with a filetype of TEXT.

GENIMAGE	<i>fn</i> [SYSIN]	<i>ft</i> [FILE]	<i>fm</i> [*]	<i>sfn</i> [SYSPRINT]	<i>sft</i> [LISTING]	<i>sfm</i> [A1]
----------	----------------------	---------------------	------------------	--------------------------	-------------------------	--------------------

fn

specifies the filename of the input control file. The name may be up to eight alphameric characters in length. If not specified, the default filename is SYSIN.

ft

specifies the filetype of the control file. Filetype may be up to eight alphameric characters in length. If not specified, the default filetype is FILE.

fm

specifies the filemode of the control file. Filemode may be up to two alphameric characters in length. If not specified, the default filemode is *.

sfn

specifies the filename into which the messages listing is placed. If not specified, the default file name is SYSPRINT.

Service Programs

sft

specifies the filetype for the messages listing. If not specified, the default file type is LISTING.

sfm

specifies the filemode for the messages listing. If not specified, the default file mode is A1.

Figure 4-2 and Figure 4-3 list the “starter set” of character sets supplied by IBM. They are supplied with a filename and filetype of XTB1xxxx TEXT.

AN	FM12	GU10	ODA	RN
AOA	FM15	GU12	ONA	SN
AOD	GF10	GU15	ONB	TN
AON	GF12	G11	PCAN	TU10
A11	GF15	HN	PCHN	T11
BOA	GN	H11	PN	XN
BON	GS10	KN1	P11	YN
DUMP	GS12	OAA	QN	2773
FM10	GS15	OAB	QNC	2774

Figure 4-2. VM-Supplied Character Arrangement Tables for the 3800 Model 1 and Model 3 Printers

AE10	E ITR	GP12	LR12	SB12
B ITR	ESTR	GR10	OB10	S I10
BRTR	GB10	GSC	OR10	S I12
CE10	GB12	GT10	PB12	S012
CE12	GF10	GT12	P I12	SR12
CO10	GF12	GT15	PR10	ST10
CR10	GF15	GUC	PR12	ST12
DOTR	GFC	LB12	RT10	ST15
EBTR	G I12			

Figure 4-3. VM-Supplied Character Arrangement Tables for the 3800 Model 3 Only

Responses

Completion messages and codes are returned to the user in the file indicated by the *sfn* and *sft* operands of the GENIMAGE control statement. The GENIMAGE command produces two outputs: the listing file (*sft* above) and a TEXT deck for each NAME statement in the input deck. Figure 4-4 on page 4-21 contains the filemode prefixes of these TEXT decks.

	3800 Model 1	3800 Model 3
LCS	LCS1	LCS2
CAT	XTB1	XTB1
GRAPHMOD	GRAF	GRF2
FCB	FCB3	FCB3
COPYMOD	MOD1	MOD1

Figure 4-4. TEXT Deck Names Output by the GENIMAGE Command

For a description of control statements in SYSIN FILE, messages and codes in SYSPRINT LISTING, and the character sets listed above, see the following documents:

- *OS/VS IBM 3800 Printing Subsystem Programmer's Guide* for the 3800 Model 1
- *IBM Printing Subsystem Model 3 Programmer's Guide: Compatibility* for the 3800 Model 3.

The IMAGELIB Service Routine

Before issuing the IMAGELIB command, you must create a control file whose filename is the same as the IMAGELIB you want to build, and whose filetype is CNTRL. The format of this file is one statement per TEXT deck you want included, with the names in columns one through eight.

IMAGELIB reads the control file containing the list of text images. The files are then loaded into the specified named system. The format of the IMAGELIB control statement follows.

IMAGELIB	<i>imagelib</i>
-----------------	-----------------

imagelib

specifies the named system that is being created or replaced.

The IMAGEMOD Command

The IMAGEMOD command allows an installation to make changes to 3800 named systems without the need for generating a completely new named system from scratch. It uses the current contents of the 3800 Named System and adds, deletes, or replaces only those members that you change. The format of the IMAGEMOD command is as follows:

Service Programs

IMAGEMOD	<div style="display: flex; align-items: center;"> <div style="font-size: 3em; margin-right: 10px;">{</div> <div style="display: flex; flex-direction: column; align-items: center;"> <div style="margin-bottom: 5px;">(GEN)</div> <div style="margin-bottom: 5px;">ADD</div> <div style="margin-bottom: 5px;">REP</div> <div style="margin-bottom: 5px;">DEL)</div> </div> <div style="margin-right: 10px;">} imagelib [modname [modname]...]</div> </div> <div style="display: flex; align-items: center;"> <div style="margin-right: 10px;">MAP</div> <div>imagelib [(Options)]</div> </div> <div style="margin-left: 40px;">Options</div> <div style="border: 1px solid black; padding: 2px; display: inline-block; margin-left: 20px;"> TERM PRINT DISK </div>
-----------------	--

GEN

specifies the creation from scratch of the new 3800 named system including the specified module names.

ADD

specifies the addition of the indicated module to an already existing 3800 named system.

REP

specifies the replacement of the indicated member with a new version of that member in an already existing 3800 named system.

DEL

specifies the deletion of an existing member from an already existing 3800 named system. This option causes the named system to then compress so that all members are contiguous with one another.

MAP

specifies the creation of a map of the 3800 named system on the user's terminal, virtual printer, or CMS disk. This map gives the member name, relative byte displacement in both decimal and hexadecimal, and the byte size of the member in both decimal and hexadecimal.

Note: The MAP function does not require the specification of any module names.

TERM

specifies that the map of the 3800 named system be written to the user's terminal.

PRINT

specifies that the map of the 3800 named system be written to the user's virtual printer.

DISK

specifies that the map of the 3800 named system be written to a CMS disk file called "libname MAP A5".

libname

specifies the name of the 3800 named system to be operated upon. Unless the GEN operand is specified, this named system must have been created with at least one member by the IMAGELIB command and must be defined by the installation administrators via the NAME3800 macro in the DMKSNT module.

modname

specifies the name of a module (CAT, LCS, GRAPHMOD, FCB, COPYMOD) that is to be added, replaced or deleted. One to 10 modules may be specified on one IMAGEMOD command.

Device Support Facilities

Device Support Facilities is a program used with IBM operating systems to perform various operations on direct-access storage devices. It replaces IBCDASDI, INITDISK, and SURFANAL in that it can:

- Initialize direct-access storage volumes so that they can be used in OS/VS or DOS/VSE systems
- Inspect a volume for defective tracks
- Reformat the volume label, and IPL bootstrap and program records
- Provide an analysis function for nonremovable DASD, both CKD and FBA.

The CMS FORMAT command initializes minidisks for CMS. The CP format program formats CP-owned volumes, such as the system residence, paging, and spooling volumes.

For information on disk initialization, refer to the *Device Support Facility User's Guide and Reference*. This manual describes procedures for initializing, formatting, and analyzing disk space.

Formatting Volumes--General Information

Disk Initialization

Disk initialization and alternate track assignment should be performed by the Device Support Facility stand-alone utility which is on the CMS system disk with a fileid 'IPL DSF S2'.

Disk Formatting

All direct access volumes used by the VM/SP system (for paging, spooling, system residence, directory, or temporary disk allocation) must be properly labeled, formatted, and allocated. The CP Format/Allocate service program (DMKFMT module) prepares disks for use by CP. A CMS Format program is also available and must be used to format CMS and RSCS disks.

All direct access volumes (including both real disks and minidisks) to be used by VSAM under CMS, OS, or DOS, must be formatted by the device support facility utility.

If certain information in the OS Format 4 label on track 0 cylinder 0 is destroyed, no additional alternate tracks can be assigned by the device support facility until the volume is reformatted by the device support facility. When either the device support facility or the CP Format/Allocate program (IPL FMT) is used to format a volume, this information in the OS Format 4 label is preserved. But, when the CMS Format Program formats the volume (or when it formats a minidisk whose origin is cylinder 0 of the volume) the OS Format 4 label is destroyed and no further alternate tracks can be assigned until the device support facility is used to reformat the volume.

Format/Allocate Service Program (DMKFMT)

The Format/Allocate service program formats, allocates, and labels direct access volumes for paging, spooling and CP file residence. This service program is executed as part of CP system generation procedures and may also be executed as a stand-alone program to:

- Format direct access volumes for CP use
- Allocate specific disk areas to particular functions or to CP use
- Write six-character volume serial number labels.

Note: The Format/Allocate program should be used with care since it destroys existing data (if any). Also, user minidisks and temporary minidisks must not begin

on real cylinder zero of CP-owned volumes, because information critical to CP is stored in that cylinder.

An object deck version of the CP Format/Allocate service program is a standalone program and can be loaded from a virtual or real card reader into a virtual or a real machine. (If run in a virtual machine, the virtual machine must have write access to the volume being formatted.) The program accepts control statements from the operator's system console (commands) or from the IPL device (card reader).

Note: Messages DMKFMT736E and DMKFMT735E may be issued if an available path to the device cannot be found after an appropriate number of retries. High activity may cause this situation.

Cylinders used by CP for paging, spooling, and so on, must be preformatted with fixed length unblocked records of 4096 bytes.

It is important to note the differences in the terminology for fixed-block DASD devices (for example, 3310) and count-key-data DASD devices (for example, 3330).

The basic unit of DASD space used by the VM/SP system is the page. A page contains 4096 bytes of data. On count-key-data devices, a page is recorded in a DASD record data area that is 4096 bytes in size. The count area of this record indicates the location (by cylinder and track) and size (4096 bytes) of the recorded page. The preformatting of count-key-data volumes initializes the count areas and sets the data areas to zero.

On FBA devices, a page is recorded in eight successive blocks of data. Each block contains 512 bytes of a page. Control data, such as the count area for count-key-data, is not required. Preformatting FBA volumes sets all the pages to zero.

The unit of formatting or allocation for count-key-data devices is the cylinder. After you decide how many pages of DASD space are required for system operation, convert the number to a corresponding number of cylinders for communication with DMKFMT.

Device capacity and count-key-data DASD capacities when formatted for CP use (4096-byte pages) are:

2314/2319	32 pages/cylinder	(8 pages/5 tracks)
3330	57 pages/cylinder	(3 pages/track)
2305	24 pages/cylinder	(3 pages/track)
3340	24 pages/cylinder	(2 pages/track)
3350	120 pages/cylinder	(4 pages/track)
3375	96 pages/cylinder	(8 pages/track)
3380	150 pages/cylinder	(10 pages/track)

The format operation writes 4096-byte blocks on all cylinders being formatted. The service program does write-checking to verify that parts of the track are not defective. A count is maintained of pages with read check errors detected during

Service Programs

the format operation. At the completion of the format operation, the count of the pages with read check errors is printed.

For FBA devices, the unit of formatting or allocation is the page.

The format operation writes 4096-byte blocks on all cylinders being formatted. Fixed block device capacity when formatted for CP use is:

3310	15752 pages per spindle
3370, A1 or B1	69750 pages per spindle
3370, A2 or B2	89094 pages per spindle

Format for All Cylinders Except Cylinder Zero

For example, the 3330 track format for all formatted cylinders except cylinder 0 is shown in Figure 4-5.

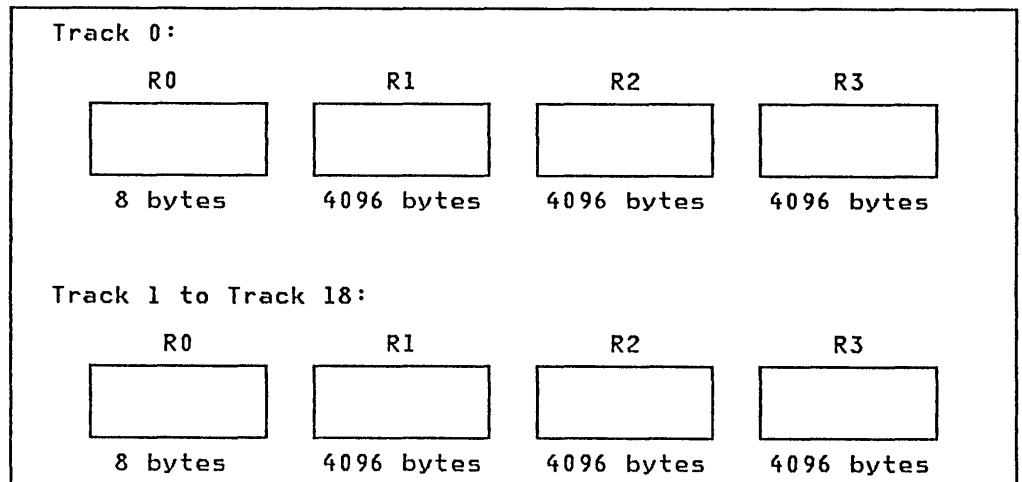


Figure 4-5. Format of 3330 Cylinders for Use by CP

Format for Cylinder Zero

All volumes containing space for CP use (paging, spooling, and so on) must have a properly formatted cylinder 0. The only service program that can do this is the Format/Allocate program (DMKFMT).

Cylinder 0 is formatted like other cylinders except that the space associated with the first three 4096-byte blocks is reserved for system use. This area is then formatted as illustrated in Figure 4-6 on page 4-27.

Service Programs

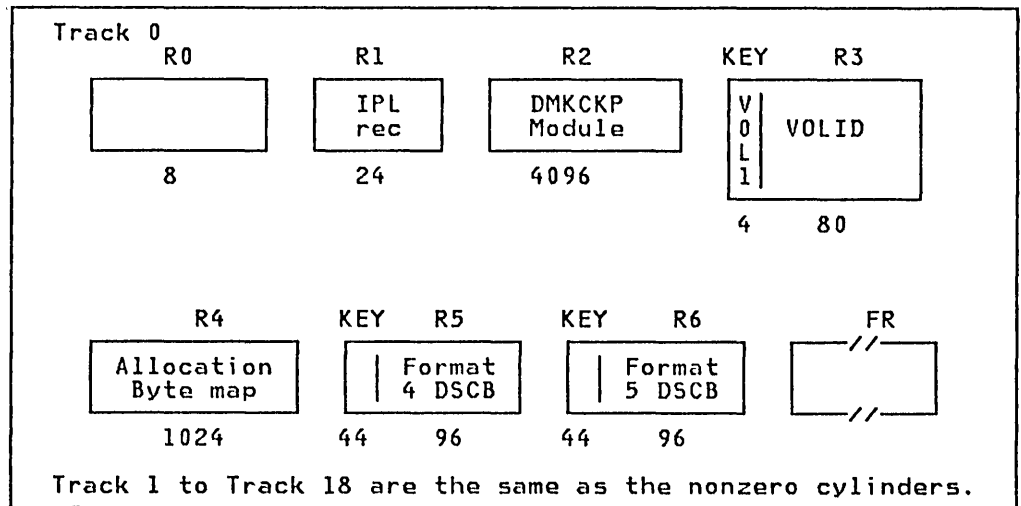


Figure 4-6. 3330, 3340, or 3350 Cylinder 0 Format

The contents of each record in cylinder 0 track 0 are as follows:

R0

Nothing.

R1

IPL record. Puts the system into wait state if storage volume is loaded before CP nucleus is built.

R2

Checkpoint record. Used by CP to save and retrieve information for a warm start.

R3

Volume label. Same as OS VOL1 label. On CP system residence volume, area in data record marks the beginning of the system directory. A label is automatically written when cylinder 0 is formatted. The owner field of the label record contains "CP370" if there is allocation data present in R4.

R4

Allocation Byte Map. Each byte identifies a cylinder and specifies its usage (paging, spooling, directory, and so on). This map is filled in by the ALLOCATE function of the DMKFMT service program.

R5

Format 4 OS DSCB type label. For compatibility with OS. Also, the device support facility program uses this label to keep a record of how many alternate tracks remain available for assignment on this disk. The Format/Allocate program will preserve this information by first reading it from any existing Format 4 label, and then writing it back in the new label.

Service Programs

R6

Format 5 OS DSCB type label. For compatibility with OS. Label indicates to OS that no space is available on this volume.

FR

Is one or more filler records.

Format for CP-Owned FBA Devices

Each block is 512 bytes long. The first 16 blocks (pages 0 and 1) are reserved for system use.

IPL REC	VOLID	VTOC	Allocation Extent Map	DMKCKP	RESERVED	PAGES 2-X
------------	-------	------	--------------------------	--------	----------	--------------

0 1 2 3-4 5-12 13-15 16-N

Each block contains:

0

IPL record. Places the system into the wait state if IPL occurs before the CP nucleus is built.

1

Volume label. Same as DOS/VS1 VOL1 label. On CP system residence volumes, a field in this record contains a pointer to the page where the system directory starts. A label is automatically written when the format function is used, specifying page 0 as the starting page. The owner field of the label record contains "CP370" if there is allocation data in blocks 3 and 4.

2

Volume Table of Contents. The Format 4 and Format 5 DSCB's. These labels are written for compatibility with DOS/VS. They indicate that no space is available on this volume.

3-4

Allocation Extent Map. Each entry is 12 bytes long and describes a range of pages on the device as well as the usage of the pages (PERM, TEMP, DRCT, or TDSK).

5-12

Checkpoint Program. The CP module (DMKCKP).

13-15

Reserved.

16-N

Contains the pages used by the VM/SP system. The area starts at block 16, which corresponds to page 2 on the volume.

Format/Allocate program control statements may be supplied in card form via a card reader, or may be entered at the system console. All error messages regarding improper specification of control statements are displayed at the console.

Format/Allocate Program Card Input

Punch control statements for card input start in column 1, and each field is separated from the adjacent field by a comma. Two commas in a row cause the insertion of a default value. Three commas in a row cause the insertion of two default values.

Note: The only default values permitted are those that define the starting and ending cylinders or DASD extents. The defaults are the first and last cylinders of the volume (or pages), respectively.

Comments must be preceded by at least three blanks.

The control card entries for the Format/Allocate program must be in the following order:

- Format function

FORMAT, devadr, devtype, volser, startadr, endadr

- Allocate function

*ALLOCATE, devadr, devtype, volser
TEMP, startadr, endadr
PERM, startadr, endadr
TDSK, startadr, endadr
DRCT, startadr, endadr
OVRD, startadr, endadr
PAGE, startadr, endadr
DUMP, startadr, endadr
END*

- Label functions

FORMAT, devadr, devtype, volser, LABEL

FORMAT, ALLOCATE, and LABEL are Format/Allocate program control words and may be abbreviated to one letter.

FORMAT Control Statement

The format of the FORMAT control statement is:

FORMAT, devadr, devtype, volser, startadr, endadr

Service Programs

devadr

is a three-digit hexadecimal number that identifies the address of the device that the Format/Allocate program is to act upon. Valid device addresses under CMS are '001' to '5FF' for ECMODE OFF and '001' to 'FFF' for ECMODE ON.

Note: To avoid possible I/O contention when formatting two device types (3375/3380) in two virtual machines, do not use consecutive even/odd pairs of device addresses (e.g., 290,291); rather, it is advisable to use consecutive odd/even pair device addresses (e.g., 291,292).

devtype

is a four-to-seven character field that defines an approved device for the Format/Allocate program.

2314
2319
3330
3330-11
3340-35
3340-70
3350
2305-1
2305-2
3330 for a 3333 device
3340-70 for 3340-70F or 3344 devices
3330 for a 3350 device in 3330-1 compatibility mode
3330-11 for a 3350 in 3330-11 compatibility mode
3375
3380
FB-512 for a 3310 or 3370 device

volser

is a one-to-six character field that represents the volume serial number of the volume you are formatting.

startadr

is the starting cylinder address, and in Basic System Extensions for count-key-data devices, this is the starting cylinder address on the DASD on which the format function is to be performed. For fixed-block devices, this is the starting page number. The starting address is entered as decimal digits.

endadr

is the last cylinder address, and for count-key-data devices, this is the last cylinder address on the DASD on which the format function is to be performed. For fixed-block devices, this is the ending page number. The ending address is entered as decimal digits.

Note: FORMAT is a control word and may be abbreviated to F.

ALLOCATE Control Statements

The formats of the ALLOCATE control statements are:

```
ALLOCATE, devadr, devtype, volser  
TEMP, startadr, endadr  
PERM, startadr, endadr  
TDSK, startadr, endadr  
DRCT, startadr, endadr  
OVRD, startadr, endadr  
PAGE, startadr, endadr  
DUMP, startadr, endadr  
END
```

devadr

is a three-digit hexadecimal number that identifies the address of the device that the Format/Allocate program is to act upon. Valid device addresses under CMS are '001' to '5FF' for ECMODE OFF and '001' to 'FFF' for ECMODE ON.

devtype

is a four-to-seven character field that defines an approved device for the Format/Allocate program. Approved device types are specified as:

```
2314  
2319  
3330  
3330-11  
3340-35  
3340-70  
3350  
2305-1  
2305-2  
3330    for a 3333 device  
3340-70 for 3340-70F or 3344 devices  
3330    for a 3350 device in 3330-1 compatibility mode  
3330-11 for a 3350 in 3330-11 compatibility mode  
3375  
3380  
FB-512  for a 3310 or 3370 device
```

volser

is a one-to-six character field that represents the volume serial number of the volume you are formatting.

startadr

is the starting cylinder address on the DASD on which the format function is to be performed. For fixed-block devices, this is the starting page number. The starting address is entered as decimal digits.

Service Programs

endadr

is the last cylinder address on the DASD on which the format function is to be performed. For fixed-block devices, this is the ending page number. The end address is entered as decimal digits.

TEMP

indicates that the following operands identify temporary storage space reserved for spooling or paging activity.

PERM

defines an area that can contain the logout area, the CP nucleus, and space that is not used by the system but is available for use by virtual machine users (for example, for user minidisks).

TDSK

defines the temporary minidisk space available for virtual machine users during a single terminal session on the VM/SP system.

Whenever spooling space is needed, DMKPGT allocates TEMP storage space for spooling.

DRCT

indicates that the following space is reserved for directory files.

Note: If you wish to reallocate the system directory, it may be necessary to rerun the system directory program (DMKDIR) because the active directory pointer can be destroyed.

OVRD

indicates the space reserved for an override file. This file assigns classes to CP commands to override the IBM-defined command structure. For count-key-data devices, the system only allocates the first cylinder that it finds allocated for OVRD. For FBA devices, the system will use the first extent defined as OVRD. For FBA devices, the minimum allocation is 2 pages.

PAGE

indicates that the following operands identify DASD storage space for preferred paging activity.

Note: Cylinders which are to be used for non-preferred paging space (spooling, overflow paging operations, and so forth) should be allocated as "TEMP."

DUMP

indicates that the appropriate DASD space be marked reserved for CP allocation of system dumps. This reserved area will not be used for CP paging and spooling.

Note: DUMP allocation must be contiguous since CP will look for contiguous cylinders or blocks when assigning the system dump spool file.

END

the allocation termination control statement. This statement causes the allocation record to be written when this is the only control statement. The allocation record is not written back, except for FBA devices which have an allocation record mismatch. For example, a 3370 restored to a 3310 device.

Note: ALLOCATE is a control word that may be abbreviated to its first letter, A.

TEMP, PERM, TDSK, PAGE, DUMP, DRCT, and OVRD are all functions of ALLOCATE. These cards can follow the ALLOCATE control statement in any sequence. Each card in turn overlays the cylinder (or allocate) table, and any space not reallocated remains the same. If an ALLOCATE function overlays the previous allotment, then the previous cylinder space allotment is truncated to the beginning of the next allotment. The allocation operation is executed after you re-ipl VM/SP. For example:

Disk Storage Allocation	First Cylinder	Last Cylinder
1st Entry PAGE	000	002
2nd Entry TEMP	003	202
3rd Entry PERM	010	050
4th Entry TDSK	040	050
5th Entry DRCT	003	004
6th Entry OVRD	005	005
7th Entry END		

The result of this disk volume allocation is:

Disk Storage Allocation	First Cylinder	Last Cylinder
PAGE	000	002
DRCT	003	004
OVRD	005	005
TEMP	006	009
PERM	010	039
TDSK	040	050

Once an ALLOCATE control statement is encountered, all cards following it until an END card is encountered are assumed to be part of a single allocation. The Format/Allocate service functions cannot be performed on another disk volume until the END card is encountered. Any area not allocated will default to "TEMP" space. (See Figure 4-8 on page 4-36.)

Note: Reallocation of an area containing an active VM/SP directory deallocates the directory to allow a new directory to be written in the same area.

After reallocation, the directory program (DMKDIR) must be executed to reinitialize the directory. If this is not done, your program will abend with an ABENDCPI002 when you initialize via IPL.

Service Programs

LABEL Control Statement

The format of the LABEL control statement is:

FORMAT, *devadr*, *devtype*, *volser*, LABEL

devadr

is a three-digit hexadecimal number that identifies the address of the device that the Format/Allocate program is to act upon. Valid device addresses under CMS are '001' to '5FF' for ECMODE OFF and '001' to 'FFF' for ECMODE ON.

devtype

is a four-to-seven character field that defines an approved device for the Format/Allocate program. Approved device types are specified as:

2314
2319
3330
3330-11
3340-35
3340-70
3350
2305-1
2305-2
3330 for a 3333 device
3340-70 for 3340-70F or 3344 devices
3330 for a 3350 device in 3330-1 compatibility mode
3330-11 for a 3350 in 3330-11 compatibility mode
FB-512 for a 3310 or 3370 device
3375
3380

volser

is a one-to-six character field that represents the volume serial number.

LABEL

is a keyword designating the label function of the Format/Allocate program.

Notes:

1. Format of cylinder 0 is a prerequisite for the label only function.
2. FORMAT and LABEL are control words and may be abbreviated to FORM and L, respectively.

If the volume is to be used for paging, spooling, dump, directory, override, or temp space, 'CP370' is required in record 3 for CKD or block 1 for FB-512. The label

option does not write it out; the FORMAT function must be used to format cylinder 0 for CKD or page 0 for FB-512 to insure the inclusion of the 'CP370'.

Examples:

FORMAT

```
FORMAT,232,3330,MYDISK,000,006
FORMAT,232,3330,MYDISK,,,
FORMAT,232,3330,MYDISK,,00
FORMAT,232,3330,MYDISK,001,,
```

ALLOCATE

```
ALLOCATE,232,3330,MYDISK
PAGE,000,020
TEMP,021,150
PERM,055,060
TDSK,100,108
OVRD,109,109
DRCT,110,120
DUMP,121,150
END
```

LABEL

```
F,232,3330,MYDISK,label
```

Format/Allocate Console Input

The Format/Allocate program can be controlled by control statements entered into the real or virtual console instead of by a deck of cards containing control statements. If the program finds no control statements at the card reader, it issues a prompting message to the console. The proper response causes the prompting message for the next operand to appear until the Format, Allocate, or Label function is completely defined; then the Format/Allocate program is executed.

After execution, the prompting begins again until all DASD allocation requirements are fulfilled.

The sequence for console typewriter processing of the Format/Allocate program (after making the operator's console ready), is as follows:

1. Load the card reader with a loader, followed by the Format/Allocate deck
2. IPL the card reader
3. Respond to the first message displayed at the system console
4. Respond to other messages.

Service Programs

Note: If you are in a virtual machine environment, you must re-IPL CMS to exit from the Format/Allocate program.

Following are examples of Format/Allocate program execution under CP control. Figure 4-7 is an example of the label operation. Figure 4-8 is an example of the allocate operation. Figure 4-9 on page 4-37 is an example of the allocate overlap operation. All responses are entered after the colon, except for the 3270 device. After a function is complete, the program returns and issues again the ENTER "FORMAT" OR "ALLOCATE": statement.

```
VM/SP FORMAT/ALLOCATE PROGRAM
ENTER "FORMAT" OR "ALLOCATE": f
FORMAT FUNCTION SELECTED
ENTER DEVICE ADDRESS (CUU): 131
ENTER DEVICE TYPE: 2314
ENTER START CYLINDER (XXX) OR "LABEL": 1
ENTER DEVICE LABEL: cpdsk2
```

Figure 4-7. Using the Format Program Label Function

```
ENTER "FORMAT" OR "ALLOCATE": a
ALLOCATE FUNCTION SELECTED
ENTER DEVICE ADDRESS (CUU): 131
ENTER DEVICE TYPE: 2314
ENTER DEVICE LABEL: cpdsk2
ENTER ALLOCATION DATA FOR VOLUME CPDSK2
TYPE CYL CYL
.....
drct 000 001
ovrd 002 002
perm 004 008
page 009 070
dump 071 099
tdsk 100 150
end
ALLOCATION RESULTS
DRCT 000 001
OVRD 002 002
TEMP 003 003
PERM 004 008
PAGE 009 070
DUMP 071 099
TDSK 100 150
TEMP 151 202
DEVICE 131 VOLUME CPDSK2 ALLOCATION ENDED
```

Figure 4-8. Using the Format Program Allocate Function

```
ENTER "FORMAT" OR "ALLOCATE": a
ALLOCATE FUNCTION SELECTED
ENTER DEVICE ADDRESS (CUU): 131
ENTER DEVICE TYPE: 2314
ENTER DEVICE LABEL: cpdsk2
ENTER ALLOCATION DATA FOR VOLUME CPDSK2
TYPE CYL CYL
.....
perm 004 004
temp 000 010
tdsk 000 010
perm 010 202
drct 000 004
ovrd 005 005
end
ALLOCATION RESULTS
DRCT 000 004
OVRD 005 005
TDSK 006 009
PERM 010 202
DEVICE 131 VOLUME CPDSK2 ALLOCATION ENDED
```

Figure 4-9. Using the Format Program Allocate Overlap Function

Figure 4-10 shows the label function for FBA devices. Figure 4-11 on page 4-38 shows the allocate function for FBA devices. Figure 4-12 on page 4-39 shows the allocate overlap function for FBA devices.

```
VM/SP FORMAT/ALLOCATE PROGRAM

ENTER "FORMAT" OR "ALLOCATE": f
FORMAT FUNCTION SELECTED
ENTER DEVICE ADDRESS (CUU): 131
ENTER DEVICE TYPE: FB-512
ENTER START PAGE NUMBER OR "LABEL": 1
ENTER DEVICE LABEL: cpdsk2
```

Figure 4-10. Using the Format Program Label Function for FBA Devices

Service Programs

```
ENTER "FORMAT" OR "ALLOCATE": a
ALLOCATE FUNCTION SELECTED
ENTER DEVICE ADDRESS (CUU): 131
ENTER DEVICE TYPE: FB-512
ENTER DEVICE LABEL: cpdisk2
ENTER ALLOCATION DATA FOR VOLUME CPDSK2
TYPE PAGE PAGE
.... ... ..
perm 2 1000
temp 1001 5000
drct 5001 5100
ovrd 5101 5108
tdsk 5109 9000
end
ALLOCATION RESULTS
PERM 2 1000
TEMP 1001 5000
OVRD 5101 5108
DRCT 5001 5100
TDSK 5109 9000
TEMP 9001 15751
DEVICE 131 VOLUME CPDSK2 ALLOCATION ENDED
```

Figure 4-11. Using the Format Program Allocate Function for FBA Devices

```

ENTER "FORMAT" OR "ALLOCATE": a
ALLOCATE FUNCTION SELECTED
ENTER DEVICE ADDRESS (CUU): 131
ENTER DEVICE TYPE: FB-512
ENTER DEVICE LABEL: cpdsk2
ENTER ALLOCATION DATA FOR VOLUME CPDSK2
TYPE PAGE PAGE
.... .....
temp 2 500
page 9500 11000
end
ALLOCATION RESULTS
TEMP 2 500
PERM 501 1000
TEMP 1001 5000
DRCT 5001 5100
OVRD 5101 5108
TDSK 5109 9000
TEMP 9001 9499
PAGE 9500 11000
TEMP 11001 15751
DEVICE 131 VOLUME CPDSK2 ALLOCATION ENDED
-----
ENTER "FORMAT" OR "ALLOCATE": a
ALLOCATE FUNCTION SELECTED
ENTER DEVICE ADDRESS (CUU): 131
ENTER DEVICE TYPE: FB-512
ENTER DEVICE LABEL: cpdsk2
ENTER ALLOCATION DATA FOR VOLUME CPDSK2
TYPE PAGE PAGE
.... .....
perm 0 1000
LOWEST ALLOCATABLE PAGE IS PAGE 2 -- RESPECIFY
perm 2 17345
HIGHEST ALLOCATABLE PAGE IS PAGE 15751 -- RESPECIFY
perm 2 1000
end
ALLOCATION RESULTS
PERM 2 1000
TEMP 1001 5000
DRCT 5001 5100
OVRD 5101 5108
TDSK 5109 9000
TEMP 9001 9499
PAGE 9500 11000
TEMP 11001 15751
DEVICE 131 VOLUME CPDSK2 ALLOCATION ENDED

```

Figure 4-12. Using the Format Program Allocate Overlap Function for FBA Devices

Note that before the ALLOCATE function was invoked, cylinder 0 was formatted and labeled CPDSK2. The area associated with the first three 4096-byte blocks on cylinder 0 are not used for spooling but contain system information (page allocation map, label, and so on).

These CP-formatted volumes can be made usable by CP in one of two ways:

Service Programs

1. They may be attached to the system by the VM/SP operator.
2. Their volume serial numbers may appear in the SYSOWN macro in the DMKSYS module. The CP system residence volume's serial number must appear in the SYSOWN macro.

How To Print a CMS Dump File

Use the PRTDUMP command to print a previously created dump file under CMS by entering:

```
PRTDUMP PRBnnnnn [ options]
```

at the terminal. The message:

```
PRINTING FILE 'PRBnnnnn Dnn DUMP'
```

then occurs at the terminal. When the specified file has been formatted and printed, one of two completion messages is displayed:

```
DUMP FILE 'PRBnnnnn Dnn DUMP' PRINTED AND KEPT  
DUMP FILE 'PRBnnnnn Dnn DUMP' PRINTED AND ERASED
```

How To Print a CP ABEND Dump from Tape

When the CP abend dump is sent to a tape, the records are 132 characters long, unblocked, with a blocksize of 132 and carriage control characters.

If the CP dump unit has been specified as a tape drive, and one or more dumps have been placed on the tape, use the following procedure to print the dumps.

1. Log on to the VM/SP system with any userid that has the capability of running CMS. No other special privilege classes or options are required.
2. Attach a tape drive to the virtual machine as address 181.
3. Mount the tape that has the CP abend dumps.
4. IPL the CMS system and perform the usual access requirements.
5. Issue the following CMS commands:

```
FILEDEF ddname1 PRINTER (RECFM UA LRECL 132)  
FILEDEF ddname2 TAP1 (optionC RECFM U LRECL 132)  
MOVE ddname2 ddname1  
cp close prt
```

Note: Refer to the FILEDEF command description in *VM/SP CP Command Reference for General Users*.

Step 5 can be repeated for as many dumps as are on the tape. Note that the CP dump routines write two tape marks at the end of each file. Therefore, to process the next dump, the TAPE FSF command line must be issued to position the tape for reading the next dump file.

Network Dump Operations

This section only applies to 3704 or 3705 communication controllers that have been loaded by VM/SP. If you want to dump the contents of a 3725 or a 3705 that has been loaded by ACF/SSP, refer to *ACF/NCP V4, ACF/SSP V3 Diagnosis Guide*, SC30-3255.

If 3704/3705 operations are erratic, fatal hardware errors occur, or some other internal error appears, the Communications Controller's storage should be dumped. The NETWORK DUMP command dumps the contents of 3704/3705 storage for NCP, PEP, or EP 3704/3705 control programs, if unit check or IPL required conditions are detected.

The format of the NETWORK command with the DUMP operand is:

NETWORK	DUMP <i>raddr</i> [IMMED AUTO OFF]
---------	--

raddr

is the real hexadecimal address of the 3704/3705.

IMMED

is the default operand; it forces an immediate dump. The IMMED operand, if specified, does not reload the control program. Before 3704/3705 resources can be used again, the control program must be reloaded. To reload the control program after the NETWORK DUMP *raddr* IMMED command has executed, use the NETWORK LOAD *raddr* *ncpname* command.

If the IMMED operand is specified, a check is made to determine whether the "IPL required" sense status is present. If it is not, the following message occurs:

CTLR *raddr* IPL NOT REQUIRED; ENTER YES TO CONTINUE:

This pause in operations allows the operator an opportunity to check the NETWORK DUMP command line before engaging or terminating the operation.

Service Programs

AUTO

causes a dump if VM/SP subsequently detects a unit check condition or "IPL required" condition. If AUTO is specified, each time a dump is taken, the Communications Controller is reloaded with the 3704/3705 control program that was previously active.

OFF

resets a previously set AUTO (automatic dump) status.

Note: The dumps produced by the NETWORK command cannot be processed by the IPCSDUMP service program. NETWORK-initiated dumps are processed by the NCPDUMP (Network Control Program DUMP) service program created for this task.

NCPDUMP Service Program and How To Use It

NCPDUMP applies only to dump files that were dumped with the NETWORK DUMP command after the 3704/3705 was loaded by VM/SP.

NCPDUMP is a CMS command. It processes CP spool reader files created by 3705 dumping operations, that is, dump files that are produced as a result of the CP NETWORK command specified with the DUMP operand and either automatic or immediate mode.

The NCPDUMP file processing operation can include:

- Erasing a specific CMS NCPDUMP file after printing it
- Formatting the dump
- Printing the dump
- Assigning an identifier to the CMS NCPDUMP file
- Creating the CMS NCPDUMP file from the spool file.

Although NCPDUMP is a CMS command, its use is restricted to the user identified by the SYSDUMP operand of the SYSOPER macro in DMKSYS during VM/SP system generation. The operation of NCPDUMP is similar to IPCSDUMP operations. A general description of the NCPDUMP operation follows the command description.

The NCPDUMP command has the following format:

NCPDUMP	[DUMP _{xx}] [([ERASE] [NOFORM] [NCPBUFF] [])]
---------	---

DUMP_{xx}

is the filename of a CMS file containing a 3704/3705 Communications Controller program dump. This dump was created by a previously invoked NCPDUMP command with the ERASE operand not specified.

ERASE

erases the current CP DUMP file or a specified DUMPxx (filename), saved CMS file.

NOFORM

specifies that a formatted control block is not desired.

NCPBUFF

specifies that a formatted listing of the NCP buffer pool is desired.

The NETWORK command invoked with the DUMPxx operand, as stated previously, produces CP files that contain the contents of a designated 3704/3705 Communications Controller unit buffer. These CP files reside as a spooled reader input assigned to a system designated user. The CMS NCPDUMP command invoked by this user formats (if requested) and prints the contents of these files.

The NCPDUMP program creates a CMS file with a filename DUMPxx and a filetype of NCPDUMP, and erases the original spooled NETWORK initiated dump reader file. The created CMS file is erased if you specify ERASE; otherwise it is kept.

A maximum of ten dumped spooled files can be processed and saved, and later recalled, if necessary, by the system assignment of an xx identifier suffix to the CMS DUMPxx filename. The "xx" is a decimal number from 00 to 09, depending on any existing files of a similar name. For example, if the files DUMP00 NCPDUMP and DUMP01 NCPDUMP already exist, the new file would be called DUMP02 NCPDUMP. The file thus created is retained for later use unless the ERASE option is specified, in which case the file is erased immediately after the dump is printed.

ZAP Service Program and How To Use It

ZAP is a CMS command that modifies or dumps MODULE, LOADLIB, or TXTLIB files. It may be used to modify either fixed or variable length MODULE files. It is for use by system support personnel only.

Input control records control ZAP processing. They can be submitted either from the terminal or from a disk file. Using the VER and REP control records, you can verify and replace data or instructions in a control section (CSE CT). Using the DUMP control record, you can dump all or part of a CSECT, or an entire member of a LOADLIB or TXTLIB file, or an entire module of a MODULE file.

Service Programs

The format of the ZAP command is:

ZAP	$\left\{ \begin{array}{l} \text{MODULE} \\ \text{LOADLIB} \\ \text{TXTLIB} \end{array} \right\} [\text{libname1} \dots \text{libname3}] [(\text{Option} \dots)]$ <p style="text-align: center;">Options</p> $\left[\begin{array}{l} \text{TERM} \\ \text{INPUT filename} \end{array} \right] \left[\begin{array}{l} \text{PRINT} \\ \text{NOPRINT} \end{array} \right]$
------------	---

MODULE
LOADLIB
TXTLIB

indicates the type of file that is to be modified or dumped.

For LOADLIB files, the maximum blocksize supported by the ZAP service program is 1024 bytes.

libname

is the library name containing the member to be modified or dumped. You can specify one to three library names. The libname is valid only for LOADLIB and TXTLIB files.

Options:

TERM $\left[\begin{array}{l} \text{PRINT} \\ \text{NOPRINT} \end{array} \right]$

indicates that input to the ZAP service program is submitted through the terminal. If you specify TERM, the prompting message ENTER: is issued, and you can then enter input control records up to 80 characters long. If you specify PRINT with TERM, all output prints on the printer, but only error messages display at the terminal. If you specify NOPRINT with TERM, nothing prints on the printer. All output except control records displays at the terminal.

INPUT filename $\left[\begin{array}{l} \text{PRINT} \\ \text{NOPRINT} \end{array} \right]$

specifies that input is submitted from a disk file, filename. This file must have a filetype of ZAP, and must be a fixed 80-byte sequential file residing on any accessible device. If you specify PRINT with INPUT filename, all output produced by the ZAP service program prints on the printer. In addition, commands and control records in error and error messages display at the terminal. If you specify NOPRINT with INPUT filename, nothing prints on the printer. Selected output produced by the ZAP service program prints on the printer. All output displays at the terminal.

Figure 4-13 shows the resulting output resulting from valid option combinations:

Options	PRINT	NOPRINT
INPUT	Commands and control records in error and error messages on the terminal. Everything on the printer.	Everything on the terminal. Nothing on the printer.
TERM	Only error messages on the terminal. Everything on the Printer.	Everything except control records on the terminal. Nothing on the printer.

Figure 4-13. Valid Options and Their Output

ZAP Input Control Records

Eight types of ZAP control records exist: DUMP, NAME, BASE, VER or VERIFY, REP, LOG, COMMENT, and END.

ZAP control records are free-form and need not start in position one of the record but the ZAP program can accept only 80 characters of data for each control record. Separate all information by one or more blanks. All address fields including disp (displacement) fields in VER and REP control records must contain an even number of hexadecimal digits, to a maximum of six digits (OD, 02C8, 014318). Data fields in VER and REP control records must also contain an even number of hexadecimal digits.

If you wish, you may separate the data anywhere by commas (for example, 83256482 or 8325,6482). The commas have no effect on the operation.

Note: Do not use blank spaces as separators within data fields.

The program sets the NOGO switch on if a control record is found to be in error. A file cannot be modified once the NOGO switch is turned on. The next valid NAME record turns the NOGO switch off. This means that if the control record is the NAME record, all succeeding records are ignored until the next NAME, DUMP, or END record. For any other error, only REP control records that follow are ignored.

DUMP Control Record

The DUMP control record resets the NOGO switch off. The DUMP control record must not immediately precede a BASE, VER, or REP control record. A NAME control record must precede the BASE, VER, and REP control records (if any) that follow a DUMP control record.

The DUMP control record allows you to dump a portion or all of a specified control section, or the complete member or module. The format of the output of the dump is hexadecimal with an EBCDIC translation of the hexadecimal data.

Service Programs

The DUMP control record is optional. The format of the DUMP control record is:

$\text{DUMP } \left\{ \begin{array}{l} \text{membername} \\ \text{modulename} \end{array} \right\} \left[\begin{array}{l} \text{csectname } [\text{startaddress } [\text{endaddress}]] \\ \text{ALL} \end{array} \right]$
--

membername

is the name of the member to be dumped, or the member that contains the CSECT(s) to be dumped. This member must be found in one of the libraries specified in the ZAP command line.

For a CMS TXTLIB, the format of the dump control record requires that you specify both a member name and a CSECT name. Because the library directory does not contain member names, any word may be used to replace membername. The program searches for only the second name following the dump operand; therefore, the second name must be the CSECT name.

modulename

is the name of the module to be dumped, or the module that contains the CSECT(s) to be dumped. If you specify a module that has no loader table, the program dumps the entire module.

csectname

is the name of the control section that is to be dumped. If you do not specify csectname, the program dumps only the first CSECT. The csectname is required for CMS TXTLIBs, optional for OS TXTLIBs, LOADLIBs, and MODULE files. (See the discussion of csectname in "NAME Control Record" on page 4-47). You must not specify csectname for a module created with the NOMAP option.

ALL

specifies to the program to dump all CSECTs within the specified member or module. You can specify ALL for MODULE files, LOADLIBs, and OS TXTLIBs, but not for CMS TXTLIBs. If you wish to dump all the CSECTs in a member of a CMS TXTLIB, you must issue a separate DUMP control record for each CSECT.

startaddress

is the location within the specified CSECT where the dump is to begin. This must be two-, four-, or six-hexadecimal digits. The start address is the displacement from the beginning of the CSECT. For example, if you wish to start dumping at address 08 in a CSECT that begins at location 400, you specify start address 08, not 0408.

endaddress

is the last address to be dumped. This must be two, four, or six-hexadecimal digits. If you specify no address, the program dumps from the start address of the CSECT to the end of the CSECT. Note that start and end addresses apply only when you specify a csectname.

Notes:

1. Displacements listed in the dump output for a module file are calculated from the beginning location of the module. Therefore, the addresses in the output may differ from the displacements within a CSECT.
2. If a DUMP control record references a TXTLIB file that contains ORG statements causing more than one occurrence of an address, data found at the first occurrence will be displayed, but any subsequent redefinition of data for the same location will be ignored.

NAME Control Record

The NAME control record specifies the member or module and CSECT that contain the data to be verified or replaced by the ZAP operation. The NAME control record must precede the BASE, VER, and REP control records. If it does not, the program sets the NOGO switch on. The format of the NAME control record is:

NAME { <i>membername</i> } [<i>csectname</i>] { <i>modulename</i> }

{ *membername* }
{ *modulename* }

is the member or module that you want to be searched for the desired CSECT.

csectname

is the name of the desired control section.

You must specify *csectname* if the CSECT you wish to modify is in a CMS TXTLIB (that is, TXTLIB created by the TXTLIB command from CMS TEXT decks that do not have a NAME card following the END card). The directory of a CMS TXTLIB contains only CSECT names and no member names. Select any word to replace *membername* as the first entry following the NAME operand in the NAME statement for a CMS TXTLIB. The CSECT name specified in the NAME record is compared with CSECT names in the directory. If a CSECT match is found and no member name match is found, the member selected is the one that contains the CSECT name.

The *csectname* is optional if the CSECT you wish to modify is a LOADLIB or an OS TXTLIB (that is, a TXTLIB created by the TXTLIB command from CMS TEXT decks that have a NAME card after the END card). The dictionaries of the specified libraries are searched for the member name and the member is then searched for the CSECT name, if you specified one. If you do not specify *csectname* for a LOADLIB or an OS TXTLIB, the program uses the first control section.

Service Programs

The csectname is optional for a MODULE file. The module named in the NAME control record is located and, if you specified csectname, the first record is read to determine the number of records in the module and the availability of a loader table, which the program can then search for the csectname. If you do not specify csectname, the program uses the beginning location of the module.

You are not allowed to specify csectname if the module was created with the NOMAP option.

Note: The word selected to replace membername for a CMS TXTLIB should be a meaningful name because the filename of the LOG control record is determined by the specified membername or modulename in the NAME control record.

BASE Control Record

The BASE control record adjusts displacement values for subsequent VER or REP control records for a CSECT whose starting address is not location zero in an assembly listing. The format of the BASE control record is:

BASE <i>address</i>

address

is the starting address of the CSECT. The address must be two-, four-, or six-hexadecimal digits. For example, for a CSECT starting at location 400, you would specify the BASE 0400 in the BASE control record. If a subsequent VER card requests verification of location 0408, the BASE of 0400 is subtracted from 0408, and the program verifies location 08 in the CSECT. This example applies if you specify TXTLIB, LOADLIB, or MODULE and the module map is present. However, if no module map is present for a MODULE file (that is, the module was generated with the NOMAP option), then all operations are performed as if the BASE address is location 0. For example, if you specify a BASE of 400 and the address you wish to inspect or modify is 408, then you must specify 08 and not 408 in REP and VER control records. The address in this case is from the start of the module. If you do not specify csectname in the NAME control record, you cannot specify any BASE value other than 00. The BASE control record is optional. See the discussion under "VER or VERIFY Control Record" on page 4-49. If specified, the BASE control record must follow the NAME record, but it need not follow the NAME record immediately. For example, you could have the following sequence of control records: NAME, VER, REP, BASE, VER, REP.

VER or VERIFY Control Record

The VER control record requests verification of instructions or data within a CSECT. If the verification fails, the program does not perform a subsequent REP operation until it encounters another NAME control record.

The VER control record is optional. More than one VER record can follow a single NAME record.

The format of the VER control record is:

$\left\{ \begin{array}{l} \text{VERIFY} \\ \text{VER} \end{array} \right\} \text{ disp data}$

disp

is the displacement from the start of the CSECT containing the data to be inspected if you did not submit a BASE control record for this CSECT, or *disp* is the actual location of the data to be inspected if you did submit a BASE control record. *disp* must be two, four, or six hexadecimal digits. This displacement does not have to be aligned on a fullword boundary. If this displacement value is outside the limits of the CSECT specified by the preceding NAME control record, the VERIFY control record is rejected.

data

is the data against which the data in the CSECT is to be compared. This must be an even number of hexadecimal digits.

Note: If the VER control statement references data in a TXTLIB file that is later redefined by ORG statements, only the first data definition is verified.

Example: If the location you wish to verify is 3CC, and the CSECT begins at location 2B0, you can enter:

```
BASE 02B0  
VER 03CC data
```

or you can omit the BASE control record, subtract the CSECT start address from the address of the data, and enter:

```
VER 011C data
```

REP Control Record

The REP control record modifies instructions or data at the specified location within the CSECT that you specified in a preceding NAME control record. The data specified in the REP control record replaces the data at the CSECT location specified by the *disp* operand. This replacement is on a “one-for-one” basis; that is, one byte of data defined in the control record replaces one byte of data at the location that you specified. If the replacement fails, the program does not perform additional REP operations until it encounters another NAME control record.

Service Programs

The REP control record is optional. More than one REP record can follow a single NAME record.

The format of the REP control record is:

REP <i>disp</i> <i>data</i>

disp

is the displacement from the start of the CSECT of the data to be replaced if you did not submit a BASE control record for this CSECT, or *disp* is the actual location of the data if you did submit a BASE control record. *disp* must be two, four, or six hexadecimal digits. This displacement need not address a fullword boundary. If this displacement value is outside the limits of the CSECT being modified, the program does not perform the replacement operation.

data

is the data that is to replace the data in the CSECT. This must be an even number of hexadecimal digits.

Notes:

1. Although you do not have to verify a location before replacing data, you should do so to make sure that the data being changed is what you expect it to be.
2. If the REP control statement references data in a TXTLIB file that is later redefined by ORG statements, the replacement of data will take place at the first occurrence of the data address of the TXTLIB member.

Example: If the location you wish to replace is 3CC, and the CSECT begins at location 2B0, you can enter:

```
BASE 02B0  
REP 03CC data
```

or you can omit the BASE control record, subtract the CSECT start address from the address of the data, and enter:

```
REP 011C data
```

LOG Control Record

The LOG control record allows the user to specify, after application of a fix, a unique fix number which is recorded in a log file for the module or member. The file name of the log file is the same as the membername or modulename in the NAME control record. The format of the LOG control record is:

```
LOG  fixnum [ filetype [user data]
              ZAPLOG ]
```

fixnum

specifies the number associated with the fix. Its length may vary from one-to-eight alphameric characters.

filetype

specifies the file type of the log. If not specified, filetype defaults to ZAPLOG.

user data

specifies any data that the user wishes to enter into the log. If user data is specified, the filetype operand is mandatory.

Usage Notes

Notes:

1. The LOG control record is optional and is allowed only if valid NAME and REP control records are found. The file name is obtained by the log routine from the module or member named in the NAME control record. However, whenever a LOG control record is found, a dummy log record will be written at the completion of the user's valid REPs.
2. Log multiple names by including a LOG control record after each name. If the LOG record is not included after each name, error message DMSZAP070E results. Processing continues after the error messages occur.
3. The LOG record is 80 bytes in length and has the following information.

Columns 1 - 63: The fixnum and, if specified, the filetype and user data.
Columns 64-80: The date and time of the ZAP.

Comment Control Record

The ZAP program ignores comment control records. If the PRINT option is in effect, the program prints the comments. The format of a comment record is:

```
* comment
```

The asterisk must be followed with at least one blank before you enter the text.

Service Programs

END Control Record

The END control record ends ZAP processing. The END record is required and must be the last control record for input from the console. The format of the END control record is:

END

Special Considerations for Using the ZAP Service Program

Before using the ZAP command against MODULE files, you can use the MODMAP command to determine whether a module map exists and what it contains.

When a ZAP input file has more than one pair of VER and REP control records and a VER control record (other than the first) fails, you must remove the records prior to the failing record and correct the error before you issue the ZAP command again. Otherwise, the file being modified returns to its original status.

The REP control record cannot be used to place data in an undefined area such as a Define Storage area. If any part of a data field specified in a pair of VER and REP control records is an undefined area, warning message DMSZAP248W will be displayed, and no data replacement will occur. If you do not issue a VER control record prior to the REP control record, partial modification of data may result. User-defined data may be inserted in undefined areas of text files by using the REP statement described under the LOAD command. (See the *VM/SP CMS Command and Macro Reference*.)

If the file to be dumped contains undefined areas (such as a DS or ORG statement in a TXTLIB member), the hexadecimal portion of the dump contains blanks to indicate that the corresponding positions are undefined.

VER and REP control words can be used to modify TXTLIB members produced by FORTRAN compilers that store the length of the compiled text in the END card rather than in the ESD card. However, if a member of this type contains multiple CSECTs, only the first CSECT can be modified by the ZAP program.

ZAPTEXT Service Program

ZAPTEXT is a CMS command that modifies or dumps individual text files. Use ZAPTEXT like the ZAP service program, but only for text files, not MODULES, TXTLIBs, or LOADLIBs, which ZAP processes.

The format of the ZAPTEXT command is:

ZAPTEXT	<i>fn</i> [<i>ft</i> [<i>fm</i>]] [(Options....[])] Options: [<u>INPUT</u> <i>filename</i>] [<u>PRINT</u> <u>NOPRINT</u>]
----------------	---

fn ft fm

is the fileid of the text file that you want to alter. If you omit the filetype or filemode, the system assumes TEXT for the filetype and A1 for the filemode. The filemode must specify a read/write disk.

Options:

INPUT *filename*

filename is the name of the file that has the ZAP control records. This file must:

- Have a filetype of ZAP
- Be a fixed 80-byte sequential file that resides on any accessible disk.

If you do not specify a filename, it defaults to *fn*.

PRINT

prints all output produced by ZAPTEXT on the printer. The system also displays error messages, commands in error, and control records in error.

NOPRINT

does not print anything on the printer, and instead displays all output at the terminal.

ZAPTEXT Input Control Records

ZAPTEXT uses the same control information as the current ZAP service program, with the addition of the EXPAND control record. The ZAP service program will ignore any EXPAND control records. Refer to "ZAP Input Control Records" on page 4-45 for information about the control records, other than EXPAND, that ZAPTEXT uses.

Use the ZAP control records with ZAPTEXT according to ZAP's TXTLIB conventions.

Service Programs

EXPAND Control Record

The EXPAND control record lets you increase the size of a named control section contained in the text file. The format of the EXPAND control record is:

EXPAND *csect size* [, *csect size* ...]

csect

specifies the symbolic name of a control section whose length you want to increase.

size

specifies the decimal number of bytes for the system to add to the control section length. The system will initialize the added bytes to binary zeros. The maximum number of bytes for each control section that you indicate is 4095.

Each control record may have multiple entries, but you must separate them with commas. Do not spill an entry onto the next line.

Notes:

1. The system processes all EXPAND control records before any other control records, regardless of their position in the control file.
2. The effective length of the expansion, which is the actual number of bytes added to the control section, may be greater than the length that you specify for the expansion. This may occur if, after the specified expansion, the system must add padding bytes to align the next control section or common area.
3. When you increase a control section's size, it may affect the offset address of any following control section. This is important when you determine values for BASE, REP, and VER control records. Use the effective expansion lengths when you are determining control section offsets.

EXPAND Command

ZAPTEXT calls EXPAND if you specify an EXPAND control record in the ZAP control file. Use EXPAND to add space to a program in object deck form. The system creates object decks when you assemble or compile a source program. This is especially useful when you do not have the source code for a program or the program does not have a patch area.

Note: EXPAND can only add extra space at the end of named control sections (csects). EXPAND cannot expand private code (unnamed csect) and common areas (named or unnamed).

Service Programs

INPUT

is the name of an EXPAND input file that contains EXPAND control records. If you do not specify INPUT, the filename will default to the name of the text file that you are expanding (fn1). The filetype must be EXPAND. The system will search all accessed disks for this file.

Do not specify this option with the CSECT or SIZE options.

PRINT

prints on the printer all output that EXPAND produces. In addition, the system displays error messages, commands in error, and control records in error.

NOPRINT

does not print any output on the printer, and instead displays it at the terminal.

System Response

- After the system expands each CSECT that you specified, the system will issue a message indicating the following:
 - The number of bytes added to the control section.
 - The number of bytes added may be greater than the length that you specify for the expansion. This may occur if, after the specified expansion, the system must add padding bytes to align the next control section or common area.
 - The offset, relative to the start of the specified control section where the expansion began.
- If the system encounters an error during processing, it aborts the update and does not perform the expansions.

The Programmable Operator Facility

The Programmable Operator Facility is designed to increase the efficiency of system operation and to allow remote operation of systems in a distributed data processing environment. It does this by comparing incoming messages against a list of entries in the routing table (a CMS file); when a match occurs, the appropriate action is performed. The tasks that can be performed by the programmable operator facility include:

- Logging messages
- Suppressing message display and routing messages to a logical (real) operator
- Executing commands
- Responding with preprogrammed message responses.

Informational messages, such as those appearing on the CP system console as a result of the LOGON and LOGOFF commands, can be filtered out by the programmable operator facility and not clutter the real operator's display.

Messages with requests that cannot be handled by the programmable operator facility either because they require physical intervention, because the user is not authorized to issue the programmable operator facility command, or because the request does not match an entry in the routing table are routed to a real operator (the logical operator) at another virtual machine for consideration and/or appropriate action. These requests may include tasks such as mounting tapes and loading paper into the printer or may be messages to the operator from CMS users.

The logical operator may or may not be located on the same physical system as the programmable operator facility. When the logical operator and the programmable operator facility are not on the same VM system, the communications are handled through RSCS networking. Several different physical systems may be connected by an RSCS network and controlled by one logical operator from a single console. When the logical operator is an NCCF (Network Communication Control Facility) operator, the communications are handled primarily by the PROP/NCCF Message Exchange (PMX) portion of the programmable operator facility.

Use in a Single System

When the programmable operator facility is operational in a single-system environment, it can:

- Ease message traffic to the system operator, by:
 - Filtering (logging) non-essential, information-only messages
 - Routing messages (for example, I/O intervention requests) to someone else for specialized action.
- Increase productivity, by freeing the system operator from certain routine responses or tasks. Such responses (whether they consist of one or a series of commands, whether VM/SP or guest operating system) may be programmed to execute automatically upon receipt of a given message.

Thus, only essential, non-routine messages (that is, those requiring the skill and experience of a system operator to handle) are sent on to the operator for response or action.

Use in a Distributed VM Environment

The capabilities of the programmable operator, outlined above, also allow for the remote operation of systems in a distributed environment. When the programmable operator facility is operational in a distributed system, it can:

- Issue responses and perform tasks that do not require an on-site operator.
- Filter (log) non-essential, information-only messages.

Service Programs

- Route messages requiring on-site (that is, manual) intervention to someone, not necessarily an operator, at the distributed site for action.
- Route messages that require the skill and experience of a system operator to handle to the operator at the host system. The operator at the host site also has the capability to send commands to the programmable operator facility to control its operation, as well as commands to execute on the distributed system to control the system itself.

By running the programmable operator facility on VM/SP systems distributed at several different locations (network nodes), one operator at a host site can control a network of systems.

Figure 4-14 shows a simplified configuration of the logical operator controlling a distributed system through RSCS networking. Communications between the programmable operator facility running in the system operator virtual machine (OPERATOR) at the remote site and the logical operator virtual machine (LGLOPR) at the host site are handled through the RSCS network that connects the two VM systems.

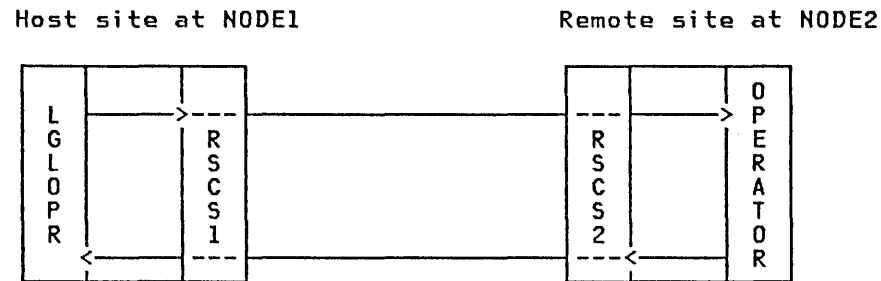


Figure 4-14. Example of a Programmable Operator Facility Configuration.

Use in a Mixed Environment

The programmable operator facility also provides for distributed data processing in an SNA environment with mixed VM, OS/VS, and VSE distributed systems and host systems, called a "mixed environment". The Programmable Operator/NCCF Message Exchange (PMX) provides an interface with NCCF so that an operator on an OS/VS or VSE system can operate a VM distributed system from an NCCF operator station. This means that an NCCF operator can be the logical operator for a programmable operator facility. Also, an NCCF operator (not necessarily the logical operator) can issue programmable operator commands, including the ability to run any VM command that the programmable operator virtual machine is authorized to run.

Notes:

1. The programmable operator facility does not let a user logged onto VM issue NCCF commands.

2. VM/SP Group Control System (GCS) is a requirement for this support. The Programmable Operator/NCCF Message Exchange (PMX) uses facilities unique to GCS, and cannot run on any other supervisor. For more information on GCS, see *VM/SP Group Control System Guide*, SC24-5249.

When the logical operator is an NCCF operator, the programmable operator does not use RSCS to route messages to the logical operator; instead, the programmable operator passes the messages to NCCF through the NCCF Message Queuing Service, so NCCF displays the messages at the appropriate NCCF operator console.

Figure 4-15 shows a single system environment, VM-only, where NCCF has been added. In this configuration, if you log on to an NCCF operator station you can operate the VM system. You can extend this idea to multiple unconnected systems which can be controlled by different NCCF operator stations centralized at the same location. This, however, requires one operator station for each of the systems being controlled.

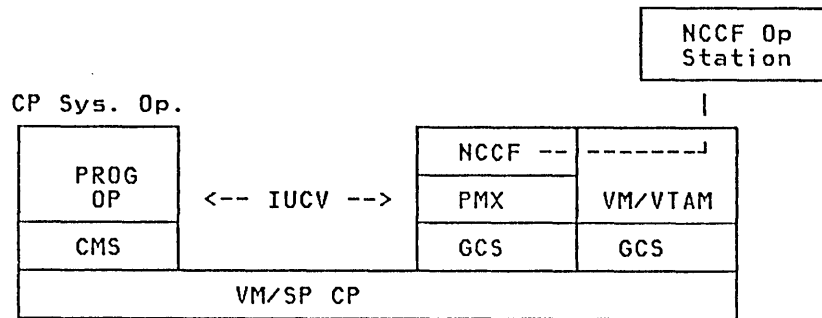


Figure 4-15. A single system with NCCF

Invoking the Programmable Operator Facility

Before using the Programmable Operator Facility, it must be installed as prescribed by the steps given in the *VM/SP System Programmer's Guide* under the section "Installing the Programmable Operator Facility." After the programmable operator facility has been installed and if it is not automatically invoked, the PROPST EXEC can be used to manually invoke the facility.

The PROPST EXEC drops any IBM-supplied programmable operator routines that are currently loaded as nucleus extensions and loads the programmable operator as a nucleus extension. It then invokes the programmable operator facility with the specified RTABLE. If you do not specify a routing table, the RTABLE name defaults to "PROP." You may specify a disconnect parameter to cause the programmable operator virtual machine to be disconnected before the programmable operator is invoked. The format of the PROPST EXEC is as follows:

Service Programs

PROPST	<code>[<i>rtable-name</i>] [DISConn]</code> <code><u>PROP</u></code>
---------------	---

Optionally, you can take the following steps each time you invoke the programmable operator facility:

1. Issue a FILEDEF command to assign a CMS filename to the PROPLIB LOADLIB file so CMS can read and load from it. This is done as follows:

```
FILEDEF PROPLIB DISK PROPLIB LOADLIB *
```

2. Load the programmable operator program as a CMS nucleus extension via the NUCXLOAD command. The command is issued as:

```
NUCXLOAD PROP DMSPOP PROPLIB
```

(The details of the NUCXLOAD command are given in *VM/SP CMS Command and Macro Reference*.)

These first two steps may be omitted for subsequent invocations as long as you do not:

- IPL CMS, or
 - Have a CMS abend from which the programmable operator was not able to automatically recover.
3. Following its loading as a CMS nucleus extension, invoke the programmable operator facility in the same way as any CMS command. The format of the invocation is:

```
PROP [routing-table-name]  
PROP
```

routing-table-name

is the filename of the routing table that is to be used by the programmable operator facility. The routing table provided with the programmable operator facility has a file name of PROP and a file type of RTABLE.

PROP

is the default file name if no other is specified by the installation.

When the programmable operator facility initialization is completed, the message PROP RUNNING - ENTER ' STOP ' TO TERMINATE is displayed on the console of the programmable operator facility virtual machine. The programmable operator facility then waits for either an incoming message or a programmable operator facility console command (either STOP or SET). At this point, the user may disconnect from the programmable operator facility virtual machine by

entering CP mode (pressing the PA1 key or equivalent) and typing DISCONN. The logical operator should now log on to the appropriate system or virtual machine if not already logged on.

If initialization cannot be successfully completed, an error message followed by the message PROP HAS TERMINATED is displayed on the console of the programmable operator facility virtual machine and programmable operator facility operation is terminated.

Invocation of Programmable Operator Facility Commands

The commands used with the Programmable Operator Facility can be executed by anyone who is authorized by the active routing table. If an unauthorized user issues a programmable operator facility command, the command is not executed but is routed to the logical operator. To send a command to the programmable operator facility you must send a message to the programmable operator facility virtual machine. The text of the message is the command to be issued. Unless otherwise noted, the response from the command is returned to the user who sent the command to the programmable operator facility virtual machine. Responses are sent one line at a time via the CP MSGNOH or MESSAGE commands.

The format of the message sent to the programmable operator facility virtual machine is the same as used with the CP MESSAGE command for local use and the CP SMSG command for distributed (network) use. In a mixed environment, an NCCF operator uses PROP, an NCCF command, to send messages to the programmable operator facility virtual machine.

The local format is:

Message MSG	<i>userid</i> propcmd [parameters]
------------------------	--------------------------------------

The distributed (network) format is:

SMsg	<i>netid Msg nodeid userid</i> propcmd [parameters]
-------------	---

netid

is the userid of the RSCS network machine at the user's node.

Msg

is the required RSCS message command. It can be entered as M or MSG.

nodeid

is the nodeid of the programmable operator facility virtual machine.

Service Programs

userid

is the userid of the programmable operator facility virtual machine.

propcmd

is the command to be executed by the programmable operator facility. See “Programmable Operator Facility Commands” on page 4-70 for information about the commands.

parameters

are the parameters associated with the command to be executed.

The format for an NCCF operator station is:

PROP	propcmd [parameters]
------	----------------------

propcmd

is the command that the programmable operator facility will execute

parameters

are the parameters associated with the command to be executed.

Notes:

1. To use this format, you must be authorized by NCCF.
2. If you (NCCF operator) are not attached to the local system, use the NCCF ROUTE command to specify the domain where you want to execute the command.

Issuing Commands in the Local Environment

If the Programmable Operator Facility is running in the operator’s virtual machine, a user on the same physical machine might send the messages:

MESSAGE OPERATOR QUERY RTABLE

MESSAGE OPERATOR WHAT TIME IS SHUT DOWN?

where “QUERY” is a programmable operator facility command. See “QUERY Command” on page 4-80 for information about the programmable operator facility QUERY command.

Figure 4-16 on page 4-63 shows how messages are transferred between a user, the programmable operator facility, and the logical operator when all are located on the same physical system.

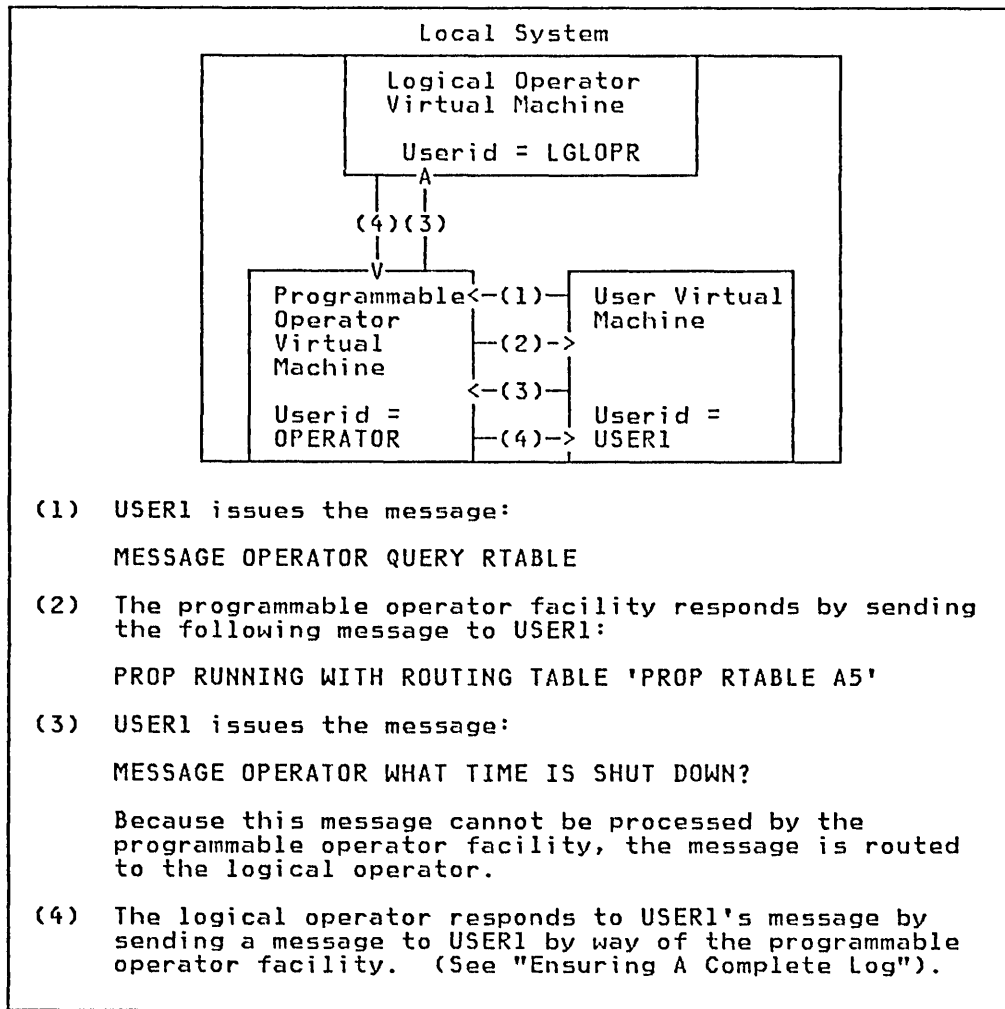


Figure 4-16. Example of Communication in the Local Environment

| Issuing Commands In The Distributed VM Environment

A user (USER21) on another VM system (assuming a remote system with the nodeid of "NODE2," a network machine with the id of "NET2," and that the programmable operator facility is running in the operator's virtual machine with a userid of "OPERATOR") might send the messages:

```
MESSAGE OPERATOR QUERY RTABLE
MESSAGE OPERATOR WHAT TIME IS SHUT DOWN?
```

where "QUERY" is a programmable operator facility command. See "QUERY Command" on page 4-81 for information about the programmable operator facility QUERY command.

Service Programs

Figure 4-17 shows how a user on one physical system exchanges messages with the programmable operator facility on a different physical system and how the programmable operator facility responds directly to a user on the same physical system or routes messages to the logical operator who is located on a different physical system.

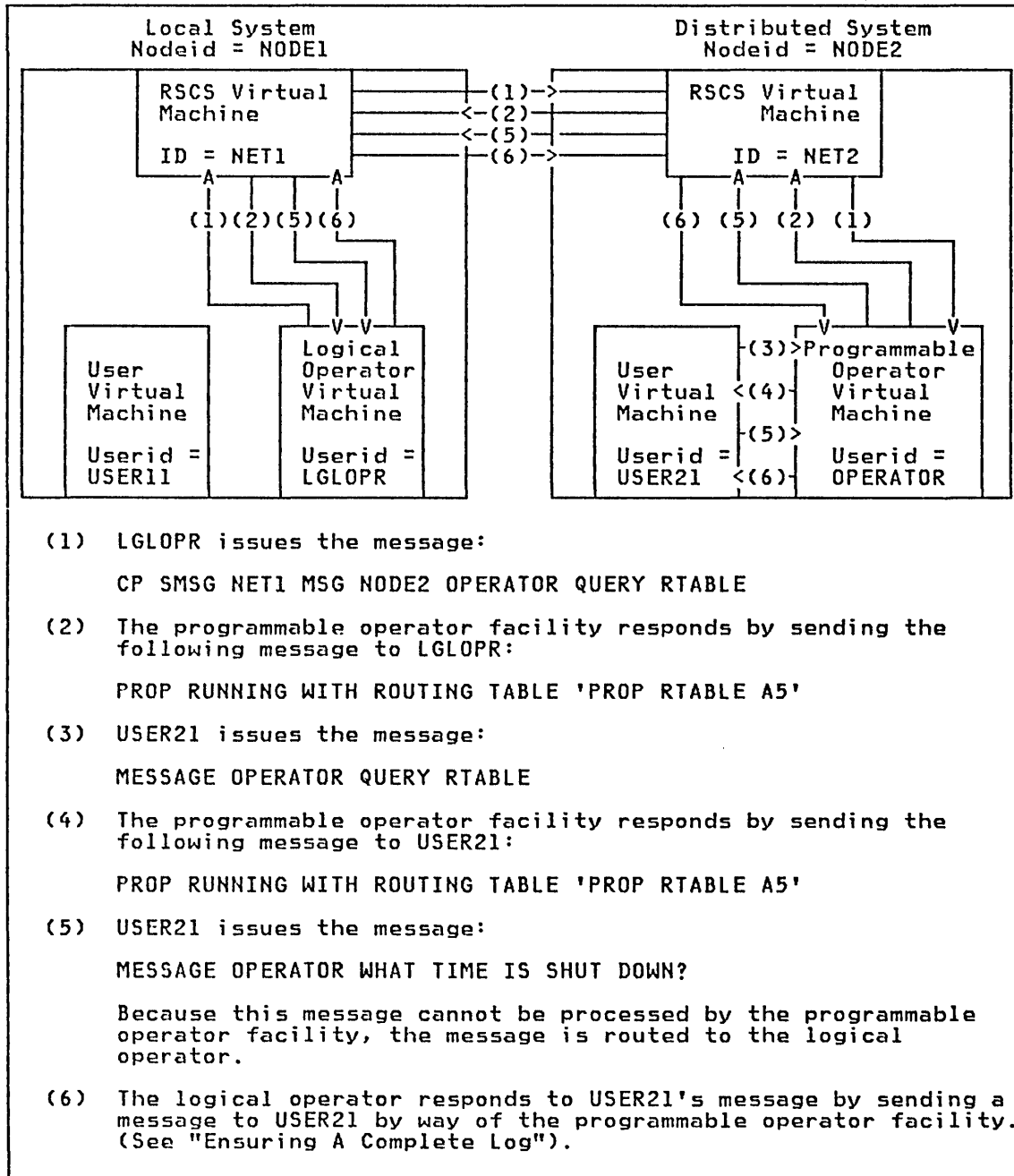


Figure 4-17. Example of Communication in the Distributed Environment

| Issuing Commands in a Mixed Environment

| In a mixed environment, message paths are more complex. A user on a VM system
| (USER3) might send the messages:

| MESSAGE OPERATOR QUERY RTABLE

| MESSAGE OPERATOR WHAT TIME IS SHUT DOWN?

| where "QUERY" is a programmable operator facility command. See "QUERY
| Command" on page 4-80 for information about the programmable operator
| QUERY command.

| The message sent containing the QUERY command simply follows the paths of a
| message sent by a user to the programmable operator facility in the same system.
| The message is processed by the programmable operator facility and a response is
| returned to the user (USER3).

| Figure 4-18 on page 4-66 shows how messages are transferred between a user,
| the programmable operator facility, and an NCCF logical operator. It also shows
| how messages and commands are transferred from an NCCF operator to the pro-
| grammable operator facility and its VM system.

Service Programs

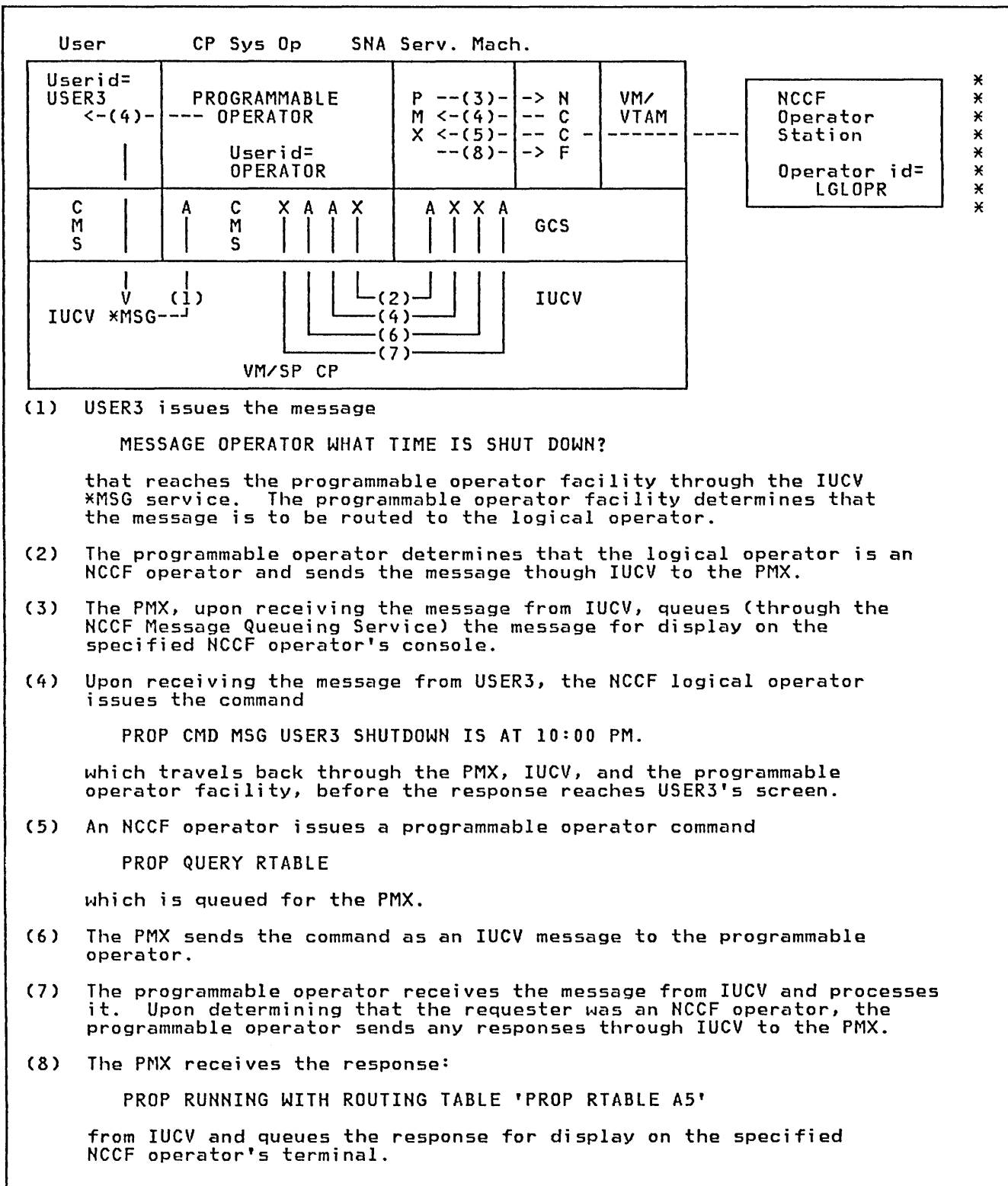


Figure 4-18. Example of Communication in the Mixed Environment

Helpful Hints

The typing of large text strings, such as the message command string preceding the message text, can be minimized by assigning the string, up to the userid, to a PF key. An EXEC can also be used that prompts for, or accepts as parameters, that part of the message command that the user must type in.

For example:

```
SET PF01 IMMED SMSG NET1 MSG NODE2 OPERATOR QUERY RTABLE
```

will allow the logical operator to press PF01 to find out the name of the active routing table when the logical operator and programmable operator facility are located on different physical systems.

The CMS TELL and NAMES exec procedures may also be used by the logical operator to send a message to a user. If the logical operator has a NAMES file entry assigning the nickname PROP1 to OP at node NODE1, the logical operator could send a message in the following manner:

```
TELL PROP1 QUERY RTABLE
```

```
TELL PROP1 CMD WNG ALL ...WILL RE-IPL IN 5 MIN - PLS LOGOFF.
```

These exec procedures are described in *VM/SP CMS Command and Macro Reference* and *VM/SP CMS User's Guide*.

The Log File

If LOGGING is not OFF, the programmable operator facility puts every message that it received into a CMS file called the **log file**. If LOGGING is set to ALL, the programmable operator facility puts all generated error messages and command responses in the log file.

If LOGGING is ON, the programmable operator facility does not log responses from CP, CMS, and programmable operator facility commands, but does log messages. Each message is identified by the date and time received. The userid and nodeid are in the message only if the text was sent by a CP MSG, SMSG, WNG, or sent using SCIF (Single Console Image Facility). For a message sent by CP, the userid and nodeid are blank. A message from a remote RSCS network virtual machine has a nodeid, but no userid. A message sent by an NCCF operator console has '*NCCF' as the nodeid.

Log entries generated and logged by the programmable operator have a userid of PROP. The log file has the following format:

Service Programs

col 1	col 10	col 19	col 28	col 39
↓	↓	↓	↓	↓
yy/mm/dd	hh:mm:ss	[userid	nodeid]:	text

The log file has variable length records. The maximum record length that can be in the log file is 132 characters (38 prefix characters plus 94 text characters). Messages with a text length greater than 94 characters will be continued on the next log record. The continuation record will have the prefix characters (columns 1 through 36), but no colon (:) in column 37.

A separate log file is started for each day. The name of the file is:

LGyymmdd nodeid A5

yy
is the current year

mm
is the current month

dd
is the current day.

nodeid
is the current RSCS nodeid of the system where which the programmable operator facility is running.

Any non-NCCF user listed in the active routing table can use the GET LOG command to get the log file from the programmable operator facility as a reader spool file.

Any user authorized in the active routing table to use the programmable operator facility CMD command can purge an old log file. Use the CMS ERASE command to purge the file.

Ensuring A Complete Log

When the programmable operator facility routes a message to the logical operator, the message has the userid of the sender. To respond to the message, you can send a message directly to the user without going through the programmable operator facility. If you do send a message this way, the message will not be logged by the programmable operator facility.

To ensure that the programmable operator facility logs the messages, use the programmable operator facility CMD command to send the message to the user. See "CMD Command" on page 4-71 for information about the programmable operator facility CMD command. To be sure that user messages always have the same id (the programmable operator facility id), send messages from the logical operator through the programmable operator facility.

To route a message through the programmable operator facility where the operator and user are not on the same physical VM system, use the SMSG command:

SMSG *net1 msg node1 operator* CMD MSG *user1* - RESPONDING TO YOUR REQUEST

net1

is the userid of the network machine at the user's node.

msg

is the RSCS message command.

node1

is the nodeid of the programmable operator facility virtual machine.

operator

is the userid of the programmable operator facility virtual machine.

CMD

is the programmable operator facility CMD command.

MSG

is the VM command that the programmable operator facility will execute.

user1

is the userid of the user who will receive the message.

RESPONDING TO YOUR REQUEST

is the message text sent to the user.

To route a message through the programmable operator facility where the operator and user are on the same physical system, use the MSG command:

MSG *operator* CMD MSG *user1* - RESPONDING TO YOUR REQUEST

operator

is the userid of the programmable operator facility virtual machine.

CMD

is the programmable operator facility CMD command.

MSG

is the VM command that the programmable operator facility will execute.

user1

is the userid of the user who will receive the message.

RESPONDING TO YOUR REQUEST

is the message text sent to the user.

Service Programs

To route a message through the programmable operator facility from the logical operator who is an NCCF operator to the user on a VM system, use PROP, an NCCF command as follows:

PROP CMD MSG *USER1* - RESPONDING TO YOUR REQUEST

PROP

is the NCCF command that sends the message from an NCCF operator to the programmable operator facility.

CMD

is the programmable operator CMD command.

MSG

is the VM command that the programmable operator facility will execute.

USER1

is the userid of the user who will receive the message.

RESPONDING TO YOUR REQUEST

is the message text sent to the user.

Note: Refer to “Helpful Hints” on page 4-67 for ways to reduce typing of long text strings.

Programmable Operator Facility Commands

In the following command descriptions, an authorized user is any user for which there is a corresponding routing table entry for the command he wishes to issue. In other words, authorization for the programmable operator facility is controlled entirely by the contents of the routing table. All programmable operator facility commands are sent by the user as messages. (See the programmable operator section of the *VM/SP System Programmer's Guide* for more information on the routing table).

CMD Command

Use the CMD command to execute CP or CMS commands in the programmable operator facility virtual machine. The command is accepted or rejected by CP based on the CP user class defined for the programmable operator facility virtual machine. Also, the programmable operator facility may reject or accept the command based on the authorization granted in the active routing table. The user class assigned to the programmable operator facility virtual machine is determined by the installation. If accepted, the response from the command is returned to the issuer of the command.

The format of the CMD command is:

CMD	<i>vmcmd</i>
-----	--------------

vmcmd

is the VM command sent to the programmable operator facility virtual machine for execution according to the CMS IMPCP and IMPEX settings.

Usage Notes

Notes:

1. Any commands that alter or overlay CMS storage (CP DEFINE STORAGE, CP IPL CMS, CP SHUTDOWN, etc.) will have an adverse effect on the operation of the programmable operator facility and should not be issued under the programmable operator facility.
2. Reissuing the PROP command once the programmable operator facility is running will cause it to stop operating correctly. The user must then reIPL CMS and restart the programmable operator facility using the procedure described under “Invoking the Programmable Operator Facility” on page 4-59.
3. Issuing commands that cause a VM READ or a CP READ (such as the DDR command) will stop the programmable operator facility. It must then be restarted using the procedure described under “Invoking the Programmable Operator Facility” on page 4-59, or the read must be answered from the console of the programmable operator facility virtual machine. Commands of this type should not be sent to the programmable operator facility.
4. Line editing characters (CHARDEL, LINEDEL, LINEND, and ESCAPE), although interpreted by your terminal when you have SET LINEDIT ON in effect, are not interpreted as line editing characters when sent to the programmable operator facility. In other words, they are interpreted as the characters they are (that is, @, ¢, #, :). For example, the string:

```
M OP CMD ACCESS 191 A"#RECEIVE
```


Service Programs

will cause an “INVALID MODE ‘ A#RECEIV ’” message to be returned to the issuer. The default line editing characters may be defined by the installation. You may also define your own line editing characters by using the CP TERMINAL command.

5. The programmable operator facility does not recognize the CMS immediate commands (HB, HI, HO, HT, HX, RO, RT, SO, TE, and TS). If you issue any of these commands, the programmable operator facility issues “UNKNOWN CP/CMS COMMAND.”

Examples

The command issued at a virtual machine console:

M OP CMD INDICATE

M OP CMD QUERY FILES

M OP CMD QUERY PRINTER ALL

M OP CMD ERASE LG801015 NODE1 A5

M OP CMD QUERY SEARCH

M OP CMD LISTFILE

The command issued at an NCCF operator terminal:

PROP CMD INDICATE

PROP CMD QUERY FILES

PROP CMD QUERY PRINTER ALL

PROP CMD ERASE LG801015 NODE1 A5

PROP CMD QUERY SEARCH

PROP CMD LISTFILE

Responses

The response returned by CP or CMS is sent to the issuer of the command. After the response from CP or CMS, the programmable operator facility responds with “COMMAND COMPLETE”.

FEEDBACK Command

Use the FEEDBACK command to place comments about the operation of the system and/or the programmable operator facility in the feedback file. These comments are available for review by personnel responsible for maintenance of the programmable operator facility. The comment (preceded by the date and time it was received, and the sender's userid and nodeid) is placed in a file named "FEEDBACK nodeid A5,".

The format of the FEEDBACK command is:

FEEDBACK FB	<i>text...</i>
------------------------	----------------

text...

is the user's comments placed in the feedback file (FEEDBACK nodeid A5).

Usage Notes

Notes:

1. The feedback file can be retrieved by authorized non-NCCF users, using the programmable operator facility GET command.
2. The length of the message is limited by the maximum length of the command used to send the message to the programmable operator facility. If the user desires to send a longer message, the command must be used multiple times.

Example

The command issued at a virtual machine console:

```
M OP FEEDBACK SYSTEM RESPONSE WAS SLOW DURING THE MORNING  
SHIFT
```

The command issued at an NCCF operator terminal:

```
PROP FEEDBACK SYSTEM RESPONSE WAS SLOW DURING THE MORNING  
SHIFT
```

Response

```
COMMAND COMPLETE
```

Service Programs

GET Command

Use the GET command to retrieve one of the programmable operator facility files; the feedback file (FB or FEEDBACK) or the log file (LOG). The file is sent to the requesting user if he is authorized in the active routing table to receive it. If LOG is specified, the user will receive the log file for either the current day or the specified day.

The format of the GET command is:

GET	{ FEEDBACK FB LOG [yymmdd]}
-----	--------------------------------------

FEEDBACK

FB

indicates that the feedback file is to be retrieved.

LOG [yymmdd]

indicates that the log file for date "yymmdd" is to be retrieved. If no date is given, the log file for the current day is retrieved.

Usage Notes

1. An NCCF operator cannot use the GET command to get the LOG and FEEDBACK files. Use CMS commands (and the programmable operator CMD command) to type the file(s) or portions of the files, or send the file(s) to some userid where you can process them.
2. The file appears in the requesting user's virtual reader in DISK DUMP format. The user must execute a DISK LOAD or RECEIVE command to read the file.

Examples

```
M OP GET FB
```

```
M OP GET LOG 801030
```

Response

```
COMMAND COMPLETE
```

LGLOPR Command

Use the LGLOPR command to assign or release yourself as the logical operator for the programmable operator facility under which the command is executed. This programmable operator command can be used by a VM user or an NCCF operator.

Users are authorized in the active programmable operator routing table.

LGLOPR	{ ASN RLS RPL }
--------	-----------------------------

where:

ASN

assigns the issuer of the command as the logical operator, if a logical operator is not currently assigned (i.e. the current LGLOPR is the default). If a logical operator is already assigned to the programmable operator facility, an error response is given.

RLS

releases the issuer from being the logical operator, if he currently is the logical operator and assigns the default LGLOPR. If the issuer is not the logical operator, no operation is performed and the system gives the following response:

```
DMSPOR763E NOT CURRENTLY ASSIGNED AS LGLOPR, CANNOT BE  
RELEASED
```

RPL

replaces the current logical operator with the issuer of the command. The programmable operator determines if a logical operator is currently assigned. If there is, an implicit release is done. Then the issuer of the command becomes the logical operator.

Usage Notes

Notes:

1. The system keeps the logical operator that is specified on the LGLOPR statement in the routing table as a default. You cannot release this logical operator. The LGLOPR ASN and LGLOPR RPL commands override the default.
2. To ensure that messages are not lost when changing logical operators, the new logical operator should issue a "LGLOPR RPL" command, rather than the current logical operator issuing a "LGLOPR RLS" command and the new logical operator issuing a "LGLOPR ASN" command.

Service Programs

3. If the default logical operator is the current logical operator and he issues the LGLOPR ASN command, CP will send a message stating that he already assigned as the logical operator
4. The HOSTCHK function is suspended when an NCCF operator or a local VM user is assigned as the logical operator. It is resumed when a remote VM user is assigned as the logical operator.

Examples

The command issued at a virtual machine console:

```
MSG OP LGLOPR ASN
```

The command issued at an NCCF operator terminal:

```
PROP LGLOPR ASN
```

Responses

Both the new and old logical operators receive the message:

```
{NCCF|VM} USER 'userid' ['nodeid'] IS NOW LGLOPR  
FOR PROP ON NODE 'nodeid'
```

to notify them of the change of logical operators. The requester also receives the response:

```
COMMAND COMPLETE
```

LOADTBL Command

Use the LOADTBL command to dynamically load a new routing table. You, as a VM user or an NCCF operator, can specify whether or not the currently assigned logical operator should be replaced by the logical operator specified in the new routing table.

```
LOADTBL [filename] [(RPL [ ])]
```

where:

filename

is the name of the routing table to be loaded. If filename is not given, the default routing table (filename = PROP) is used.

RPL

replaces the currently assigned logical operator with the logical operator specified in the new routing table. The logical operator is replaced only after the new routing table has been successfully loaded.

If RPL is not specified, the logical operator in the new routing table simply becomes the new default logical operator, and any explicitly assigned logical operator (i.e. a logical operator assigned by the LGLOPR command) remains the logical operator.

Usage Notes

Notes:

1. If any action routines named in the specified routing table can not be located or loaded, an error message is issued, and the programmable operator facility drops any action routine modules associated with the specified routing table. It then tries to reload the action routine modules associated with the routing table that was active before the LOADTBL command was issued. If it cannot load these modules, the programmable operator facility terminates operation.

Note: If any of the action routines associated with the previous routing table were modified (that is, replaced in LOADLIB) between the time the programmable operator facility dropped the specified routing table modules and reloaded the previously active routing table modules, the modified version of the action routines are used when the previous routing table is reloaded.

2. Only the filename is used to identify the routing table file to the LOADTBL command.
3. Because DMSPOL is the action routine module executing the LOADTBL command, it is not dropped and reloaded as are the other action routine modules listed in the routing table. If you want to replace this module, you must stop the programmable operator facility (using the STOP command), make the desired modifications or replacement, and reinvoke the program-

Service Programs

mable operator facility as described under “Invoking the Programmable Operator Facility” on page 4-59.

If the programmable operator virtual machine is set up so that the programmable operator is started automatically when CMS is IPLed in that virtual machine, it is sufficient to do the replacement and then IPL CMS again.

4. With the loading done by DMSPOL, it is possible for the other routines in DMSPOR to be replaced when a LOADTBL occurs. This permits changes to action routines other than DMSPOL to be made dynamically, without stopping the programmable operator. It is also possible to specify the name of a table to be loaded as a parameter to the action routine. The logical operator will be notified of the loading.
5. If the current logical operator is the default logical operator (not explicitly assigned), then the current logical operator will be replaced even if the RPL option is not specified.

6. When you issue LOADTBL and replace the logical operator, both the old and new logical operators receive the following message:

```
[NCCF|VM] USER 'userid [nodeid]' IS NOW LGLOPR FOR PROP ON NODE
nodeid'
```

Both operators receive this message, even if you have not specified the RPL option.

Example

The command issued at a virtual machine console:

```
MSG OP LOADTBL ROUTE3 (RPL
```

The command issued at an NCCF operator terminal:

```
PROP LOADTBL ROUTE3 (RPL
```

Responses

```
NEW RTABLE NOT LOADED
```

```
PROP TERMINATED
```

```
PROP RUNNING WITH ROUTING TABLE 'fn ft fm'
```

LOG Command

Use the LOG command to write a message to the log file. Use of the LOG command allows messages to be placed in the log file with no action taken by the programmable operator facility.

The format of the LOG command is:

LOG	<i>text...</i>
------------	----------------

text...

is the message text to be placed in the log file.

Usage Notes

Notes:

1. All messages are logged whether or not the LOG command is explicitly used.
2. Authorized non-NCCF users can retrieve the log file using the GET LOG command. The log file has a fileid of "LGyymmdd nodeid A5" where "yy" is the year, "mm" is the month, "dd" is the day, and "nodeid" is the nodeid of the system on which the programmable operator facility is running.

The userid and nodeid of the sender is recorded along with the message.

3. The length of the message is limited by the maximum length of the command used to send the message to the programmable operator facility. If you want to send a longer message, you must use the command multiple times.

Example

| The command issued at a virtual machine console:

| M OP LOG THIS MESSAGE IS TO BE LOGGED.

| The command issued at an NCCF operator terminal:

| PROP LOG THIS MESSAGE IS TO BE LOGGED.

Response

COMMAND COMPLETE

Service Programs

QUERY Command

Use QUERY RTABLE to find the name of the active routing table.

Use QUERY PROPCHK and QUERY HOSTCHK to query the status of the programmable operator node-checking.

Use QUERY LOGGING to query the status of the logging messages.

Use QUERY LGLOPR to find the name of the currently assigned logical operator.

The format of the QUERY command is:

QUERY	{ RTABLE PROPCHK [nodeid] HOSTCHK LOGGING LGLOPR }
-------	--

RTABLE

displays the name of the active routing table.

PROPCHK

displays a message with node-checking status. The message will state whether PROPCHK is set ON or OFF. If a nodeid is supplied, only the node specified is checked.

HOSTCHK

displays a message with node-checking status. The message will state whether HOSTCHK is set ON or OFF.

LOGGING

displays a message with logging status. The message will state whether messages and responses are being logged (LOGGING ALL), incoming messages and special programmable operator messages are being logged (LOGGING ON), or there is no log (LOGGING OFF).

LGLOPR

for a VM logical operator, displays a message with the userid and nodeid of the operator. For an NCCF logical operator, LGLOPR displays only the userid (i.e. operator-id) of the operator.

Example

The command issued at a virtual machine console:

```
M OP QUERY RTABLE
M OP QUERY PROPCHK NODE1
M OP QUERY HOSTCHK
M OP QUERY LOGGING
M OP QUERY LGLOPR
```

The command issued at an NCCF operator terminal:

```
PROP QUERY RTABLE
PROP QUERY PROPCHK VMSYS1
PROP QUERY HOSTCHK
PROP QUERY LOGGING
PROP QUERY LGLOPR
```

Responses

```
PROP RUNNING WITH ROUTING TABLE 'fn ft fm'
```

is received if the programmable operator facility is running.

fn
is the filename of the active routing table.

ft
is the filetype of the active routing table.

fm
is the filemode of the active routing table.

```
{PROPCHK|HOSTCHK} IS ON
```

is received if node-checking is in effect.

```
{PROPCHK|HOSTCHK} IS OFF
```

is received if node-checking was specified in the RTABLE but is currently off.

```
PROPCHK IS {ON|OFF} FOR NODEID nodeid
```

is received if a specified nodeid is being queried.

Service Programs

HOST-CHECKING IS SUSPENDED; LGLOPR NOT ON A CHECKABLE NODE
is received if the logical operator is an NCCF operator or a local VM user.

{PROPCHK|HOSTCHK} NOT SPECIFIED IN RTABLE
is received if node-checking was not specified in the current RTABLE.

PROPCHK NOT SPECIFIED IN RTABLE FOR NODEID nodeid
is received if a nodeid was specified on the QUERY command and node-checking was not specified in the current RTABLE for that node.

LOGGING ALL
is received if incoming messages and all programmable operator responses are being logged.

LOGGING ON
is received if incoming messages and special programmable operator responses are being logged.

LOGGING OFF
is received if no logging is being done.

[NCCF|VM] USER 'userid [nodeid]' IS NOW LGLOPR FOR PROP ON NODE
'nodeid'
is received when the logical operator is being queried.

SET Command

Use SET DEBUG to enter or exit the programmable operator facility DEBUG mode. DEBUG mode is used to do problem determination on the programmable operator facility.

Use SET PROPCHK to set the periodic checking of the programmable operator on the distributed systems ON or OFF. The distributed systems are identified by the PROPCHK statements in the routing table of the host programmable operator. The programmable operator facility with the PROPCHK statement (e.g. the host system) does the checking.

Use SET HOSTCHK to set the periodic checking of the link to the host system ON or OFF. HOSTCHK must be specified in the routing table of the programmable operator at the distributed system. The programmable operator facility with the HOSTCHK statement (e.g. the distributed system) does the checking.

Use SET LOGGING to control messages going to the programmable operator log file. SET LOGGING allows the message sender control the logging level: no logging, logging incoming messages and special programmable operator messages, or incoming messages plus response messages.

The format of the SET command is:

SET	<table style="border: none; width: 100%;"> <tr> <td style="padding-right: 10px;">DEBUG</td> <td style="padding-right: 10px;">{ ON }</td> <td></td> </tr> <tr> <td></td> <td>{ OFF }</td> <td></td> </tr> <tr> <td style="padding-right: 10px;">PROPCHK</td> <td style="padding-right: 10px;">{ ON }</td> <td>[nodeid]</td> </tr> <tr> <td></td> <td>{ OFF }</td> <td>[nodeid]</td> </tr> <tr> <td style="padding-right: 10px;">HOSTCHK</td> <td style="padding-right: 10px;">{ ON }</td> <td></td> </tr> <tr> <td></td> <td>{ OFF }</td> <td></td> </tr> <tr> <td style="padding-right: 10px;">LOGGING</td> <td style="padding-right: 10px;">{ ON }</td> <td></td> </tr> <tr> <td></td> <td>{ OFF }</td> <td></td> </tr> <tr> <td></td> <td>{ ALL }</td> <td></td> </tr> </table>	DEBUG	{ ON }			{ OFF }		PROPCHK	{ ON }	[nodeid]		{ OFF }	[nodeid]	HOSTCHK	{ ON }			{ OFF }		LOGGING	{ ON }			{ OFF }			{ ALL }	
DEBUG	{ ON }																											
	{ OFF }																											
PROPCHK	{ ON }	[nodeid]																										
	{ OFF }	[nodeid]																										
HOSTCHK	{ ON }																											
	{ OFF }																											
LOGGING	{ ON }																											
	{ OFF }																											
	{ ALL }																											

DEBUG { ON }
{ OFF }

SET DEBUG ON stops the programmable operator facility from intercepting responses to CP commands. SET DEBUG OFF allows the programmable operator facility to return to its normal function of intercepting messages and responses from CP. SET DEBUG OFF is the initial setting.

PROPCHK { ON } nodeid
{ OFF } nodeid

SET PROPCHK OFF halts checking of the programmable operators on the distributed systems until this command is reissued to set the checking back

Service Programs

on, or until the programmable operator is stopped and restarted, or the PROP LOADTBL command is issued. SET PROPCHK ON restarts the checking. If nodeid is specified, PROPCHK applies to the specified node only. The initial setting is determined by the existence of the PROPCHK statement(s) in the RTABLE (ON if they exist, OFF if they do not).

An error message is received if the SET PROPCHK command is given and the routing table does not contain a PROPCHK statement or if the nodeid specified in the SET PROPCHK command is not found in a PROPCHK statement in the routing table.

HOSTCHK { ON }
 { OFF }

SET HOSTCHK OFF halts checking of the host system by the distributed system. SET HOSTCHK ON restarts the checking. The initial setting is determined by the existence of a HOSTCHK statement in the RTABLE (ON if it exists, OFF if it does not.)

LOGGING { ON }
 { OFF }
 { ALL }

SET LOGGING OFF causes the programmable operator facility to stop writing any messages to the log file. SET LOGGING ON or ALL causes logging to be resumed. SET LOGGING ALL causes logging of all programmable operator command responses, such as virtual machine console I/O generated by action routines. The LOGGING statement in the configuration portion of the routing table determines the initial setting. If no LOGGING statement appears in the routing table, the default is ON.

Usage Notes

Notes:

1. The SET DEBUG command is only valid from the console of the programmable operator facility virtual machine.
2. The SET DEBUG ON command permits an authorized user to stop the programmable operator facility from intercepting messages associated with CP commands entered from the console of the programmable operator facility virtual machine. The programmable operator facility responds to CP messages (MSG), warnings (WNG), and special messages (SMSG), and messages sent using the Single Console Image Facility (SCIF), but not to responses from CP.
3. When SET DEBUG OFF is in effect, all responses to CP commands are intercepted by the programmable operator facility. SET DEBUG OFF is in effect when the programmable operator facility is initialized.
4. If the logical operator is on a non-checkable node (for example an NCCF operator) when he turns HOSTCHK ON, the issuer of the SET command receives an error message.

Example

| The command issued at the programmable operator virtual machine console:

| SET DEBUG ON

| SET DEBUG OFF

| The command issued at a virtual machine console:

| M OP SET HOSTCHK OFF

| M OP SET PROPCHK ON SYS2

| M OP SET LOGGING OFF

| The command issued at an NCCF operator terminal:

| PROP SET HOSTCHK OFF

| PROP SET PROPCHK ON SYS2

| PROP SET LOGGING OFF

Responses

For commands sent by message:

COMMAND COMPLETE

For commands issued at the programmable operator console:

{PROPCHK|HOSTCHK} HAS BEEN STARTED

{PROPCHK|HOSTCHK} HAS BEEN STOPPED

LOGGING HAS BEEN STARTED

LOGGING HAS BEEN STOPPED

PROP RUNNING IN DEBUG MODE

PROP HAS EXITED DEBUG MODE

Service Programs

STOP Command

Use the STOP command to stop the operation of the programmable operator facility. When the STOP command is issued, the programmable operator facility processes all outstanding messages, closes files, stops operation, and returns control to CMS. The STOP command is logged in the log file for the current day.

The format of the STOP command is:

STOP	
------	--

Usage Note

The STOP command is also valid from the console of the programmable operator facility virtual machine if an operator is logged on to the programmable operator facility virtual machine.

Response

PROP HAS TERMINATED

Stopping the Programmable Operator Facility

The STOP command stops the operation of the programmable operator facility. This command can be issued by a user logged on to the programmable operator facility virtual machine by typing STOP and pressing the ENTER key (or its equivalent). The programmable operator facility will then complete processing of all pending messages before stopping and returning control to CMS.

An alternative way to stop the programmable operator facility is for an authorized user to pass the STOP command to the programmable operator facility virtual machine from another virtual machine. The facility will stop processing as described above and return control to CMS. If the programmable operator facility virtual machine was running disconnected, it is left disconnected and the programmable operator facility is not active. To restart the programmable operator facility, someone must log on to the programmable operator facility virtual machine and invoke the facility as described in "Invoking the Programmable Operator Facility" on page 4-59.

Effect on an NCCF operator: If you use STOP to stop the programmable operator facility, the system sends a message to you (NCCF operator) and the PMX enters a dormant state. Restart the programmable operator facility to continue operating in the mixed environment. If the programmable operator stops because of an abend and is able to recover, the programmable operator facility tries to re-establish the IUCV connection between the programmable operator and the PMX. In either case (STOP or ABEND), the programmable operator informs any ASN'd NCCF operator of its status (stopped or abended) by the PMX.

Stopping the PMX

If PMX stops or the NCCF operator session that is controlling the programmable operator stops, the NCCF logical operator (if any) is implicitly released. The programmable operator continues with the default logical operator specified on the LGLOPR statement.

The PMX may stop because:

- VM/SP GCS stops
- NCCF stops
- ABEND stops the PMX.

When the PMX stops, NCCF does not necessarily stop (the PMX attempts to wait until NCCF has stopped). However, you must stop NCCF before it restarts PMX.

Problem Determination - DEBUG MODE

Debug mode is used to do problem determination on the programmable operator facility itself. It allows responses to commands issued from the programmable operator facility virtual machine console to be sent back to the console without being intercepted by the programmable operator facility. This permits any CP command (the CP TRACE and ADSTOP commands, for example) to be issued without having its response trapped by the programmable operator facility.

SET DEBUG ON may be used after the programmable operator facility responds with the message:

```
PROP RUNNING - ENTER ' STOP ' TO TERMINATE
```

indicating that the programmable operator facility is running and operational. The programmable operator facility then responds with the message:

```
PROP RUNNING IN DEBUG MODE
```

When in debug mode, the programmable operator facility waits to receive messages from another virtual machine, or for the system programmer to enter input from the console. Because only two commands, STOP and SET, are acceptable from the console of the programmable operator facility virtual machine, the system programmer must enter the CP environment (via the PA1 key) to issue any CP com-

Service Programs

mands or the commands are intercepted and rejected as invalid programmable operator facility commands.

Pressing the PA1 key a second time or issuing the BEGIN command returns control to the programmable operator facility. From this environment, issuing SET DEBUG OFF causes the programmable operator facility to return to its normal function of trapping messages.

Stand-Alone Dump Facility

Overview

With the stand-alone dump facility, you can dump up to 16 megabytes of real storage when VM/SP cannot create a CP Abend dump. This facility dumps all resident pages, CP and non-CP. The stand-alone dump facility cannot dump virtual machine storage and non-resident pages from the paging device.

To use the stand-alone dump program to dump the real storage, you must have access to IPL the real machine. You can IPL the stand-alone dump program from tape or disk and direct the output to tape or printer. When using tape as the output device, reserve the complete tape for the stand-alone dump facility. Basic error recovery is available for DASD, tape, and printer devices used as IPL or output devices.

Typically, an installation can have several stand-alone dump programs generated and ready to run. It would be useful to have the following configurations available for the stand-alone dump facility:

- IPL from tape with output directed to printers
- IPL from tape with output directed to tapes
- IPL from DASD with output directed to printers
- IPL from DASD with output directed to tapes

These configurations let you take a stand-alone dump with any of the supported possible environments.

The stand-alone dump program communicates with the user with PSW wait codes. Refer to *VM/SP System Messages and Codes* in "Stand-Alone Dump Facility Wait State Codes." Once the CPU has gone into a wait state, the user can display the PSW, using conventional means, to find if the dump was successful.

The starting and ending addresses of the CP Trace Table are stored in the PSA at locations X'7B0' and X'7B4', respectively, in addition to the PSA locations X'0C' and X'10'.

Devices that You Can Use to IPL Stand-Alone Dump

The following are the devices you can use to IPL the stand-alone dump program:

DASD	Tape
3330	2401
3333	2415
3340	2420
3344	3410
3350	3420
3375	3430
3380	

Notes:

1. If a disk is selected as the IPL device:
 - a. It cannot be the resident system device
 - b. It must be CP formatted
 - c. Cylinder 0 must be allocated as permanent space
 - d. Cylinder 0 will be used.
2. The stand-alone dump IPL tape can be the same as the tape you direct the dump output to.
3. Do not try to IPL from a device that is not in the above list.

Devices to which You Can Send Dump Output

The following are the devices to which the dump output can be directed:

Tape	Printer
2401	1403
2415	1443
2420	3211
3410	3203 (Models 4 & 5)
3420	3289E (Model 4)
3430	3262 (Models 1, 5, & 11)
	4245
	4248

Notes:

1. You can specify a maximum of eight real addresses for the dump output device.
2. Do not mix printers and tapes in the same list. If you use a printer as the output device, the FCB should match the forms loaded in the printer. If the FCB does not match the form, data may be lost when the printer runs out of paper.

Service Programs

3. When you configure the stand-alone dump facility, you can use any printer type or tape type from the list of supported devices.

For example, the SADUMP exec prompts you for the output device type with the following:

```
PLEASE ENTER ONE OF THE FOLLOWING OUTPUT DEVICE TYPES:  
PRINTER: (1403, 1443, 3203, 3211, 3262, 3289, 4245, 4248)  
TAPE: (2401, 2415, 2420 3410, 3420, OR 3430)
```

If you specify 3420, the system expects the output to be directed to a tape device.

The system will then request the output device address with the following:

```
PLEASE ENTER REAL OUTPUT DEVICE ADDRESS OR LIST ADDRESSES  
(MAXIMUM OF 8) FOR TAPE: ENTRIES IN A LIST MUST BE SEPARATED  
BY A MINIMUM OF ONE BLANK.
```

You must then respond with the address or list of addresses of the tape device(s) which can receive the output. Be sure the output addresses match the device type (tape in this example); otherwise, results are unpredictable. Keep in mind that the generated stand-alone dump program does not check the address of the device for validity.

4. The stand-alone dump must have channel 1 defined in the FCB or carriage control tape, if the output device is a printer.
5. Do not send the stand-alone dump output to a device that is not included in the above list.

When you specify tape as the output device, the stand-alone dump program selects, as the dump output device, the first available device in the list, excluding the IPL device (if it is in the list). If you want the stand-alone dump output to go to the IPL tape, make all other devices that are in the list not ready. If no other device within the output address list is available and the IPL tape address is in the list, the IPL device will receive the dump.

If you select a tape for the dump output device, other than the IPL tape, the stand-alone dump facility:

1. Rewinds the tape to ensure that the dump is at the beginning of the tape
2. Sets the density to the highest value for the tape device.

If the tape device selected is the one on which the stand-alone dump facility resides, the facility will write the dump at the same density as the stand-alone dump program was written.

When using tape, reserve the complete tape for the stand-alone dump program; do not put the stand-alone dump program on a tape with the other stand-alone utilities. If you do not want to use the IPL tape as the dump output tape, you may want to put the stand-alone dump program on a mini-reel to make better use of tape resources.

If you select tape to be the output device type, use a single-volume, non-labeled tape for the stand alone dump program. Be sure that the tape is non-labeled, because the facility does not check to ensure that it is a non-labeled tape.

You can issue the SPTAPE command with the SADump option to move the data from the output tape to a class V reader spool file which is IPCS compatible. From that point on, you can invoke IPCS to handle the stand-alone dump.

Stand-Alone Dump Program Generation

Your installation can generate the stand-alone dump program to customize the facility to your system configuration. This gives you control over the device used to IPL the stand-alone dump program and the output device for the dump. Invoke the SADUMP EXEC in a CMS virtual machine to do the generation.

Do not call the SADUMP EXEC from within another EXEC. Also, do not queue up the answers ahead of time when running the SADUMP EXEC. To generate a stand-alone dump program, enter "sadump".

To use the SADUMP EXEC:

- You must have R/W access to the A-disk.
- The following files must exist on an accessed disk:

DMKSP CNTRL
DMKLD00E LOADER
LDT DMKSADWT
DMKSAD TEXT

You are asked to answer a series of questions that describe the environment where the stand-alone dump program will run. The SADUMP EXEC checks all input for validity, and returns messages if you enter invalid data. An example of the prompts and replies that appear on the virtual machine console during SADUMP EXEC execution is shown in "Example for Configuring the Stand-Alone Dump" on page 4-93.

Following the data that you provide, the SADUMP EXEC does one or more of the following:

- Creates a file with a name of SADGEN¹ ASSEMBLE, and places the file on the user's A-disk. The file has the SAD MACRO with the selected parameters.

¹ SADGEN is the default. The filename will be the same as specified in the SADUMP command if you do not use the default.

Service Programs

- Assembles the SADGEN file to create the SADGEN¹ TEXT file.
- Places the stand-alone dump program in the user's virtual card reader to be IPL'ed as desired. When the virtual reader is IPL'ed, the stand-alone dump program will be written on the IPL device.

Notes:

1. The device address from which the stand-alone dump program is IPLed does not need to be the same as the address where it was created.
2. It is impossible to verify the dump output address(es) and type at stand-alone dump generation time.

Using the Stand-Alone Dump Facility

To use the stand-alone dump facility:

1. Configure the stand-alone dump program.
2. Take the stand-alone dump.
3. Process the stand-alone dump from tape to a spool file.

Configuring the Stand-Alone Dump

Before you can use the stand-alone dump program, you must configure the facility. This lets you configure the IPL device and the dump output device(s) for the stand-alone dump facility to match the real I/O.

Use the SADUMP EXEC during configuration to create and assemble the SADGEN¹ ASSEMBLE file. The SADUMP EXEC places an IPLable deck in the virtual card reader. If the system detects an incorrect response, the exec gives an error message to you and requests a new response. For assembly errors, the exec will exit. If the stand-alone dump IPL device is a DASD, it must be CP formatted and the facility will use cylinder 0. Allocate cylinder 0 as permanent space.

The format of the SADUMP command line is:

SADUMP	{ <i>filename</i> }
	{ SADGEN }

filename

is the filename of the ASSEMBLE file that has the SAD MACRO. SADGEN ASSEMBLE is the default if you do not supply an operand. (Comply with the CMS guidelines for filenames if a filename is specified).

The following items apply to configuring the stand-alone dump:

1. If the system generated more than one stand-alone configuration, use unique names for each configuration. The default is SADGEN. If you answer "Y" to replace one that already exists (refer to "Example for Configuring the Stand-Alone Dump" on page 4-93), the original is erased.

2. If you respond with "N" to any of the questions (refer to "Example for Configuring the Stand-Alone Dump" on page 4-93), the exec will go directly to the next question without doing the indicated work.
3. The stand-alone dump configuration deck that the system puts in the virtual card reader is a class D file. You must place the deck in front of all the class D reader files before IPLing.
4. You must IPL the stand-alone dump reader file within a virtual machine. After the IPL'able deck is in your reader, perform the following instructions:
 - a. SET ECMODE ON
 - b. SPOOL 00C CLASS D
 - c. System CLEAR
 - d. IPL 00C

Note: Before you IPL the virtual reader, make sure that the IPL device is mounted and ready. If the IPL device is a tape, make sure the write ring is in.

Example for Configuring the Stand-Alone Dump

The following is an example of a stand-alone dump facility generation. In this example:

- You are placing the stand-alone dump program onto a 3330 device with an address of 150.
- The system will send the dump output to the first available 3420 tape drive whose address is 570, 571, 572, 573, 574, 575, 576, or 577.
- The filename of the file that has the SAD MACRO defaults to SADGEN.

Note: In the example, your input (user) is lower case and indented one space, system responses are upper case.

```
sadump
```

The SADUMP EXEC:

- OPTIONALLY CREATES A NEW SADGEN ASSEMBLE FILE CONTAINING A SAD MACRO WITH SELECTED PARAMETERS ON YOUR A-DISK.
- OPTIONALLY ASSEMBLES THE SADGEN ASSEMBLE FILE.
- OPTIONALLY PLACES A SADUMP CONFIGURATOR DECK INTO THE VIRTUAL CARD READER.

NOTE: YOU MAY EXIT FROM THIS EXEC BY ENTERING 'QUIT' FOR ANY RESPONSE.

Service Programs

DO YOU WANT TO CREATE A NEW SADGEN MODULE? (Y|N)

y

PLEASE ENTER THE VIRTUAL DEVICE ADDRESS TO WHICH THE SAD PROGRAM WILL BE WRITTEN (IPL DEVICE):

150

PLEASE ENTER ONE OF THE FOLLOWING IPL DEVICE TYPES:
DASD: (3330, 3333, 3340, 3344, 3350, 3375, 3380)
TAPE: (2401, 2415, 2420, 3410, 3420, or 3430)

3330

PLEASE ENTER ONE OF THE FOLLOWING OUTPUT DEVICE TYPES:
PRINTER: (1403, 1443, 3203, 3211, 3262, 3289, 4245, 4248)
TAPE: (2401, 2415, 2420 3410, 3420, OR 3430)

3420

PLEASE ENTER REAL OUTPUT DEVICE ADDRESS OR LIST ADDRESSES (MAXIMUM OF 8) FOR TAPE:
ENTRIES IN A LIST MUST BE SEPARATED BY A MINIMUM OF ONE BLANK.

570 571 572 573 574 575 576 577

DO YOU WANT TO ASSEMBLE SADGEN NOW? (Y|N)

y

THE SADGEN MODULE IS NOW BEING ASSEMBLED.
DO YOU WANT TO PLACE AN IPL'ABLE DECK IN YOUR VIRTUAL CARD READER? (Y|N)

y

AN IPL'ABLE DECK EXISTS IN YOUR VIRTUAL CARD READER IN CLASS D. IPL THE READER TO PLACE THE STAND-ALONE DUMP PROGRAM ON THE IPL DEVICE.
R;

Note: After you IPL your reader, if no errors occurred, you will receive a wait state code of 912.

Taking a Stand-Alone Dump

If you plan to dump 16 megabytes of storage, use a tape density of 1600 or 6250 bpi. A 16 megabyte dump may not fit on a tape at 800 bpi.

To invoke the stand-alone dump:

1. For multiple processor systems, stop all tightly-coupled processors. Do NOT clear storage.

2. For multiple processor systems, select the processor with the I/O configuration that has access to the resident volume address and the output device address(es).
3. Display locations X'0' to X'B' at the console. The stand-alone dump IPL sequence overlays these bytes, so they cannot be recovered.
4. Do a STORE STATUS operation on the CPU where you will IPL the stand-alone dump program. If you do not do the STORE STATUS, the following will not be saved in low storage:
 - CPU Timer
 - Clock comparator
 - Current PSW
 - Prefix
 - Model dependent features
 - Control registers
 - Floating point registers
 - General registers.

If the prefix value is not saved in low storage, the information from the prefix page is not available for the formatted section of the dump.

5. Mount and ready the volume that has the stand-alone dump program. If this is a tape, be sure to have the write ring in place.
6. Ready the output device, either tape or printer. If you want the system to place the stand alone dump on the IPL tape, make all other tapes listed as output devices (at generation time) NOT ready. If you do not want the stand alone dump on a device that is listed as a possible output device, the device must not be ready at the time you IPL the stand-alone dump program.
7. IPL the stand-alone dump program. The stand-alone dump program will initially write the first nine pages of storage to the IPL device. This will provide an area to load the stand-alone dump program and work space. (See "Tape Format" on page 4-96 and "DASD Format" on page 4-98 for information about the DASD or tape format.) This step causes the system to place the dump on the output tape or the printer. (See "Tape Format" on page 4-96 and "Printer Format" on page 4-99 for information about the format of the output.)
8. When the system enters the wait state, display the PSW. A wait state of 912 indicates successful completion of the stand-alone dump. If the stand-alone dump program is unsuccessful because of some error that you can fix (for example, an unrecoverable I/O error on the output tape):
 - a. Correct the error.
 - b. Invoke a hardware RESTART to restart the stand-alone dump. (For example, type in RESTART on the appropriate panel on a 4300 processor.)

Service Programs

If you re-IPL the stand-alone dump facility again, part or all the first nine pages of storage will be invalid. After the initial IPL, you cannot change the IPL address or IPL volume.

Processing the Stand-Alone Dump Data on Tape

If you directed the output to tape, re-IPL VM/SP. Then issue the SPTAPE command with the LOAD raddr and SADUMP operands to create an IPCS compatible spool file. This is the only way to transfer the data. After the system has created the spool file, enter the IPCS command, IPCSDUMP, to view the stand-alone dump like you would view a CP Abend dump.

IPL and Dump Formats

Tape Format

A tape used with the stand-alone dump facility has the format shown in Figure 4-19 on page 4-97.

- If the IPL tape and the dump device are not the same, the IPL tape includes sections A, B, and C.
- If the dump device is a tape, but not the same tape as the IPL tape, the dump output tape includes sections C and D.
- If the IPL device and the dump device are the same, the tape includes sections A, B, C, and D.

Service Programs

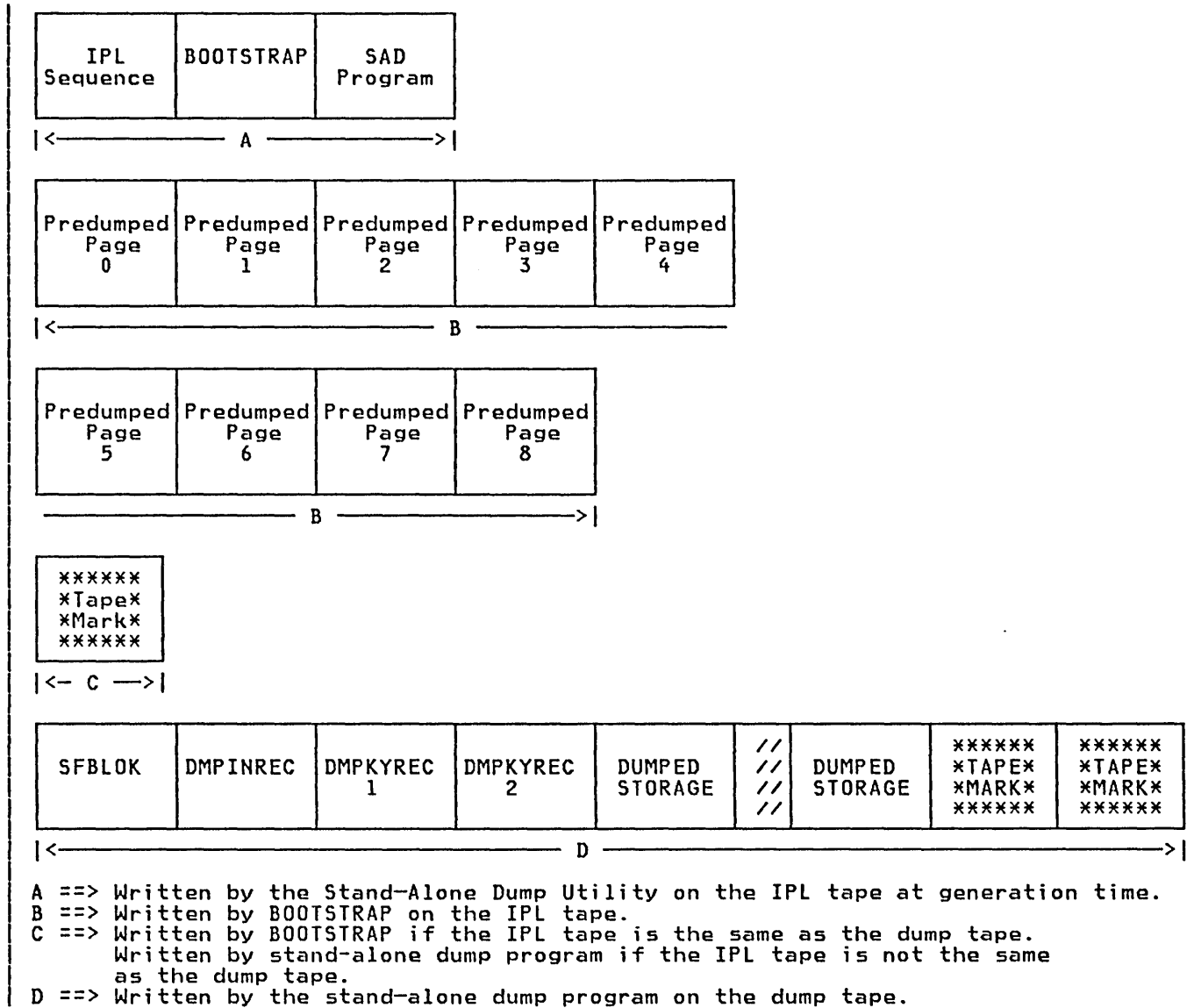
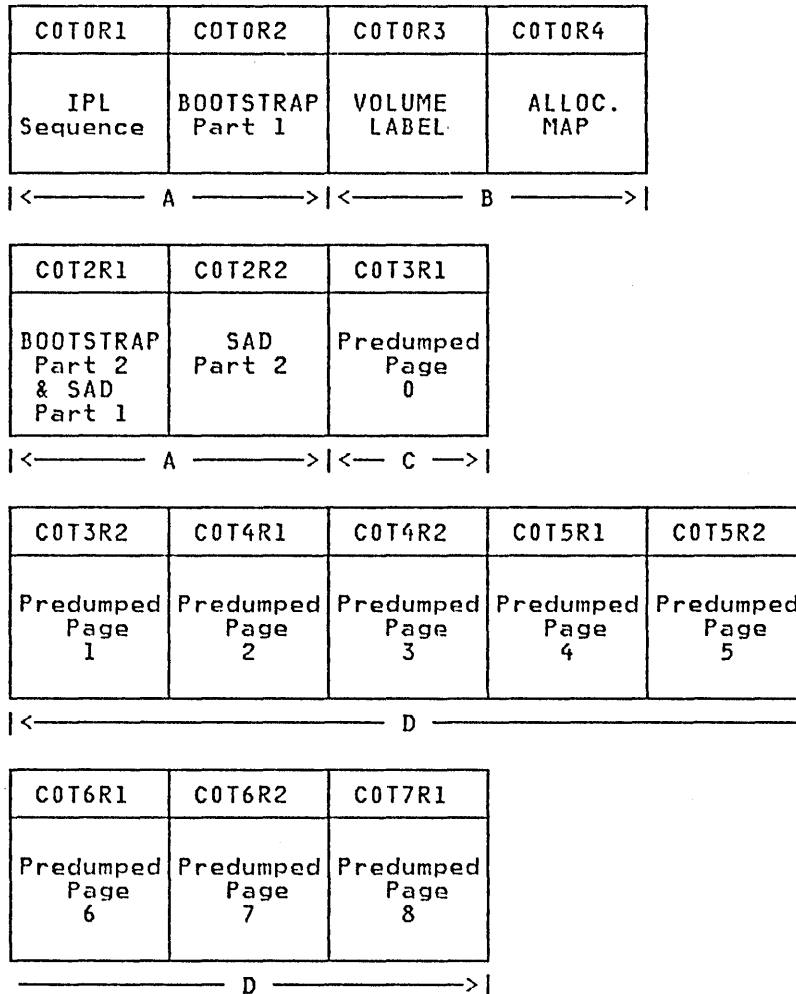


Figure 4-19. Stand-Along Dump Facility Tape Format

Service Programs

DASD Format

When you use a DASD device to IPL the stand-alone dump program, the system uses cylinder 0 to hold the program. Cylinder 0 must be CP formatted and allocated as permanent space. The stand-alone dump facility has the format shown in Figure 4-20.



- A ==> Written by the Stand-Alone Dump Utility on the IPL DASD at generation time.
 - B ==> Written by FORMAT/ALLOCATE program.
 - C ==> Written by BOOTSTRAP Part 1 on the IPL DASD.
 - D ==> Written by BOOTSTRAP Part 2 on the IPL DASD.
- CnTnRn identifies cylinder, track, and record numbers.

Figure 4-20. Stand-Alone Dump Facility DASD Format

Printer Format

Dumps to printer devices are printed as follows:

- CP formats the following data fields for each processor, beginning with the processor where the stand-alone dump program was IPLed:
 - CPU address (only if in AP or MP mode)
 - General purpose registers
 - Control registers
 - Floating point registers
 - Clock comparator
 - CPU timer values
 - Stored-status PSW
 - Prefix value (only if in AP or MP mode)
 - External interrupt old/new PSWs
 - SVC old/new PSWs
 - Program check old/new PSWs
 - Machine check old/new PSWs
 - I/O interrupt old/new PSWs.
- The following fields are printed for the processor where the stand-alone dump program was IPLed:
 - TOD clock
- Lines of duplicate data will have a suppression message after the first line of the data is printed.
- A half page (2048 bytes) of all zeros has one line of zeros printed with the key, followed by a line suppressed message.
- The dump page is interpreted on the right-hand side of the printout.

Error Handling

Basic error recovery is available for DASD, tape, and printer devices used as IPL or output devices. In addition, the following information may be of value when the system detects errors:

- The Channel Status Word is at location X'40'.
- The I/O address is at location X'BA'.
- 32 bytes of sense data are at location X'2E0'.
- The starting and ending addresses of the CP Trace Table are stored in the PSA at X'7B0' and X'7B4', respectively, in addition to the low storage locations.

Under certain error conditions, storage areas may be overlaid. This could cause fields in SFBLOK and DMPINREC to be incorrect. (For example, fields containing date and time information.)

Service Programs

Section 5. Operator Spooling Functions

Spooling

Input and output files for use and access by virtual machines through unit record devices are maintained by CP as disk data files using a mechanism called spooling. Individual files can be identified and manipulated using various console functions. The disk records are chained to form a logical file from dynamically assigned areas on specially formatted CP disk areas. Data records from disk are read into available page space obtained through the CP paging mechanism. The data records contain the actual data to be used and the CCWs to properly control the format. These CCWs are directly executed to perform the actual unit record operations on the real hardware. CP can support any number of virtual and real unit record devices given sufficient system resources.

The data is placed in the spool buffers through the virtual machine unit record simulation routines in CP. Certain spool files have a special data format (system dumps, for example) and are accessed using a special interface.

Spooling Considerations

VM/SP spooling facilities allow several virtual machines to share one or more unit record devices. Since virtual machines controlled by CMS ordinarily have modest requirements for unit record I/O, such device sharing is quite advantageous, and it is the standard mode of system operation.

Each user has, as a general rule, a virtual reader, a virtual punch, and a virtual printer as his spooling devices. In addition, the virtual console can also be classified as a spool file generator as all input and output to the console can be logged on a spool file. This console log, and the files created by the user's virtual spooling devices, can be processed by the real unit record devices that attach to the system.

CP controls and schedules the operation of the real unit record devices via spooling techniques. Virtual machine SIO instructions directed to those unit record devices designated as spool devices in the user directory entry are intercepted and modified by CP. CP generates another I/O operation, transparent to the virtual machine, which replaces the one specified. The new operation is directed to a CP spooling disk area which acts as intermediate storage between the real unit record device and the virtual machine. The data transfer operation between a spooled unit record device and the virtual machine is, in reality, between a CP spool file and the virtual

Spooling Functions

machine. Spool file records are page size (that is, 4096 byte blocks), and are transferred between storage media via the CP paging mechanism.

When the system informs you that the spooling space is full or nearly full of spool files, you can use the SPTAPE command to write those unit record files to tape to relieve the situation. Then, when spool space is more plentiful you can use the SPTAPE command to read the spool files back to the system for eventual spool file output. See the description of the SPTAPE command in Section 3, "CP Commands."

Spooling Functions

CP spooling support performs the following five functions for virtual machines:

1. It simulates, with software routines, the operation of the virtual unit record devices that are attached to each user's virtual machine. The simulation makes it appear that the program in the virtual machine controls real unit record devices. Unit record device simulation involves the interception and interpretation of user Start I/O (SIO) instructions, the movement of data to and from the user's virtual storage space, and the reflection of interruption codes and ending conditions in the virtual machine.
2. It operates the real unit record equipment attached to the real machine. Spooling moves print-image and punch-card-image files to a real printer and card punch, and creates spool files from data read in from the real card reader.
3. It provides an easy-to-use interface between the virtual machine users, the system operator, and the spooling system to allow flexible and easy switching of system resources between many users. A set of general user and operator commands can request these functions.
4. It allows keyed-in CP commands and responses as well as virtual machine console input and output to be placed on disk instead of, or in addition to, being displayed at the terminal. You can initiate or terminate virtual console spooling at any time during a terminal session.
5. It provides the ability to spool files across the RSCS teleprocessing network.

Spooling commands can be separated by type into those that affect virtual devices, those that affect real devices, and those that affect queued spool files within the system. The commands that affect virtual devices are available to all general users; a user may affect the status of devices that attach to only his virtual machine. For a discussion of the Class G spooling commands, see *VM/SP CP Command Reference for General Users*.

Only the spooling or resource operator can use commands that affect the status of the real spooling devices. Commands affecting the user's virtual machine closed spool files that are awaiting processing are available to all users, with some additional capabilities available to the spooling operator. For example, a user can alter characteristics of only those files that have the same userid as his, but the spooling

operator can change the status of any spool file in the system. These commands are described in Section 3, "CP Commands."

You can create input spool files (that is, data available at a specific user's virtual card reader), when you place cards in the real card reader that are preceded by a special VM/SP card that identifies the virtual machine userid of the user requesting spool files.

When the virtual machine operating system writes to a virtual printer or card punch, it creates output spool files and stores them on direct access devices. Real output is scheduled for a real printer or card punch, or for remote output, whenever a user logs off the system or issues a CP spooling command to close the file.

You can transfer specific files from the spooled printer or card punch of a virtual machine to the card reader of the same or another virtual machine. (A virtual card reader is not limited to 80-character records.) Files are not physically printed or punched when transferred between virtual unit record devices by the spooling routines. With this method, you can make files available to multiple virtual machines, or to different operating systems executing at different times in the same virtual machine.

The Remote Spooling Communications Subsystem Networking, with the CP spooling system, provides support for spooling across a teleprocessing network. For detailed information about RSCS, see the *VM/SP Program Reference and Operations Manual*.

VM/SP spooling includes many options for the virtual machine user and the real machine operator. These include printing multiple copies of a single spool file, backspacing any number of printer pages, and defining or reordering spooling classes and forms for real output scheduling.

Real printers with the Universal Character Set feature must have the block data check set when using VM/SP spooling. The LOADBUF command automatically sets the block data check. For more details on spooling, see "Spooling Considerations" on page 5-1.

Virtual Console Spooling

The terminal user can spool virtual console output and virtual console input keyed in by him in addition to typing or displaying it at the terminal. Furthermore, if the virtual console is disconnected and the virtual machine is active, the console spool file acquires all console output that would normally print or display at the terminal. Virtual console I/O consists of all initiated CP commands and responses, as well as the communication to and from the operating system running in the virtual machine.

You start virtual console spooling when you issue the command:

```
SPOOL CONSOLE START
```


Spooling Functions

You close the console spool file at logoff time or when you stop and close the file. When you close the console spool file, it goes on the printer spool file and then the operator can manipulate it in the same way as any other printer spool file.

The display output of local or remote 3270s in DISPLAY mode (every 16 lines) is written automatically on the spool file if the operator is the system operator and the device was specified as a graphic device supported as a virtual machine operator's console.

Console spooling for the system operator continues even if the operator disconnects from one terminal and logs onto another. To stop the automatic console spooling, the operator must issue the SPOOL CONSOLE STOP command. If the system fails, up to 16 lines of output may be lost from the system operator's spool file, but the system closes the console spool file when the failure occurs. However, virtual machine operators may lose up to one page of spooling data in a system failure because their spooling buffer is one page (4096 bytes) in size. In regard to DASD I/O, errors occurring on spool and paging devices produce error messages that appear on the system operator's terminal.

A user can start or stop virtual console spooling any time during a terminal session with the SPOOL CONSOLE command. For a description of the CP SPOOL and CLOSE commands, see the *VM/SP CP Command Reference for General Users*.

Note: The user cannot use console spooling to stack commands for subsequent execution. The console spool file is for historical purposes only. In addition, the following types of data will not be placed in the console spool file for the indicated conditions:

- CP command output - if this is being received in a buffer via DIAGNOSE 8
- Messages and Warnings - if they are being trapped via the IUCV and MSG System Service.

I/O Error Handling

If a permanent I/O error occurs during a real I/O operation, you restart the spooled files as follows:

- Printer files - from the beginning of the current page (the last skip to channel 1).
- Punch files - from the beginning of the spool file.
- Reader files - from the beginning of the real deck (the operator must normally place the card deck back in the reader).
- Console files - do not restart. CP ends console spooling and purges the console file. The virtual machine continues without further console spooling.

Note: If an I/O error occurs while punching accounting cards, CP will repunch only the error card and not the whole file as in the case of normal punch spool files.

Spool Buffers

The buffers used in virtual machines while collecting and writing spool data are each one page (4096 bytes) long, and contain both the data to be transcribed and all CCWs necessary for operating the unit record devices that perform the transcription. System failures that end system operation can cause virtual machine operators to lose up to one page of spooling data.

Buffers used for the temporary storage of spool data on its way between auxiliary storage and the user's virtual machine are allocated from a pool of virtual pages that belong to CP. Direct access devices provide the auxiliary storage necessary for CP spool buffering.

CP uses unit record devices to spool input and output operations. However, certain features or functions that pertain to this group of machines are not supported. Consult the VM/SP restrictions listed in the *VM/SP Planning Guide and Reference*

Spool Files

Each spool file in the system has a number of attributes that are assigned to it, either explicitly or by default, at the time that it is created. These attributes and their values are as follows:

Filename and Filetype: Each of these consist of alphanumeric fields of up to eight characters. The CHANGE or CLOSE command can change these fields.

Spoolid: This is a system-assigned number between 1 and 9,900. It is automatically assigned when the file is closed, and is unique to that spooled file. To identify a given file, it is necessary to specify the userid of the file's owner, the device type, and the spoolid number. In most instances, the userid defaults to the ID of the user issuing the given command. Since the ID number, rather than the filename and filetype, is used as an identifier, duplicate user-assigned names do not present an identification problem.

Owner's User ID: This is the ID of the current owner of the file.

Originating User ID: This is the ID of the file's creator. This is the same as the current owner, unless the originator has used the SPOOL or TRANSFER commands to send the file to another user.

Number of Copies: This is the number requested for an output spool file; it is a number between 1 and 255. Unless specified by the user or operator, it defaults to 1.

Date and Time: This is the date and time that the file was created. Users create most files from virtual spool devices and the date and time indicate when the originator of the file closed the device.

Number of Records: This is an eight-digit number indicating the number of logical record (printer or card images) in the file and is an indication of the size of the file.

Spooling Functions

Distribution Code: This is an eight-character value that is normally assigned to each user by the system administrator, although users may alter their assigned value. It is printed in large block letters on the separator pages, and punched in the separator cards. It may assist you in distributing output to users.

Hold Status: A spool file may be in “user hold,” “system hold,” or both. User hold can be removed by the user, but only the operator can remove system hold. Either hold status prevents a spool file from being printed or punched.

3800 Status: There are several attributes that apply only to printer files on the 3800 Printing Subsystem. See the discussion of the SPOOL command in the *VM/SP CP Command Reference for General Users*, and the START command in “START” on page 3-179 for additional information.

Output Classes

In addition to the attributes described above, a file queued for output on a real unit record device always has an output class associated with it. A single alphanumeric character (A through Z, 0 through 9, and the special character *) controls what real device the file is to be printed or punched on, and the relative priority and sequence of output on the device. Although each file is assigned a single class, you can designate each real spooling output device to handle from one to four classes. The device processes only files that have a class code that corresponds to one of its own, and processes these files in the order you specify. For example, if you assign a printer the classes A, D, and 2, it always processes any printer file with class A before it searches the printer output queue for a file with class D, and all class D files are printed before any file with class 2. You can assign a class of * to a real output unit record device to allow that device to process any class spool file.

The output class for a file is assigned at the time of the file’s creation and is the class that is associated with the virtual device that created it. When a user logs on to the system, the class associated with the device is the one defined in his user directory entry for that device; however, he may alter this class at any time with the SPOOL command. After files are closed and are awaiting output, their class can be altered with the CHANGE command either by the file’s owner or by the spooling operator. The spooling operator can alter the standard assignments of output classes of a real output device with the START command.

Output priorities can also be rearranged by altering the hold status of a file. The system operator can hold or release the files of specific users, all files of a given type, or all files in the system. The hold function can ensure the immediate processing of high priority output files, while allowing high volume or low priority jobs to be deferred for output until periods of low system usage.

Output Forms

An output file has two form numbers associated with it - the user form and the operator form. Each form is one to eight characters long. Any combination of letters, numbers, or special characters may be used.

The user form is assigned by the user when he creates the spool file. He can later change it.

The operator form is assigned by the system when the spool file is created. The system looks in a table for the file's user form. The table provides the corresponding operator form. If the user form is not found in the table, the operator form is set equal to the user form. The table is created by your installation's system programmer using the SYSFORM macro. You can change the operator form of any spool file with the CHANGE command.

When you start a real spooling printer or punch, you can specify the form number that device is to process. Spool files with the specified operator form number will be processed if the class matches and they are not held. This is called MANUAL mode.

You can also specify AUTO move when you start the real spooling printer or punch. In this mode, the system will process files with any form number. The system will send you the following message and sound the audible alarm each time a new form is selected:

```
{ PRT } raddr MOUNT REQ CLASS a... FORM form AUTO { SEP }  
{ PUN }                                     { NOSEP }
```

raddr
is the real address of the printer or punch.

a...
are the classes that the device is processing.

form
is the form number for which a mount is requested.

The system then waits for you to make any necessary manual device adjustments (such as mounting new forms). You then press START on the device, or enter the command "START raddr" at the console. The printer or punch device will then print or punch the file.

All other files with the same form number will be printed or punched without further intervention.

Some special printer forms require careful alignment on the printer (pre-printed forms, for example). SETUP mode is provided to assist you in performing this alignment. If you start the printer in SETUP mode, all of the steps described above for AUTO mode are followed. However, when you press START on the printer, only the first page of the file prints, with letters replaced by X's and numbers replaced by 9's. You can then manually adjust and align the printer. Each time

Spooling Functions

you press the START button, another page of the file is printed, and you can further adjust the forms. When the forms are properly aligned, enter the command "START raddr" at the console. The file will then print in its entirety.

At any time during the SETUP process, you may use the FLUSH command to terminate the process, or you can use a BACKSPAC command to return to the previous page of output.

Spooling Commands

The commands shown in Figure 5-1 control VM/SP spooling operations.

Command	User Classification	
	General User	Spooling Operator
BACKSPAC		X
CHANGE	X	X
CLOSE	X	
DRAIN		X
FLUSH		X
FREE		X
HOLD		X
LOADBUF		X
ORDER	X	X
PURGE	X	X
REPEAT		X
SPACE		X
SPOOL	X	
SPTAPE		X
START		X
TAG	X	
TRANSFER	X	X

Figure 5-1. CP Spooling Commands

This list represents all the CP commands that pertain to spooling operations. Users having privilege classes D (the spooling operator) and G (the general user) can issue these commands. Users with only the G privilege class cannot invoke class D commands.

The spooling operator has responsibility for the following unit record spooling operations. He must:

1. Verify that the special VM/SP user identification (ID) card precedes a user's physical deck of cards to associate the cards with the correct user's virtual machine (See "Spooled Card Input" on page 5-9.).
2. Make sure that sufficient blank cards are available for the punch unit, and that the paper in the printer is the proper size, width, and number of copies, and that the proper print train is mounted and the print buffer is properly loaded.
3. Separate punched and printed output by user identification (userid) code.

4. Start, stop, restart, or rearrange the sequence schedules of spool files to be printed or punched.

The spooling commands are discussed (along with other operator commands) in Section 3, "CP Commands."

NOLOG Password

The NOLOG password is reserved for users who are not assigned any privilege class. These users cannot log on to a virtual machine. The user with the NOLOG password can submit jobs to the CMS Batch facility via the real card reader. The user with the NOLOG password can also be the recipient of spool files sent by any other logged-on user only if the other user issued a SPOOL FOR command to the userid having the NOLOG password.

Spooled Card Input

Spooled input from a real card reader requires a CP ID (identification) card before the VM/SP system can accept it for processing.

Note: Load cards for only one user at a time. VM/SP does not look for ID cards in the middle of the card deck.

The CP ID card must be inserted in front of the card deck. Then place the deck in the empty feed hopper. Then ready the reader device and push the EOF button. The reader immediately processes the cards (if no errors are encountered) and places the file on a direct access device. The file now resides in the virtual machine card reader that was identified by the CP ID card. Note that a file is transferred from the real reader to the virtual reader regardless of whether that user is logged on or off. The user may log on at any time to read the file.

CP ID Card Format

ID USERID	userid [CLASS n] [NAME {filename filetype} dsname}]
--------------	--

1. Begin punching in column 1.
2. The first field must be one of the following:
ID
USERID
3. Separate all fields by one or more blanks.
4. The second field must be the user's identification (userid), limited to eight characters.

Spooling Functions

5. The third field (optional) is CLASS n (if class is assigned); one or more spaces must separate the word CLASS and the class designation (n). (A through Z and 0 through 9 are valid class designations.) If the class option is not coded, the file defaults to class A. If the class option is coded, it must immediately follow userid.
6. The fourth field (optional) is the keyword NAME, followed by filename and filetype (or dsname alone). If only a dsname is specified, it can be 24 characters long. If both filename and filetype are coded, they are each restricted to a maximum of eight characters and must be separated by a blank.

An invalid or missing user CP ID card on the front of a card deck, when read, causes an error message on the system console.

Spool Printer and Punch Output

Printed and punched output from virtual machine users is directed to the appropriate real unit record device. The operator (class D) can control this spooling output by combinations of output class, form number, hold status, and other spool command operands as discussed earlier.

Unless you used the NOSEP option to "START" a device, VM/SP prints a two-page header and a one-page trailer between all output on printers and punches a four-card separator between all output on punch devices.

The printer header gives the pertinent spool file characteristics as well as real device data. The header prints the owner userid and distribution code in block letters on two pages. The header does not print between multiple copies of the file.

The card separator gives the owner's userid and the distribution code. The distribution code is expanded to punch each character in four columns with two separating blanks so that visual interpretation of the punched data is easier.

The printer trailer prints the sequence number in large block letters. The sequence number is maintained for each printer in the system. It starts at 1 and increases by 1 for each file printed on that real printer. When it reaches 999, it recycles to 1 again. The sequence number is also printed on the header pages. It will assist you in separating and distributing output.

The SYSPCLAS macro instruction is available to classify printed output with a classification title. This classification title is printed on the output separator page and optionally at the top or bottom of each page of output.

Because output for a user file starts on a real device, the following message occurs:

```
{PRT}  raddr  {PRINTING}  USERID FILE  CLASS RECORDS RECLEFT CPY DIST SEQ
{PUN}                                     {PUNCHING}  userid file  a typ nosecs  recleft nnn dist sss
{REPEATED}
```

raddr
is the real device address.

Spooling Functions

`user id`
is the identification of the owner of the spool file.

`file`
is the spool file spoolid number.

`a`
is the spool file class.

`typ`
is the originating device typ (PRT, PUN or CON).

`norecs`
is the logical record count (lines or cards) for the file.

`rec left`
is the number of records left to be printed or punched. This number is initially the same as "norecs" but decreases as the file is printed or punched.

`nnn`
is the current copy number being produced where 001 is the last copy.

`dist`
is the spool file distribution code.

`sss`
is the sequence number of this file on this device. It is printed in large block letters on the header and trailer pages.

The PRINTING/PUNCHING message occurs when the file output first starts and the REPEATED message occurs for subsequent copies of the same file as they start.

When the card deck is read in, the operator receives the following message:

```
RDR raddr { READING } USERID FILE CLASS RECORDS  
          { HAS READ... } user id file c RDR norecs
```

`raddr`
is the real address of the card reader.

`user id`
is the user identification of the owner of the file from the ID card at the front of the deck.

`file`
is the spool file spoolid number assigned.

`c`
is the spool file class from the ID card at the front of the deck.

Spooling Functions

`norecs`

is the number of cards read. Initially zero, this number increases as the file is read.

The `READING` message occurs when the ID card is successfully read. The `HAS READ` message occurs when the entire file is read and the reader stops.

An invalid or missing user identification card on the front of a card deck, when read, causes an error message on the system console.

When the last file for an output device is processed and the device becomes idle, the following message is produced:

```
{PRT} raddr WAITING CLASS c... FORM form {MANUAL} {SEP}
{PUN}                                     {AUTO}  {NOSEP}
                                     {SETUP}
```

`raddr`

is the address of the device.

`c...`

is the class(es) the device will process.

`form`

is the form number the device will process.

Spool Files for Virtual Machines Running Batch Jobs

Usually, when an installation has only one real reader, printer, or punch, the unit record devices must be shared by all virtual machines. Before VM/SP can process any virtual machine CP spool files and direct them to a real device, those files must be closed. For virtual machines executing batch jobs and using operating systems other than CMS, usually the operator who initially program loaded the virtual machine operating system must intervene to issue the `CP CLOSE` command.

A feature that automatically closes CP spool files is available for OS/VS1 and DOS/VSE with VSE/AF. Once output files are closed, they can be processed by VM/SP without operator intervention.

Without this feature, CP spool files are not sent to the real printer or punch until the virtual machine operator intervenes.

Using the CMS Batch Facility

The batch facility is a VM/SP programming facility that runs under CMS. It allows a VM/SP user to run jobs in batch mode by sending jobs from either his own virtual machine or the real card reader to a virtual machine dedicated to running batch jobs under the batch facility. This dedicated machine is generally set up at a terminal in the installation's computer room and is controlled by the system operator.

The batch facility virtual machine runs continuously, executing all jobs spooled to its virtual card reader from other virtual machines or from the real card reader. The batch operator need pay no attention to the batch machine once he has started, and disconnected it.

Starting the Batch Virtual Machine

The system operator starts the batch virtual machine by logging on with a batch userid and loading CMS using the CP IPL command.

The Batch Userid

Every installation in which the CMS batch facility is available should establish one or more common userids for the Batch Monitor virtual machine. Users can then spool their files for execution to the card reader for that batch userid.

It is the operator's responsibility to log on the VM/SP system using the batch userid that has been established for his installation.

The userid established for the batch facility virtual machine must have a read/write disk in its directory at virtual address 195. The 195 disk is erased when it is accessed as the A-disk at the beginning of each job.

Invoking the Batch Facility

The batch facility virtual machine is invoked by the batch operator when he issues the CP IPL command with PARM BATCH or CP IPL followed by the CMSBATCH command. The latter method follows:

```
ipl cms
CMS mm/dd/yy WED 17.58.48
```

```
cmsbatch
Y/S (19E) R/O.
THE FOLLOWING NAMES ARE UNDEFINED:
  BATEXIT1 BATEXIT2
R; T=0.14/0.39 08:47:40
WAITING FOR THE READER
```

The operator may now disconnect the batch machine terminal, if he wishes, using the CP DISCONN command. The batch facility will IPL itself after each job is executed.

All virtual machine (CMS) console output is automatically spooled to a file to be printed after the program output at the real system printer. All commands entered through the virtual reader are displayed on the console to allow them to appear in the console output file. If the batch terminal is disconnected, only CP and batch initialization messages are displayed at the terminal.

If an installation wishes to use a saved system in running batch jobs, the operator must enter the name of the saved system in the CMSBATCH command line.

Spooling Functions

Controlling the Batch Facility

The batch facility permits CP and CMS commands to be issued from the batch job stream submitted by the user.

The following CP commands, and only these commands, control the batch virtual machine:

CHANGE

May not be used to change the virtual card reader.

CLOSE

May not be used to change the virtual card reader.

DETACH

May not be used to change the virtual spool devices or the system, IPL, or 195 disks.

DUMP

DISPLAY

LINK

A maximum of 10 links may be in effect at any one time. LINK must be used as shown in the following format:

CP LINK *userid vaddr vaddr mode password*

MSG

QUERY

SET

Only the following SET command operands are allowed under CMS batch:

ABBREV DOS IMPEX LDRTBLS NONSHARE RDYMSG SYSNAME
UPSI

SPOOL

May not be used to change the virtual card reader.

STORE

TAG

When CP commands are used to control the CMS batch virtual machine, they must be preceded by the CMS CP command because the implied CP function is disabled when batch initializes a job.

The following CMS commands are disabled under the Batch Monitor:

READCARD

DISK LOAD

The FILEDEF and ASSGN commands are also disabled for use in defining the virtual card reader.

A more complete description regarding control of the CMS Batch virtual machine is contained in *VM/SP CMS User's Guide*. It describes the user control cards and suggests control techniques as well as how to control the batch machine using EXEC procedures.

The batch virtual machine spools output resulting from program execution to the system printer. Output is printed under the submitting userid, with the submitting userid's distribution code, a spool filename of CMSBATCH, and a spool filetype of JOB (unless a job name was specified on the /JOB card).

The console output is always spooled. Therefore, if the console is disconnected, the CMS console output is spooled to a file that is printed following the user's program execution output at the real system printer, with the submitting userid as distribution code, a spool filename of BATCH, and a spool filetype of CONSOLE.

If the CP TAG command has been used to identify spool files or to direct these files to other virtual machines or remote work stations, BATCH resets the spooling devices for the next job.

Purging, Reordering, and Restarting Batch Jobs

When required, the spooling operator can control the execution of batch virtual machine jobs by purging, reordering, and restarting them; by the same token, because all the closed printer files are queued for system output under the submitting userid, the submitting user can change, purge, or reorder these files prior to processing on the system printer.

To purge a job executing under the Batch Monitor, follow the procedure below:

1. Signal attention and enter the virtual machine environment.
2. Enter the HX (halt execution%) immediate command.
3. Disconnect the virtual machine using the CP DISCONN command.

The HX command causes the Batch Facility to abnormally terminate. This provides the user with an error message and a CP dump of the Batch Facility virtual machine. The Batch Monitor then loads itself again and starts the next job (if any).

To purge an individual input spool file that is not yet executing, issue the CP PURGE command:

```
PURGE READER spoolid
```

In the format above, spoolid is the spool file number of the job to be purged from the batch virtual machine's job queue. For example, the statement:

```
PURGE READER 123
```

would purge 123 from the batch virtual machine's job queue.

Spooling Functions

To reorder individual spool files in the Batch Facility's job queue, use the CP ORDER command:

```
ORDER READER spoolid1 spoolid2...
```

In this format, spoolid1 and spoolid2 is the assigned spool file identification of the jobs to be reordered.

The operator can determine which jobs are in the queue by using the CP QUERY command:

```
QUERY READER ALL
```

This QUERY command lists the filenames and filetypes of all the jobs in the batch virtual machine's job queue. The operator can then reorder them, using the ORDER command.

Stopping the Batch Virtual Machine

To stop batch virtual machine execution after completion of the current job, issue the HB immediate command and press the attention key or equivalent to cause an attention interruption at any time during the job. This causes the batch virtual machine to be logged out at job completion.

When batch facility virtual machine execution is to be stopped immediately, but current files must be saved, you can use the CP SPOOL command, in the form SPOOL READER HOLD, and then issue the CP LOGOFF command.

The HOLD option causes CP to retain the virtual machine's current card reader file, so that when the batch facility is logged on again, execution resumes at the beginning of the held reader file.

If an emergency should occur, all jobs in the batch reader and all spool files are saved.

Appendix A. Summary of CP Commands

This appendix lists the following:

- The CP commands in alphabetical order
- The IBM-defined privilege classes that can execute the command
- The corresponding function type
- A blank column for you to record the user-defined classes
- A brief description of each command.

Note: Brackets indicate type is optional in the OVERRIDE statement.

COMMAND	FUNCTION TYPE*	IBM-DEFINED PRIVILEGE CLASS	USER-DEFINED CLASS	Usage
*	N/A	Any	None	Annotate the console sheet.
#CP	N/A	Any	None	Execute a CP command while remaining in the virtual machine environment.
ACNT	<O>	A		Create accounting records for logged on users, and reset accounting data. ACNT also closes the spool file that is accumulating accounting records.
ADSTOP	<G>	G		Halt execution at a specific virtual machine instruction address.
ATTACH	<R>	B		Logically connect a real device to a virtual machine for that machine's exclusive use or logically connect a DASD device for CP access and control. With CHANNEL operand, dedicate all devices on a particular channel to a specific user.

COMMAND	FUNCTION TYPE*	IBM-DEFINED PRIVILEGE CLASS	USER-DEFINED CLASS	Usage
ATTN	<G>	G		Make an attention interruption pending for the virtual machine console.
AUTOLOG	<O>	A,B		Log on any virtual machine defined in the directory.
BACKSPAC	<S>	D		Restart or reposition the current output on a real punch or printer.
BEGIN	<G>	G		Continue or resume execution of the virtual machine at either a specific storage location or at the address in the current PSW.
CHANGE	S	D		Alter one or more external attributes of a closed spool file or files.
	G	G		Alter one or more attributes of a closed spool file.
CLOSE	<G>	G		Terminate spooling operations on a virtual card reader, punch, printer, or console.
COMMANDS	N/A	Any	None	Display the commands and diagnose codes you are authorized to use.
COUPLE	<G>	G		Connect channel-to-channel devices.
CP	N/A	Any	None	Execute a CP command while remaining in the CMS virtual machine environment.
CPTRAP	<P>	C		Create a reader file of selected trace table, CP interface, and virtual machine interface entries for problem determination.
DCP	<P>	C,E		Display the contents of real storage locations at the terminal.
DEFINE	R	B		Redefine the status of a 3330V volume.
	G	G		Reconfigure your virtual machine.
DETACH	R	B		Remove a real device from the CP system. With the CHANNEL operand, remove a dedicated channel from a user.
	G	G		Detach a virtual device from a virtual machine. Detach a channel from your virtual machine.
DIAL	N/A	Any	None	Connect a terminal or display device to the virtual machine's virtual communication line.

COMMAND	FUNCTION TYPE*	IBM-DEFINED PRIVILEGE CLASS	USER-DEFINED CLASS	Usage
DISABLE	<R>	A, B		Prevent low-speed communication lines from accessing the system.
DISCONN	N/A	Any	None	Disconnect your terminal from your virtual machine.
DISPLAY	<G>	G		Display virtual storage on your terminal.
DMCP	<P>	C,E		Print the contents of real storage locations on a user's virtual spooled printer.
DRAIN	<S>	D		Stop spooling operations on a specified real unit record devices after the file currently being processed has been completed.
DUMP	<G>	G		Print the following on the virtual printer: Virtual PSW, general registers, floating-point registers, storage keys, and contents of specified virtual storage locations.
ECHO	<G>	G		Test terminal hardware by redisplaying data entered at the terminal.
ENABLE	<R>	A,B		Enable the previously disabled or nonenabled devices so users may access the system.
EXTERNAL	<G>	G		Simulate an external interruption for a virtual machine and return control to that machine.
FLUSH	<S>	D		Halt and immediately purge or hold the current output on a specified real unit record device.
FORCE	<O>	A		Force a logoff of any user on the system.
FREE	<S>	D		Remove a set of spool files belonging to a specified user from a system hold status.
HALT	<O>	A		Terminate any active channel program on a specified real device.
HOLD	<S>	D		Place user spool files in a system hold status.
INDICATE	O	A		Provides a list of statistics for all users who have the favored execution option.
	A	E		Display, at the console, the use of and contention for system processor and storage.

COMMAND	FUNCTION TYPE*	IBM-DEFINED PRIVILEGE CLASS	USER-DEFINED CLASS	Usage
	G	G		Indicate resource utilization and contention.
IPL	<G>	G		Simulate IPL for a virtual machine.
LINK	<G>	G		Provide access to a specific DASD by a virtual machine.
LOADBUF	<S>	D		On a 1403 printer load the Universal Character Set(UCS) with a specified print chain/train image. On 3203, 3211, 3262, 4245, or 4248 printers, load UCS or Forms Control Buffer(FCB) with a specified image. On 3289 Model 4 printer, load the Font Offset Buffer (FOB) with the image print belt and FCB.
LOADVFCB	<G>	G		Load virtual forms control buffer for a virtual 3203, 3262, 3289E, 3211, 4245, or 4248 printer.
LOCATE	<P>	C,E		Find the addresses of CP control blocks associated with a particular user, a user's virtual device, or a real system device.
LOCK	<O>	A		Permanently locks in selected pages of real storage.
LOGOFF	N/A	Any	None	Disable access to CP.
LOGON	N/A	Any	None	Provide access to CP.
MESSAGE	N/A	Any	None	Transmit messages to other users.
	<O>	A,B		Send message text to a specified user, to primary system operator, or to one or all logged-on users.
MIGRATE	<O>	A		Activate the normal page/swap table migration routines or force a particular user's pages to the secondary device even if that user is currently active.
MONITOR	<O>	A,E		Initiate or override the system-generated monitor function or terminate the recording of events occurring in the real machine.
MSGNOH	<R>	B		Allow a service virtual machine to send messages to specified users without the standard header associated with the MESSAGE command.

COMMAND	FUNCTION TYPE*	IBM-DEFINED PRIVILEGE CLASS	USER-DEFINED CLASS	Usage
NETWORK	O	A		Load, dump, and control operation of a 3704/3705 and control operation of a 3725 control program operating in 270x emulation mode (EP). Also control remote 3270 devices via binary synchronous lines.
	R	B		Load, dump, and control operation of a 3704/3705 and control operation of a 3725 control program operating in 270x emulation mode (EP).
NOTREADY	<G>	G		Simulate "not ready" for a device to a virtual machine.
ORDER	S	D		Place closed spool files (of a specified device type) in a different order.
	G	G		Rearrange closed spool files in a specific order.
PER	<G>	A,B,C,D, E,F,G		Monitors certain events in the user's virtual machine as they occur during program execution.
PURGE	S	D		Remove closed spool files from the system before they are printed or punched by the spooling devices or before they are read by a user.
	G	G		Remove closed spool file from the system.
QUERY	O	A		Provide status information on the real or virtual machine and miscellaneous CP functions. Also displays the status of MVS/System Extensions Support.
	R	B		Provide status information on the real or virtual machine and miscellaneous CP functions. Displays the status of the various devices.
	P	C		Provide system log messages and information about system users and processors.
	S	D		Provide system spooling information.
	A	E		Provide status information on the real or virtual machine and miscellaneous CP functions.
	C	F		Provide system log messages and information about system users.

COMMAND	FUNCTION TYPE*	IBM-DEFINED PRIVILEGE CLASS	USER-DEFINED CLASS	Usage
	G	G		Request information about machine configuration and system status.
QVM	<O>	A		Request the transition from VM/SP environment to native mode for a particular virtual machine.
READY	<G>	G		Simulate device end interruption for a virtual device.
REPEAT	<S>	D		Increase the number of copies of an output file or place the current output file in a HOLD status increasing or not increasing the number of copies to be created.
REQUEST	<G>	G		Make an attention interruption pending for the virtual machine console.
RESET	<G>	G		Clear and reset all pending interruptions for a specified virtual device and reset all error conditions.
REWIND	<G>	G		Rewind (to load point) a tape and ready a tape unit.
SAVESYS	<A>	E		Save a virtual machine storage space with registers and PSW as they currently exist. Used in the process of creating named systems.
SCREEN	<G>	G		Allows the user to change or alter the color and extended highlighting values for his virtual machine.
SEND	<G>	G		Pass commands and message replies to disconnected virtual machine for processing.
SET	O	A		Establish system parameters and perform various functions to control the CP system and virtual machine options.
	R	B		Change log message, designate the unit to receive system abend dump, change time interval for a specific device class, set off monitoring for a specified class, or terminate all monitoring of missing interruptions.
	A	E		Sets paging and sets the system resource management function.

COMMAND	FUNCTION TYPE*	IBM-DEFINED PRIVILEGE CLASS	USER-DEFINED CLASS	Usage
	C	F		Set the recording mode for a device and for soft errors.
	G	G		Control various functions within the virtual machine.
SHUTDOWN	<O>	A		Systematically end all virtual machine functions and checkpoint the system for an eventual warm start.
SPACE	<S>	D		Force the output on the specified printer to be single spaced for the current active spool file, regardless of the carriage control commands the actual file.
SPMODE	<O>	A		Establish or reset the single processor mode environment.
SPTAPE	<S>	D		Dump spool files to tape or load spool files from tape.
START	<S>	D		Restart a spooling device after it has been drained or to change the output class that it may service.
STCP	<P>	C		Alter the contents of real storage but not real PSW or real registers.
SMSG	<G>	G		Send special messages to specified virtual machine.
SPOOL	<G>	G		Alter spooling control options; direct a file to another virtual machine or to a remote location via the RSCS virtual machine.
STORE	<G>	G		Alter specified virtual storage locations and registers.
SYSTEM	<G>	G		Simulate RESET, CLEAR STORAGE, and RESTART buttons on a real system console.
TAG	<G>	G		Specify variable information to be associated with a spool file or output unit record device. Interrogate the current TAG text setting of a given spool file or output unit record device.
TERMINAL	<G>	G		Define or redefine the input and attention handling characteristics of your virtual console.

COMMAND	FUNCTION TYPE*	IBM-DEFINED PRIVILEGE CLASS	USER-DEFINED CLASS	Usage
TRACE	<G>	G		Trace specified virtual machine activity at your terminal, spooled printer, or both.
TRANSFER	S	D		Direct one or more spool files to a specified user or reclaim reader spool files previously sent to one or more users.
	G	G		Transfer input files or reclaim input files from a specified user's virtual card reader.
UNLOCK	<O>	A		Unlock page frames previously locked by a LOCK command.
VARY	<R>	B		Mark a device available or unavailable for use by a user or the control program.
VMDUMP	<G>	G		Dump virtual machine (use VM/SP IPCS to view dump.)
WARNING	<O>	A,B		Transmit high-priority messages to a specified user or to all users.

Index

A

- abbreviations of commands 3-3
- abend dump, printing from tape 4-40
- abend dump, system 2-37
- ACNT command 3-6
- addresses for stand-alone dump facility output devices 4-89
- affinity
 - automatic reset 1-6
 - usage with attached processor 2-34
- ALLOCATE control statement 4-29
 - control statement descriptions 4-32
 - example of operation 4-33
 - format 4-31
 - functions of 4-33
- allocating DASD space for CP use 4-31
- alternate console
 - at system generation 2-5
 - automatic logon 2-5
 - disconnecting 2-17
 - multiple 2-17
 - selecting (during initialization) 2-19
 - specifying 2-5
 - valid terminal designations 2-5, 2-17
- analyst function type 3-1
- ATTACH command 3-8
- attached processor system
 - affinity command usage 2-34
 - initialization 2-16
 - restarting 1-6
 - restrictions for locked shared pages 2-31
 - specifying one processor 2-33
- attaching and detaching volumes 2-3
- attributes, spool file 5-5
- AUTOLOG command 3-15
- automatic ipl 2-11
- automatic logon, operator 2-12
- automatic volume mount and demount
 - causes 2-14
 - 3330V 2-14

B

- BACKSPAC command 3-17
- BASE, ZAP control record 4-48
- batch facility
 - controlling 5-14
 - invoking 5-13
 - using 5-12
- batch jobs
 - purging 5-15

- reordering 5-15
- restarting 5-15
- batch virtual machine
 - starting 5-13
 - stopping 5-16
- block, FBA device 4-28
- braces
 - nested 3-5
 - to indicate choices 3-4
- brackets
 - nested 3-5
 - to indicate choices 3-4
- buffers, spool 5-5

C

- CANCEL key 2-2
- capacity of device when formatted 4-25
- card input
 - spooled 5-9
 - to Format/Allocate program 4-29
- card separator, spool punch output 5-10
- CATs 4-19, 4-20
- CCH (channel check handler) 1-4
- CE function type 3-1
- CHANGE command 3-20
- channel check handler (CCH) 1-4
- channel control check 1-4
- character arrangement tables 4-20
- character arrangement tables (CATs) 4-19
- character set - 3800 printer
 - modified 2-25
 - predefined 2-25
- checkpoint (CKPT) start 2-10
- CKPT (checkpoint) start 2-10
- class, privilege
 - See privilege class
- classes, output for spooling 5-6
- clock
 - setting uniprocessor application 2-8
 - time-of-day (TOD) 2-7, 3-169
- CMD command (PROP) 4-71
- CMS (Conversational Monitoring System) commands
 - EXPAND 4-54
 - GENIMAGE 4-19
 - IMAGELIB 4-21
 - IPCSDUMP 2-37
 - NCPDUMP 4-42
 - PRTDUMP 4-40
 - system operator 4-2
 - ZAP 4-43
 - ZAPTEXT 4-52
- cold start 2-11

commands

See also CMS (Conversational Monitoring System) commands

See also CP (Control Program) commands

See also programmable operator facility, commands

CP (Control Program)

command format 3-3

command privilege classes 3-1

commands 3-1

nesting 3-5

notational conventions 3-3

privilege class assignments 3-1

spooling 5-1

entering 2-1

function types 3-1

IMAGEMOD 4-21

privilege class 3-1

3704/3705/3725 control program 4-1

COMMENT, ZAP control record 4-51

Communications Control program

controlling 4-1

communications controller

See also 3704/3705 Communications Controller

dumping storage of 4-41

loading considerations 2-20, 4-1

configuring the stand-alone dump facility 4-92

considerations

for virtual=real performance option 2-28

spooling 5-1

console 2-1

alternate 2-17

description 2-17

selecting 2-19

definition 2-17

input 2-1

procedures 2-1

to Format/Allocate program 4-29

messages after VM/SP initialization 2-22

primary

description 2-17

disabling 2-19

spooling 5-3

system description 2-17

virtual 5-3

Control Program

See commands, CP (Control Program)

See CP (Control Program)

See CP (Control Program) commands

control records

BASE, ZAP program 4-45, 4-48

COMMENT, ZAP program 4-45, 4-51

DUMP, ZAP program 4-45

END, ZAP program 4-45, 4-52

EXPAND, ZAPTEXT program 4-54

LOG, ZAP program 4-45, 4-50

NAME, ZAP program 4-45, 4-47

REP, ZAP program 4-45, 4-49

VERIFY, ZAP program 4-45, 4-49

ZAP service program 4-43

ZAPTEXT 4-53

control statements

DDR program

See function control statements, DDR program

Format/Allocate program

See Format/Allocate program

controlling

the CMS batch facility 5-14

3704/3705 Communications Controller 2-20

3704/3705/3725 Communications Controller 4-1

conventions, environment 2-1

Conversational Monitoring System

See CMS (Conversational Monitoring System) commands

conversion to 3800 Model 3 pel density 2-25

COPY control statement, DDR program 4-8

copy modification for 3800 printer 2-25

copy modification modules (COPYMODs) 4-19

COPYMODs 4-19

count-key-data DASD

copy restriction 4-14

formatting cylinder volumes 4-25

label requirements 2-2

moving data with tape dump 4-5-4-14

record count area initializing 4-24

CP (Control Program)

CP assist 2-33

attached processor application 2-33

supported processors 2-33

disk areas 5-1

dispatcher 2-27

ID card

description of 5-9

fields 5-9

format 5-9

message to operator 5-11

paging routine 2-27

prefixing with 2-1

SYSRES volume 2-2

system functions from disk 2-2

using 2-1

CP (Control Program) commands

ACNT 3-6

ATTACH 3-8

AUTOLOG 3-15

BACKSPAC 3-17

CHANGE 3-20

CPTRAP 3-25

DCP 3-31

DEFINE 3-35

DETACH 3-37

DISABLE 3-41

DMCP 3-43

DRAIN 3-47

ENABLE 3-49

FLUSH 3-51

FORCE 3-54

FREE 3-56

HALT 3-57

HOLD 3-58
 INDICATE 3-61
 A privilege class 3-61
 E privilege class 3-62
 LOADBUD 3-71
 LOCATE 3-75
 LOCK 3-78
 MESSAGE 3-82
 MIGRATE 3-84
 MONITOR 3-86
 MSGNOH 3-100
 NETWORK 3-102
 A privilege class 3-104
 B privilege class 3-105
 ORDER 3-113
 PURGE 3-115
 QUERY 3-117
 A privilege class 3-118
 B privilege class 3-119
 C privilege class 3-119
 D privilege class 3-120
 E privilege class 3-121
 F privilege class 3-121
 QVM 3-151
 REPEAT 3-152
 SAVESYS 3-154
 SET 3-155
 A privilege class 3-155
 B privilege class 3-156
 E privilege class 3-156
 F privilege class 3-157
 SHUTDOWN 3-169
 SPACE 3-171
 SPMODE 3-172
 SPOOL (general user) 5-1
 SPTAPE 3-173
 START 3-179
 STCP 3-184
 TRANSFER 3-186
 UNLOCK 3-189
 VARY 3-191
 WARNING 3-195
 CPEREP program, I/O error recording 1-5
 CPTRAP command 3-25
 TRAPRED 3-30
 type X'3D' record 3-25
 type X'3E' record 3-25
 type X'3F' record 3-26
 cylinders
 format for CP use 4-24
 format of cylinder zero 4-26

D

DASD (direct access storage device)
 See also disks
 for spool buffering 5-5
 space allocating on CP owned volumes 4-31
 volumes
 formatting 4-24
 initializing 4-24
 labeling 4-34
 DASD Dump Restore (DDR) program 4-2
 control statements
 INPUT 4-5
 OUTPUT 4-5
 SYSPRINT 4-7
 DDR command 4-3
 function control statements 4-8
 COPY 4-8
 DUMP 4-8
 PRINT 4-14
 RESTORE 4-8
 TYPE 4-14
 functions of 4-2
 I/O definition statements 4-5
 invoking
 as a standalone program 4-4
 under CMS 4-3
 DASD format 4-98
 DCP command 3-31
 DDR
 See DASD Dump Restore (DDR) program
 DDR restrictions 4-3, 4-12
 debug mode, determining problems with 4-87
 dedicated device, 3800 printer subsystem support 2-25
 defaults, usage in underscored notation 3-5
 DEFINE command 3-35
 defining the console 2-17
 DETACH command 3-37
 detaching and attaching volumes 2-3
 device support facilities 4-23
 devices for stand-alone dump 4-89
 devices for stand-alone dump facility
 for dump output 4-89
 to IPL 4-89
 devices, FBA 4-28
 diagnosing software failure 2-38
 direct access storage device
 See DASD (direct access storage device)
 DISABLE command 3-41
 disabling the primary VM/SP console 2-19
 disks
 allocating space for CP use 4-24
 for spool buffering 5-5
 labeling 2-2
 volumes
 formatting 4-24
 labeling 4-34
 DMCP command 3-43

DMKDDR
 See DASD Dump Restore (DDR) program
 DMKDIR (directory program) 4-33
 DMKFMT
 See Format/Allocate program
 DMKMSS (mass storage system) error messages 2-15
 DOS virtual=real storage assignments 2-28
 DRAIN command 3-47
 dump
 format 4-96
 from 3704/3705 4-42
 processing 4-42
 function control statement, DDR program 4-8
 NETWORK command usage 4-41
 operand
 NETWORK command 4-41
 SET command 3-158
 printing from tape 4-40
 printing under CMS 4-40
 system abend 2-37
 ZAP control record 4-45
 DUMP Control Record, ZAP program 4-45
 Dump Restore service program, DASD 4-2
 dynamic trace facility 4-1

E

ECPS (Extended Control-Program
 Support:VM/370) 2-32
 attached processor application 2-33
 supported processors 2-33
 edit error records 1-5
 editor, system product 4-2
 elimination, queue drop 2-35
 Emulation Program (EP)
 See also 3704/3705 Communications Controller
 special considerations for loading 2-21
 ENABLE command 3-49
 END, ZAP control record 4-52
 entering CP commands 2-2
 environment conventions 2-1
 EP (Emulation Program)
 See 3704/3705 Communications Controller
 epoch clock value 2-9
 EREP with the CPEREPE command 1-5
 error conditions 4-99
 error handling, I/O 5-4
 error messages
 See messages
 error recording
 CPEREPE program 1-5
 I/O 1-4
 IOERBLOK, recording in 1-4
 error records, editing 1-5
 examples
 configuring the stand-alone dump facility 4-93
 programmable operator 4-62

in a distributed environment 4-64
 in a local environment 4-62
 in a mixed environment 4-65
 execution, favored 2-27
 EXPAND command 4-54
 EXPAND control record 4-54
 Extended Control-Program Support
 See ECPS (Extended Control-Program
 Support:VM/370)
 extended virtual machine assist 2-33

F

facilities
 repair 1-5
 restart 1-6
 failure, recovery from 2-9
 favored execution option 2-27
 FB-512, format for 4-28
 FBA DASD
 block format 4-28
 copy restriction 4-10
 device capacity 4-26
 format 4-28
 Format/Allocate program
 using allocate function 4-31
 using allocate overlap function 4-37
 using label function 4-34
 general description 4-25
 initialization 4-24
 label requirements 2-2
 page formatting/allocating 4-24
 tape dump, movement of data 4-9
 FCBs 4-19
 features
 defining for a 3330V 2-14
 sysvirt 2-14
 virtual 2-14
 FEEDBACK command (PROP) 4-73
 filename attribute, spool files 5-5
 filetype attribute, spool files 5-5
 fixed-block DASD
 See FBA DASD
 fixed-block mode
 See FBA DASD
 FLUSH command 3-51
 font offset buffer (FOB)
 loading print belt image 3-71
 usage with 3289 model 4 printer 3-71
 FORCE command 3-54
 force start 2-10
 format
 DASD 4-98
 of cylinder zero 4-26
 of cylinders for use by CP 4-24
 of user identification card 5-9
 printer 4-99

- tape 4-96
- FORMAT control statement, Format/Allocate program 4-29
- Format/Allocate program
 - ALLOCATE control statement 4-31
 - card input 4-29
 - console input 4-35
 - control statements 4-29
 - cylinder format 4-26
 - cylinder zero format 4-26
 - description of 4-24
 - example of program execution 4-36
 - FORMAT control statement 4-29
 - LABEL control statement 4-34
 - overlap function for FBA devices 4-36
- formatted device capacity 4-25
- formatting
 - count-key-data DASD 4-25
 - FBA DASD 4-26
 - volumes, general information 4-24
- forms control buffers (FCBs) 4-19
- forms control, description for 3800 printer 2-24
- forms overlay, description for 3800 printer 2-25
- FREE command 3-56
- function control statements
 - DDR program
 - COPY 4-8, 4-10
 - description 4-2
 - DUMP 4-8
 - INPUT 4-5
 - OUTPUT 4-5
 - PRINT 4-14
 - RESTORE 4-8
 - SYSPRINT 4-7
 - TYPE 4-14
- function types, CP command 3-1

G

- general function type 3-1
- generating the stand-alone dump program 4-91
- GENIMAGE service program 4-19
 - character sets 4-20
 - control statement format 4-19
 - description 4-19
 - output 4-20
 - responses 4-20
- GET command (PROP) 4-74
- graphic character modification modules (GRAPHMODs) 4-19
- GRAPHMODs 4-19

H

- HALT command 3-57
- handling I/O errors 5-4
- hardware assist
 - See also ECPS (Extended Control-Program Support:VM/370)
 - description 2-32
- hardware status messages 2-22
- HOLD command 3-58

I

- I/O (input/output)
 - definition statements, DDR program 4-5
 - error handling 5-4
- IBCDASDI 4-23
- IMAGE library 4-19
- IMAGELIB service program 4-21
 - command format 4-21
 - responses 4-20
- IMAGEMOD command
 - format and description 4-21
 - modifying 3800 named system 4-21
- INDICATE command 3-61
 - A privilege class 3-61
 - E privilege class 3-62
- INITDISK 4-23
- initialization
 - attached processor 2-16
 - disk 4-24
 - FBA DASD 4-24
 - mass storage system (MSS) 2-14
 - messages 2-22
 - minidisk 4-24
 - multiprocessor system 2-16
 - selecting an alternate console during 2-19
 - system 2-3
 - logging on the system operator 2-6
 - operator's console, locating 2-5
 - simplifying 2-4
 - starting 2-4
 - time-of-day clock 2-5
 - type of start, selecting 2-6
- initialization procedures
 - checkpoint start 2-10
 - clock, setting 2-8
 - cold start 2-11
 - force start 2-10
 - lines and resources, enabling 2-13
 - logon
 - automatic 2-12
 - manual 2-12
 - shutdown start 2-11
 - time-of-day clock 2-7
 - warm start 2-9

- initiating VM/SP operation 2-1
- input
 - card to Format/Allocate program 4-29
 - console to Format/Allocate program 4-35
 - control records, ZAP 4-45
 - control records, ZAPTEXT 4-53
 - spooled card 5-9
- INPUT control statement, DDR program 4-5
- input/output
 - See I/O (input/output)
- Interactive Problem Control System 2-37
- interface control check 1-4
- interval timer 2-17
 - at initialization 2-17
- interval timer assist, part of ECPS 2-33
- invoking
 - CMS batch facility 5-13
 - DDR
 - as a standalone program 4-4
 - under CMS 4-3
- IPCS (Interactive Problem Control System) 2-37
 - IPCS IPCSDUMP command 2-37
- IPCSDUMP command (IPCS) 2-37
 - format 4-45
- IPL format 4-96
 - DASD format 4-98
 - printer format 4-99
 - tape format 4-96
- IPL of stand-alone dump facility 4-88
 - configurations 4-88
 - devices for 4-89
- IPL, automatic 2-11

K

- keywords
 - See CMS (Conversational Monitoring System) commands
 - See CP (Control Program) commands

L

- LABEL control statement, Format/Allocate program 4-34
- label requirements for FB-512 DASD 2-2
- labeling
 - DASD volumes 4-34
 - disks 2-2
- LCSs 4-19
- library character set (LCS) 4-19
- library, IMAGE 4-19
- LOADBUF command 3-71
- loading
 - 3704/3705 Communications Controller 2-20

- 3704/3705 control program 2-13, 2-21
 - EP considerations 2-21
- 3704/3705/3725 control program 4-1
- LOADTBL command (PROP) 4-77
- LOCATE command 3-75
- LOCK command 3-78
- locked pages option 2-31
- locked shared pages, attached processor restriction 2-31
- LOG command (PROP) 4-79
- log file, programmable operator
 - description 4-67
 - ensuring completeness 4-68
 - getting the file 4-68
 - name 4-68
 - routing messages 4-69
- LOG, ZAP control record 4-50
- logon
 - automatic 2-12
 - manual 2-12
 - message 2-12
- lowercase letters, words and symbols 3-4

M

- machine check handler (MCH) 1-3
- manual logon, operator 2-12
- map storage allocations 2-29
- mass storage control (MSC), communicating with 2-15
- mass storage system
 - See MSS (mass storage system)
- MCH 1-3
- mdisk 4-28
- MESSAGE command 3-82
- messages
 - after startup 2-22
 - DMKMSS (mass storage system) 2-15
 - during VM/SP initialization 2-22
 - from virtual machine users 2-22
 - hardware and program status 2-22
 - logon 2-12
 - status to system operator 2-22
 - to system operator, SVC 76 2-22
- MIGRATE command 3-84
- minidisks initializing 4-24
- missing interrupt handler (MIH) 1-4
 - description 1-4
- modifying 3800 named system 4-21
- MONITOR command 3-86
- MSC (mass storage control), communicating with 2-15
- MSGNOH command 3-100
- MSS (mass storage system)
 - CP automatic volume mount and demount 2-14
 - error messages 2-15
 - initializing 2-14
 - shutdown process 3-169
 - specifying a partition 2-14
 - multiple alternate consoles selection 2-17

multiprocessor mode, recovery management 1-5
multiprocessor system
 initialization 2-16
 restarting 1-6

N

NAME, ZAP control record 4-47
NCCF interface with Programmable Operator Facility
 in a single system 4-59
 Message Exchange (PMX) 4-58
 nodeid 4-67
 operator 4-58
 operator station 4-62
 when stopping programmable operator 4-87
NCPDUMP
 command format 4-42
 program 4-43
 usage 4-43
nested commands 3-5
NETWORK DUMP during CP abend dumps 4-40
NETWORK command 3-102
 A privilege class 3-104
 B privilege class 3-105
 for remote 3270/3704/3705/3725 3-102
nolog password 5-9
notational conventions 3-3
number of copies of an output spool file 5-5

O

object deck form 4-54
operating procedures 2-2
operating the system
 installation requirements 1-1
 program requirements 1-1
operation, VM/SP initiating 2-1
operations function type 3-1
operator
 CP commands 3-3
 duties 1-1
 NCCF 4-58
 NCCF station 4-62
 responsibilities for spooling 5-8
 spooling functions 5-2
 system 1-1
 tasks
 disks, labeling 2-2
 system, initializing 2-3
 virtual devices, defining 2-3
 training 1-1
options, performance
 assignment of 2-3
 favored execution 2-27
 locked pages 2-31

 priority 2-31
 reserved page frames 2-27
 virtual=real 2-28
 VM/SP 2-26
ORDER command 3-113
output
 spool files 5-6, 5-10
 classes 5-6
 rearranging priorities 5-6
OUTPUT control statement, DDR program 4-5
output devices for stand-alone dump facility 4-88, 4-89
output forms 5-12
overlap function, FORMAT/ALLOCATE
 program 4-36
override, ALLOCATE control statement 4-31, 4-32

P

page
 allocating for FBA DASD 4-25
 recording for FBA DASD 4-25
partition in OS/VS1 system 2-15
password
 nolog 5-9
 suppression 3-15
performance options
 affinity 2-34
 assignment 2-3
 Extended Control-Program Support 2-32
 favored execution 2-27
 hardware assist 2-32
 locked pages 2-31
 priority 2-31
 reserved page frames 2-27
 virtual machine assist 2-32
 virtual=real 2-28
PMX, stopping 4-87
primary system operator
 See also operator
 description of 1-1
primary VM/SP console, disabling 2-19
PRINT function control statement
 DDR program 4-14
 DDR sample output 4-18
printer separator for spool printer output 5-10
printing
 CMS dump file 4-40
 CP abend dump file from tape 4-40
 subsystem support, 3800 2-24
priority, performance option 2-31
privilege class
 CP command 3-1
problem determination in programmable operator
 facility 4-87
procedures, operating 2-2
processing the stand-alone dump facility data on
 tape 4-96

- processing 3704/3705 spool dump files 4-42
- program interruption 2-37
- program status messages 2-22
- programmable operator facility
 - commands
 - CMD 4-71
 - FEEDBACK 4-73
 - GET 4-74
 - LGLOPR 4-75
 - LOADTBL 4-77
 - LOG 4-79
 - QUERY 4-80
 - SET 4-83
 - STOP 4-86
 - description of 4-56
 - environments
 - distributed VM 4-57, 4-63
 - mixed 4-58, 4-65
 - single system 4-57, 4-62
 - examples of communicating
 - distributed environment 4-64
 - local environment 4-62
 - mixed environment 4-65
 - installing 4-59
 - invoking 4-59
 - issuing commands with PROP 4-62-4-66
 - log file 4-67
 - message format
 - distributed 4-61
 - local 4-61
 - NCCF operator as the logical operator 4-58
 - problem determination 4-87
 - PROPST exec 4-59
 - sample configuration 4-58
 - shortcuts 4-67
 - single system with NCCF 4-59
 - stopping 4-86
 - tasks 4-56
- programmer function type 3-1
- PROPST exec to invoke PROP 4-59
- PURGE command (CP) 3-115
- purging the batch facility 5-15

Q

- QUERY command 3-117
 - A privilege class 3-118
 - B privilege class 3-119
 - C privilege class 3-119
 - D privilege class 3-120
 - E privilege class 3-121
 - F privilege class 3-121

- QUERY command (PROP) 4-80
- queue drop elimination 2-35
- QVM command 3-151

R

- real storage, dumping with stand-alone dump 4-88
- recording I/O errors with CPEREP program 1-5
- recording mode (MCH) 1-3
- recovery features 1-5
- recovery from VM/SP system failures 2-9
- recovery management in multiprocessor mode 1-5
- recovery management support
 - See RMS (recovery management support)
- remote spooling communications subsystem 5-3
 - spooling to remote locations 5-3
- reordering the batch facility 5-15
- REP, ZAP control record 4-49
- repair facilities 1-6
- REPEAT command 3-152
- reporting software failure 2-38
- reserved page frames option 2-27
- resid 3-103
- resident pages dumped with stand-alone dump
 - program 4-88
- resource function type 3-1
- resource identification, remote 3270 devices and control
 - units 3-103
- responsibilities for spooling operator 5-8
- restart facilities
 - in attached processor operations 1-6
 - VM/SP 1-6
- restarting the batch facility 5-15
- RESTORE control statement, DDR program 4-8
- restrictions
 - count-key-data devices 4-12
 - FBA DASD 4-12
 - for using DDR 4-3
 - locked pages 2-31
 - virtual=real performance option 2-28
 - VM/370 hardware assist 2-34
 - 3066/3215/3270 terminal console (CANCEL key) 2-2
- RMS (recovery management support)
 - channel check handler (CCH) 1-4
 - machine check handler (MCH) 1-2
 - missing interrupt handler (MIH) 1-4
 - System/370 1-2
 - VM/SP 1-2
- RSCS (Remote Spooling Communications Subsystem) 5-3
 - spooling to remote locations 5-3

S

- SADGEN ASSEMBLE file 4-92
- SADUMP example 4-93
- SADUMP EXEC 4-91, 4-92
- saved virtual storage 1-6
- SAVESYS command 3-154
- selecting an alternate VM/SP console 2-19
- separator
 - spool printer 5-10
 - spool punch output 5-10
- service function type 3-1
- service programs 4-1
 - DASD Dump Restore (DDR) 4-2
 - Format/Allocate 4-24
 - NCPDUMP 4-42
 - ZAP 4-43
- service virtual machine 2-35
- SET command 3-155
 - A privilege class 3-155
 - B privilege class 3-156
 - CPASSIST use with ECPS 2-33
 - DUMP command 2-37
 - E privilege class 3-156
 - F privilege class 3-157
 - FAVORED command with favored execution 2-28
 - MODE MAIN, invalid processors for 1-3
 - NOTRANS subcommand 2-29
 - QDROP subcommand 2-35
 - SASSIST subcommand
 - use with ECPS 2-33
 - use with hardware assist 2-32
 - use with virtual machine assist 2-32
 - use with ECPS 2-33
- SET command (PROP) 4-83
- shared pages in attached processor system 2-31
- sharing devices, spooling 5-1
- SHUTDOWN command 3-169
- SHUTDOWN command to terminate system 2-35
- shutdown start 2-11
- single processor mode 2-24
- software failure 2-38
- SPACE command 3-171
- SPMODE command 3-172
- spool buffers 5-5
- SPOOL command to spool virtual console output 5-4
- spool file
 - attributes
 - date 5-5
 - filename 5-5
 - filetype 5-5
 - number of copies 5-5
 - number of records 5-5
 - spoolid 5-5
 - time 5-5
 - userid 5-5
 - VM/SP unit record 2-13
- spooled
 - card input 5-9
 - input handling 2-3
 - output handling 2-3
 - printer output 5-7
 - punch output 5-7
- spooling
 - across a teleprocessing network 5-3
 - commands
 - considerations 5-1
 - devices 5-2
 - I/O error handling 5-4
 - input spool files 5-3
 - operator functions 5-2
 - operator responsibilities 5-8
 - options 5-2
 - output spool files 5-3
 - real 5-2
 - support 5-2
 - virtual console 5-3
 - with RSCS 5-3
- spooling function type 3-1
- SPTAPE command 3-173
- stand-alone dump facility 4-88
 - configuring 4-92
 - customizing 4-91
 - description 4-88
 - error recovery 4-99
 - example for configuring 4-93
 - formats
 - DASD 4-98
 - dump 4-96
 - IPL 4-96
 - printer 4-99
 - IPL devices 4-89
 - IPL of 4-88
 - non-labeled tape used 4-91
 - output 4-88
 - output devices 4-89
 - processing dump data on tape 4-96
 - program generation 4-91
 - programs 4-88
 - single-volume tape used 4-91
 - taking a dump 4-94
 - using 4-92
- start
 - checkpoint (CKPT) 2-10
 - cold 2-11
 - force 2-10
 - shutdown 2-11
 - warm start to initiate VM/SP) 2-9
- START command 3-179
- starting VM/SP operation 2-1
- status messages
 - hardware and program 2-22
 - system operation 2-22
- STCP command 3-184
- STOP command (PROP) 4-86
- stopping the batch virtual machine 5-16
- stopping the programmable operator 4-86

- storage allocations, map 2-29
- support facilities, device 4-23
- suppression of passwords 3-15
- surface analysis utility 4-23
- SVC 76
 - error recording 1-4
 - messages to system operator 2-22
- SYSCOR macro instruction with AP operand 2-16
- SYSOPT macro to specify virtual card reader file 2-36
- SYSPRINT control statement, DDR program 4-7
- system
 - abend dumps 2-37
 - console 2-17
 - definition 2-3
 - failure dump unit 2-36
 - initialization 2-3
 - messages during initialization 2-22
 - termination 2-35
- system operation
 - installation requirements 1-1
 - program requirements 1-1
- system operator 1-1, 1-2
 - See also operator
 - alerted to error conditions 1-2
 - description of 1-1
 - functions performed by 1-2
- system product editor 4-2

T

- taking a stand-alone dump 4-94
- tape format 4-96
- tape used for stand-alone dump facility output 4-91
- terminals
 - for VM/SP control 2-17
 - input procedures 2-2
- termination
 - system 2-35
 - virtual machine 1-6
- text deck names from GENIMAGE command 4-21
- time slicing supported by the interval timer 2-17
- time-of-day (TOD) clock
 - during IPL 3-169
 - obtaining current value 2-7
 - setting 2-11
 - System/370 2-7
- TOD clock
 - See time-of-day (TOD) clock
- TRANSFER command 3-186
- translate tables, description for 3800 printer 2-25
- TRAPRED command to collect data 3-30
- truncation of commands 3-3
- TYPE function control statement
 - DDR program 4-14
 - DDR sample output 4-18
- type of entries that CPTRAP collects 3-25

U

- underscore, usage in notation 3-5
- unit record spool files 2-13
- universal character set feature 5-3
- UNLOCK command 3-189
- UNLOCK command with VIRT=REAL option 2-29
- uppercase letters and words 3-5
- user identification (userid), spool file attribute 5-5
- user identification card
 - for spooled card input 5-8
 - format 5-9
- user messages to the operator (virtual machine) 2-22
- userid, spool file attribute 5-5
- using CMS to alter VM/SP files 4-2
- using service programs
 - DDR program 4-2
 - Format/Allocate program (DMKFMT) 4-24
 - IPCSDUMP service program 4-41
 - NCPDUMP program 4-42
- using the stand-alone dump facility 4-92
- utility programs 4-1

V

- VARY command 3-191
- VERIFY, ZAP control record 4-49
- virtual console spooling 5-3
- virtual devices, defining 2-3
- virtual interval timer assist 2-33
- virtual machine
 - assist 2-32
 - console 5-3
 - restoring of contents 1-6
 - spooling
 - considerations 5-1
 - functions 5-2
 - stand-alone dump facility 4-88
 - termination 1-6
 - user messages 2-22
- Virtual Machine/System Product
 - See VM/SP (Virtual Machine/System Product)
- virtual storage preservation 1-6
- virtual=real
 - performance option 2-28
 - storage assignments in DOS system 2-29
- VM/SP (Virtual Machine/System Product)
 - components, brief description 1-1
 - concepts, brief description referral 1-1
 - initiating 2-1
 - messages, initialization and status 2-22
 - operating procedures 2-2
 - operator

- See also operator
 - alerted to error conditions 1-2
 - functions performed by 1-2
- performance options 2-26
- recovery features 1-5
- recovery from system failures 2-9
- recovery management support 1-2
- repair facilities 1-5
- restart facilities 1-6
- service programs 4-2
 - See also service programs
- start
 - checkpoint (CKPT) start 2-10
 - cold start 2-11
 - force start 2-10
 - shutdown start 2-11
 - warm start 2-9
- system console 2-17
- system operator 2-9
- unit record spool files 2-13

VM/370

- Extended Control-Program Support 2-32
- hardware assist 2-32
- hardware assist restrictions 2-34

VMSAVE function 1-7

volume mount and demount, 3330V 2-14

volumes

- format of cylinder zero 4-26
- format of cylinders for CP use 4-28
- use of Format/Allocate program 4-24
 - allocating 4-31
 - formatting 4-29
 - labeling 4-34
- user and system, attaching and detaching 2-3

W

warm start 2-9
 WARNING command 3-195

X

XEDIT 4-2

Z

ZAP

- control records
 - BASE 4-48
 - COMMENT 4-51
 - DUMP 4-45
 - END 4-52

- LOG 4-50
- NAME 4-47
- REP 4-49
- VERIFY 4-49
- option output 4-45
- service program
 - description 4-43
 - special considerations for 4-52

ZAPTEXT

- description 4-52
- input control records 4-53

Numerics

- 3031, set mode main invalid 1-3
- 3032, set mode main invalid 1-3
- 3033, set mode main invalid 1-3
- 3066/3215/3270 terminal console, CANCEL key restrictions 2-2
- 3081, set mode main invalid 1-3
- 3270
 - primary/alternate console 2-5, 2-17, 2-19
 - remote system 3-103
- 3289 model 4 printer, loading print belt image 3-72
- 3310 DASD
 - See also FBA DASD
 - device capacity 4-26
- 3330V
 - automatic volume mount and demount 2-14
 - features 2-14
 - sysvirt feature 2-14
 - virtual feature 2-14
- 3370 DASD
 - See also FBA DASD
 - device capacity 4-26
 - general description 4-25
- 3480 magnetic tape subsystem
 - restrictions in single processor 2-24
 - restrictions under DDR 4-4
 - restrictions under VM/SP 2-24
- 3704/3705 Communications Controller
 - CMS commands 4-1
 - controlling 2-20, 4-1
 - dumping operations 4-41
 - Emulation Program (EP), loading considerations 2-21
 - loading considerations 2-21
 - NCPDUMP program usage 4-42
 - network load command usage 2-20
 - processing spool dump files 4-42
- 3800 printer
 - character arrangement tables for Model 1 and 3 4-20
 - character arrangement tables for Model 3 only 4-20
 - spool files 3-138
- 3800 printing subsystem
 - dedicated device support 2-24

features 2-24
translate tables 2-25
3800, modifying named systems 4-21
3850 mass storage system (MSS) 2-14

error messages 2-15
installation 2-15
partitioning 2-15
shutdown process 3-169



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? Yes No

- Did you find the material:
 - Easy to read and understand? Yes No
 - Organized for convenient use? Yes No
 - Complete? Yes No
 - Well illustrated? Yes No
 - Written for your technical level? Yes No

- What is your occupation? _____

- How do you use this publication:
 - As an introduction to the subject? Yes No As an instructor in class? Yes No
 - For advanced knowledge of the subject? Yes No As a student in class? Yes No
 - To learn about operating procedures? Yes No As a reference manual? Yes No

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.

Reader's Comment Form

Cut or Fold Along Line

Fold and Tape

Please Do Not Staple

Fold and Tape



NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL
FIRST CLASS PERMIT NO. 40 ARMONK, N.Y.



POSTAGE WILL BE PAID BY ADDRESSEE:

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

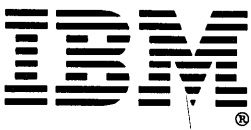
Fold

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 _____

VM/SP Operator's Guide (File No. S370/4300-40) Printed in U.S.A. SC19-6202-3



SC19-6202-3

IBM
®

SC19-6202-3

