

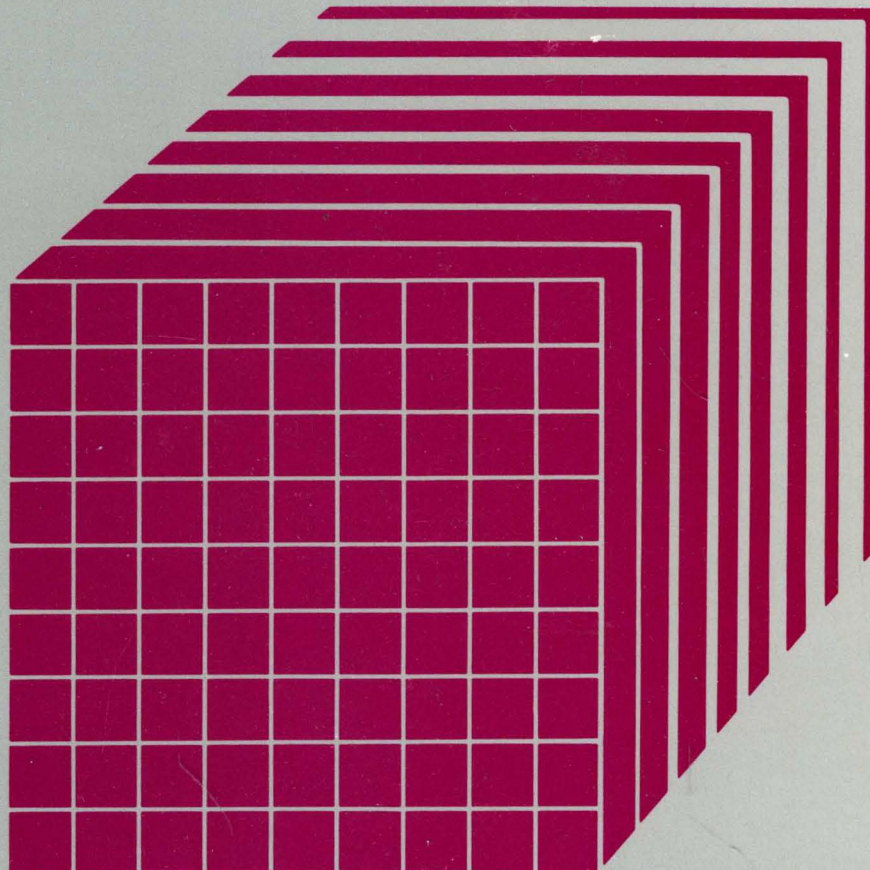


Virtual Machine/
System Product
High Performance Option

**CP Command Reference
for General Users**

Release 4.0

SC19-6227-5



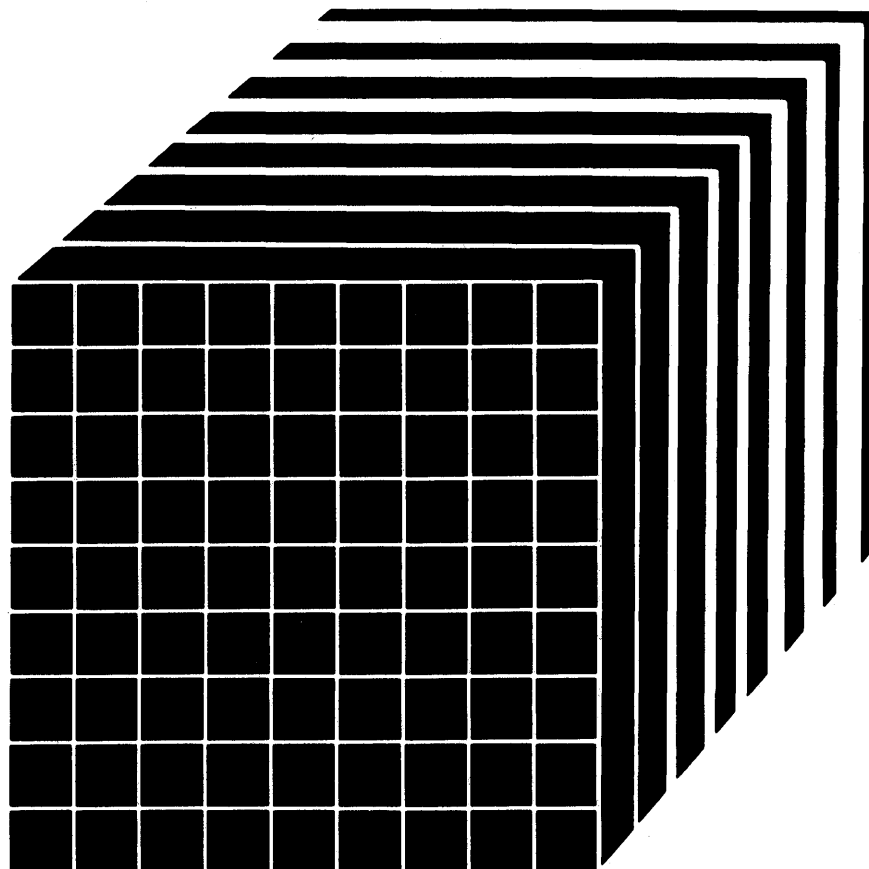


Virtual Machine/
System Product
High Performance Option

**CP Command Reference
for General Users**

Release 4.0

SC19-6227-5



The term "VM/SP High Performance Option" applies to the VM/SP High Performance Option Program Product when used in conjunction with VM/System Product Program Product.

Sixth Edition (November 1985)

This is a major revision of SC19-6227-3. See the Summary of Changes on page iii for the changes made to this manual. Technical changes or additions to the text or illustrations are indicated by a vertical line to the left of the change.

The sixth current edition applies to Release 4 of IBM Virtual Machine/System Product High Performance Option Program Product (Program Number 5664-173), and to later releases and modifications until otherwise indicated in new editions or Technical Newsletters. The sixth edition is a revision of the fourth edition (Release 3.4) and is not a revision of the fifth edition (Release 3.6). The fifth edition (SC19-6227-4) applies only to Release 3.6 of IBM Virtual Machine/System Product High Performance Option.

To order previous editions that still apply to Releases 3.4 and 3.6, use the following temporary order numbers:

Release 3.2 Third Edition ST00-1442

Release 3.4 Fourth Edition ST00-1560

Release 3.6 Fifth Edition ST00-1620

Changes are made periodically to the information herein; before using this publication in connection with the operation of IBM systems, refer to the latest *IBM System/370 and 4300 Processors Bibliography*, GC20-0001, for the editions that are applicable and current.

References in this publication to IBM products, programs, or services do not imply that IBM intends to make these available in all countries in which IBM operates. 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.

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, Department 52Q, Neighborhood Road, Kingston, N.Y. 12401. IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you.

Summary of Changes

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

Release 3.2, order ST00-1442

Release 3.4, order ST00-1560

Release 3.6, order ST00-1620

Summary of Changes for SC19-6227-5 as updated November 22, 1985 for VM/SP HPO Release 4.0

Note: Release 4 does not support 3090 processors. These processors are supported by Release 3.6.

3880 MODELS 13 AND 23 SUPPORT

New: Hardware Support

VM/SP HPO now supports the 3880 Storage Subsystem Models 13 and 23. This support is available after HPO Release 4.0 general availability via program update tape.

The 3880 Models 13 and 23 are high performance cached DASD subsystems designed especially for non-paging applications (application data that resides on DASD that is not defined as paging, swapping, spooling, or dump area). This support is provided to improve performance for MVS or CMS guest virtual machines running with 3880 DASDs. The performance improvement is accomplished principally by maximizing the number of read accesses that can be satisfied by accessing the cache copy.

This support includes:

- Operator commands to enable/disable the caching function so that the cache may be taken off-line while support personnel work on it.
- Recognition of 3880 Model 13 and 23 hardware errors.
- Monitor records that depict cache use from both storage directors.
- Dedicated guest virtual machine control of the cache.

PREFERRED MACHINE ASSIST EXTENSIONS: CONTROL SWITCH ASSIST

New: Programming Support

Control Switch Assist consists of extensions to Preferred Machine Assist. This support is available at HPO Release 4.0 general availability via program update tape.

This support provides the performance benefits of Preferred Machine Assist and selected CP DIAGNOSE support available for virtual machines. In addition, this support provides assistance to users experiencing line timeout problems on Preferred Machine Assist guest-owned devices attached to CP-owned channels.

GROUP CONTROL SYSTEM (VM/SP HPO GCS)

New: Programming Support

This new component of VM/SP HPO is a virtual machine supervisor that provides simulated MVS services and supports a multitasking environment.

VM SNA Console Services (VSCS) graphics device users use the DIAL command to access logged-on guest virtual machines.

The GRAF and CONSOLE operands of the QUERY command now display SNA device information.

EXPANSION OF USER CLASSES

New: Programming Support

The COMMANDS command has been added so that the system can send you a list of commands and diagnose codes you are authorized to use.

VM/SP 3800 MODEL 3 COMPATIBILITY

New: Hardware Support

Compatibility support allows VM/SP HPO users to access 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 compared with the 180 X 144 PEL resolution of the 3800 Model 1) and the addition of a 10 lines-per-inch (LPI) vertical space option.

3880 STORAGE SUBSYSTEM MODEL 21 SUPPORT

New: Hardware Support

VM/SP HPO now supports the 3880 Storage Subsystem Model 21. The 3880 Model 21 is designed as a high-performance paging and swapping subsystem. It is an enhancement over the 3880 Storage Subsystem Model 11. In addition, the 3880 Storage Subsystem Model 11 is supported for swapping.

MISCELLANEOUS

Changed: Documentation Only

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

ATTACHMENT OF MORE THAN 410 DEVICES

New: Programming Support

This support provides a means of allowing selected virtual machines to attach more than 410 devices.

Summary of Changes for SC19-6227-3 as Updated July 31, 1984 for VM/SP HPO Release 3.4

VM/SP HPO Release 3.4 incorporates the features from all previous VM/HPO releases and provides the functional extensions of VM/SP Release 3.

ENHANCED PAGING SUBSYSTEM

New: Programming Support

Enhanced paging subsystem support, also referred to as block paging, is a new software method designed to improve paging. Enhanced paging subsystem support groups user pages together into swap sets and moves these swap sets in and out of storage as a group. This is designed to improve response time for interactive users.

The 4248 Model 1 printer is a variable high-speed impact printer using print band technology. It features speeds of 2200, 3000, and 3600 lines per minute; the ability to produce a duplicate copy of a printed line to the right of the original line, control stacker levels, and control stacker rates; and produces new sense data that describes operating system recovery actions.

**Summary of Changes
for SC19-6227-2
as Updated July 31, 1984
for VM/SP HPO Release 3.2**

PER COMMAND

New: Programming Support

The new PER command extends problem determination capability. You can use the PER command to trace all instructions, all successful branches, all register alterations, and all instructions executed in your virtual machine that alter storage.

3088 MULTISYSTEM COMMUNICATIONS UNIT

New: Hardware Support

The 3088 Multisystem Communications Unit interconnects multiple systems that use block multiplexer channels. The 3088 uses an unshared subchannel for each unique address and is compatible with existing channel adapter protocol.

The RSCS and IPCS components of VM/370 remain unmodified by the VM/SP HPO program product. If the general user intends to use these two unmodified VM/370 components, he should refer to the following publications for information on the RSCS and IPCS command languages:

- *IBM Virtual Machine Facility/370 Remote Spooling Communications Subsystem (RSCS) User's Guide*, Order No. GC20-1816.
- *IBM Virtual Machine/System Product Interactive Problem Control System (IPCS) Guide*, Order No. ST24-5260.

In addition, VM/SP HPO users can have enhanced networking facilities provided in the RSCS Networking program product (5748-XP1). The following publications contain information about the command languages of the RSCS Networking program product and the VM/IPCS Extension program product:

- *VM/370 RSCS Networking Program Reference and Operations Guide*, Order No. SH24-5005.

If the Group Control System is used, the following publications are suggested:

- *Virtual Machine/System Product Group Control System Guide*, ST24-5220.
- *Virtual Machine/System Product Group Control System Macro Reference*, ST24-5250.

References in text to titles of prerequisite and corequisite VM/SP High Performance Option publications are given in abbreviated form.

VM/SP HPO Library

To understand the interrelationships of the publications in the VM/SP HPO library, see Figure 1.

The VM/SP HPO Library

Evaluation

VM/SP
Introduction
GT00-1575

Index

VM/SP HPO
Library
Guide,
Glossary, and
Master Index
GC23-0187

Planning

VM/SP HPO
Planning
Guide and
Reference
SC19-6223

Virtual
Machine
Running
Guest
Operating
Systems
GT00-1586

VM/SP
Distributed
Data
Processing
Guide
SQ24-5241

VM/SP HPO
Release 4.0
Guide
SC23-0189

Input/Output
Configuration
Program
User's
Guide and
Reference
GC28-1027

Installation

VM/SP HPO
Installation
Guide
SC38-0107

VM/SP HPO
System
Definition
Files
SC23-0186

Administration

VM/SP HPO
System
Programmer's
Guide
SC19-6224

VM/SP
GCS
Guide
ST24-5249

VM/SP
GCS
Macro
Reference
ST24-5250

Operation

VM/SP HPO
Operator's
Guide
SC19-6225

Applications

VM/SP
Application
Developr.ent
Guide
ST24-5247

End Use

VM/SP
Terminal
Reference
GT00-1581

VM/SP
CMS
Primer
ST00-1591

VM/SP
CMS
Primer
For Line-
Oriented
Terminals
ST24-5242

VM/SP
CMS
User's
Guide
ST00-1584

VM/SP
CMS
Command
and Macro
Reference
ST00-1583

VM/SP
SP Editor
User's
Guide
ST00-1589

VM/SP
SP Editor
Command
and Macro
Reference
ST00-1590

VM/SP HPO
CP
Command
Reference
for General
Users
SC19-6227

VM/SP
SP
Interpreter
User's Guide
ST00-1593

VM/SP
SP
Interpreter
Reference
SQ24-5239

VM/SP
EXEC-2
Reference
ST00-1368

Reference Summaries

VM/SP
HPO
Commands
(General
User)
SX22-0003

VM/SP
HPO
Commands
(Other Than
General
User)
SX22-0004

Virtual
Machine
Problem
Determi-
nation
Reference
Information
GX23-0347

VM/SP
SP Editor
Command
Reference
Summary
ST00-1597

VM/SP
SP
Interpreter
Reference
Summary
ST00-1598

VM/SP
IPCS
Reference
Summary
ST00-1601

VM/SP
EXEC-2
Reference
Summary
ST00-1372

Figure 1 (Part 1 of 2). Virtual Machine/System Product High Performance Option Library

Preface

This publication is a reference manual that discusses the IBM-defined Class G and Class Any VM/SP High Performance Option (VM/SP HPO) CP commands available to the general user.

You should be familiar with the concepts of VM/SP HPO as outlined in the *IBM Virtual Machine/System Product Introduction*, Order No. GT00-1575. In addition, you should be aware of the logon process, terminal-initiated interruptions, terminal modes, and the method of switching from CP to virtual machine mode and vice versa. This information is detailed in the *IBM Virtual Machine/System Product Terminal Reference*, Order No. GT00-1581.

With the commands described in this publication, you can:

- Reconfigure your virtual machine and attached control devices
- Perform input and output spooling functions
- Simulate the functions of a real computer console.

Organization

This publication is organized in two parts: “Part 1. Guidance Information” and “Part 2. Reference Information”. There is also an appendix.

Part 1 has three sections:

“Section 1. The CP Command Syntax” describes the VM/SP High Performance Option command environments and the general structure of the CP command language.

“Section 2. The CP Command Language” provides additional information about the CP command set, including user-defined classes.

“Section 3. CP Command Usage” describes the use of CP commands to control terminal sessions, control I/O devices, test and debug programs, and control virtual machine functions.

Part 2 has two sections:

“Section 4. How to Use Command Syntax Diagrams” describes the syntax that is used in this publication to display command formats, command truncation, and abbreviations.

“Section 5. Format of CP Commands” contains reference data for all the class G and class Any commands in alphabetical sequence by command name. The command line format includes all operands along with a description of each. Any limitations or restrictions applicable to the values of variable operands are also defined.

“Appendix A. VM/SP High Performance Option Command Summary” contains all of the VM/SP HPO commands, with a brief description of each command’s function. This is intended to give the general user a perspective of the position that the CP class G and class Any commands occupy in the overall VM/SP HPO command structure.

Prerequisite Publications

The concepts of VM/SP High Performance Option are described in the *IBM Virtual Machine/System Product Introduction*, Order No. GT00-1575.

The physical operation of the virtual machine console is covered in the *IBM Virtual Machine/System Product Terminal Reference*, Order No. GT00-1581.

If the IBM 3767 Communications Terminal is to be used as a virtual console, the publication *IBM 3767 Operator’s Guide*, Order No. GA18-2000, is a prerequisite.

Associated Publications

Other VM/SP HPO CP commands, available only to system programmers and analysts, and service representatives, are described in the publication *IBM Virtual Machine/System Product High Performance Option Operator’s Guide*, Order No. SC19-6225.

The system may issue CP error messages and return codes to a user during a terminal session. These messages and codes are documented in the *IBM Virtual Machine/System Product High Performance Option System Messages and Codes*, Order No. SC19-6226 and *System Messages Cross-Reference*, Order No. SC23-0190.

For a glossary of VM/SP HPO terms, see the *Virtual Machine/System Product HPO Library Guide, Glossary, and Master Index*, Order No. GC23-0187.

The CMS component of VM/SP remains unchanged by the VM/SP High Performance Option program product. General users intending to use CMS should refer to the following publications for information about the CMS command language and its use:

- *IBM Virtual Machine/System Product CMS Command and Macro Reference*, Order No. ST00-1583.
- *IBM Virtual Machine/System Product CMS User’s Guide*, Order No. ST00-1584.

Reference Summaries

VM/SP HPO
Quick
Reference
SX22-0005

Diagnosis

VM/SP HPO
System
Messages
and Codes
SC19-6226

VM/SP HPO
System
Messages
Cross-
Reference
SC23-0190

VM/SP HPO
OLTSEP
and Error
Recording
Guide
SC19-6230

VM/SP
IPCS
Guide
ST24-5260

VM/SP HPO
Service
Routines
Program
Logic
LY20-0898

VM/SP
Data Areas
and Control
Blocks
Volume 2
(CMS)
LT00-1606

Auxiliary Service Support

VM/SP HPO
System
Logic and
Problem
Determination
Guide-CP
LY20-0897

VM/SP
System Logic
and Problem
Determination
Guide
Volume 2
(CMS)
LT00-1604

VM/SP HPO
Data Areas
and Control
Blocks-CP
LY20-0896

Device
Support
User's Guide
and Reference
GC35-0033

**Device
Support
Facilities
5748XX9**

EREP
User's Guide
and Reference
GC28-1378

**Environmental
Record Editing
and Printing
(EREP)**

Auxiliary Communication Support

RSCS
Networking
Version 2
General
Information
GH24-5055

RSCS
Networking
Version 2
Planning
and
Installation
SH24-5057

RSCS
Networking
Version 2
Operation
and Use
SH24-5058

RSCS
Networking
Version 2
Diagnosis
Reference
LY24-5228

RSCS
Networking
Version 2
Reference
Summary
SX24-5135

**RSCS Networking
Version 2
5664-188**

VTAM
General
Information
For VM/SP
GC30-3246

VTAM
Installation
and
Resource
Definition
SC23-0111

VTAM
Messages
and
Codes
for VM/SP
SC30-3275

VTAM
Data Areas
for VM/SP
SC30-3249

VTAM
Diagnosis
Guide
SC23-0116

VTAM
Diagnosis
Reference
SC23-0117

**Advanced
Communications
Function for
VTAM
(ACF/VTAM)
5664-280**

VM/Pass-
Through
Facility
General
Information
GC24-5208

VM/Pass-
Through
Facility
Guide and
Reference
SC24-5208

VM/Pass-
Through
Facility
Logic
LY24-5208

**VM/Pass-
Through
Facility
5748-RC1**

Figure 1 (Part 2 of 2). Virtual Machine/System Product High Performance Option Library

Contents

Introduction	1
Part 1. Guidance Information	3
Section 1. The CP Command Syntax	5
VM/SP HPO Command Environments	5
CP Command Structure	6
The Command Name	6
The Command Operands	6
Comments in the CP Command Language	6
Character Set Usage	7
Section 2. The CP Command Language	9
Privilege Classes for CP Commands	9
The Class G Commands	11
The Class Any Commands	12
Section 3. CP Command Usage	13
Controlling Input and Output Functions	14
Virtual Disks	15
Virtual Unit Record Devices	17
Virtual Unit Record Spooling	17
Dedicated Devices	21
Dedicated Channels	23
Controlling the Virtual Machine	23
Loading an Operating System	23
Simulating Interruptions	25
Entering CP Commands	26
Reconfiguring the Virtual Machine	28
Testing and Debugging of Programs	29
Stopping Execution of Your Virtual Machine	29
Displaying Virtual Storage	30
Altering Virtual Storage	33
Tracing Virtual Machine Activity	35
Part 2. Reference Information	45
Section 4. How to Use Command Syntax Diagrams	47
Section 5. Format of CP Commands	51
*	52
#CP	53
ADSTOP	55
ATTN	57
BEGIN	58
CHANGE	59
CLOSE	63
COMMANDS	67
COUPLE	68

CP	70
DEFINE	72
DETACH	83
DIAL	86
DISCONN	89
DISPLAY	91
DUMP	97
ECHO	101
EXTERNAL	102
INDICATE	103
IPL	107
LINK	113
LOADVFCB	119
LOGOFF	122
LOGON	125
MESSAGE	128
NOTREADY	130
ORDER	131
PER	133
Merging Trace Elements	144
Changing Options on Trace Elements	145
PURGE	155
QUERY	157
Responses	163
READY	185
REQUEST	186
RESET	187
REWIND	188
SCREEN	189
SEND	192
SET	194
SLEEP	212
SMSG	214
SPOOL	216
STORE	227
SYSTEM	231
TAG	233
TERMINAL	237
TRACE	246
TRANSFER	253
VMDUMP	257
Appendix A. CP Command Summary	263
Appendix B. Summary of CP Commands	271
Program Analysis and Monitor Functions	271
Device Control Functions	272
Virtual Console Functions	273
Spooling and File Control Functions	274
Miscellaneous Functions	275
Index	277

Figures

1. Virtual Machine/System Product High Performance Option Library x
2. Character Sets and Their Contents 7
3. IBM-Defined and Function Type CP Privilege Class Descriptions 10
4. Controlling the Virtual Machine During a Terminal Session 13
5. System Action to #CP Command 54
6. Action Taken on CLOSE for a Virtual Input Device 65
7. Action Taken on CLOSE for a Virtual Output Device 66
8. Available Form Width Codes 80
9. Password Requirements on the LINK Command 116
10. 3270 High Intensity Features 244

Introduction

With the commands described in this publication, the general user can reconfigure his virtual machine, control devices attached to his virtual machine, perform input and output spooling functions, and simulate many other functions of a real computer console.

Throughout this publication, the text is interspersed with examples of CP commands and system responses. All user input is shown in lowercase, and all system responses are shown in uppercase.

Part 1. Guidance Information

This part of the publication contains general information about the VM/SP HPO system. It contains three sections.

“Section 1. The CP Command Syntax” describes the CP command environments and the general structure of the command languages.

“Section 2. The CP Command Language” provides additional information on the CP command set.

“Section 3. CP Command Usage” describes a variety of user requests via class G CP commands.

Part 1 contains general information only. It is not intended for use in specific applications. You should refer to Part 2 for a more detailed discussion of commands.

Section 1. The CP Command Syntax

Like any other language, the CP command language has a set of rules and structures. This section describes the CP command language environment, syntax and structure, gives general descriptions of command names, operands, and their position on the command line, and describes character set usage in CP. A further discussion of CP commands is included in *Virtual Machine Running Guest Operating Systems*.

There are two types of CP commands: system commands and user-defined commands. Depending on your privilege class, you can issue CP system commands to do various system oriented functions. Privilege class is discussed in "Section 2. The CP Command Language". You, as a class G or general, user can use CP commands to simulate the function of the system operator's console. You can issue commands to control the operating system running in your virtual machine much like an operator controls the entire VM/SP HPO operating system using the keys and switches on his system console. User-defined CP commands are allowed, too; however, your system programmer must create them for you.

VM/SP HPO Command Environments

There are two basic command environments: the control program (CP) command environment and the virtual machine command environment.

You are in the control program (CP) command environment when you logon to VM/SP HPO and issue CP commands.

You are in the virtual machine command environment when you load your virtual machine with CMS or another operating system.

You must know whether you are in the control program (CP) or in CMS (or another operating system) so that you can use the correct commands. Check with your system programmer or use the manual describing your terminal's operation to find out how your terminal indicates in which command environment you are. On a 327X display terminal, you will see "CP READ" in the lower right corner of the screen when you are in CP mode.

When you load your virtual machine with CMS or another operating system (this loading is done by using the IPL command) and that operating system issues a read to the terminal, you will be placed in the virtual machine command environment. On 327X display terminals, the words "VM READ" appear in the lower right corner of the screen to indicate that you are in the virtual machine command environment. Commands that you enter at this time will be processed by the virtual machine operating system unless they are preceded by the characters #CP, where the # is the logical line-end character established for your session. (You can find out what line-end character has been established for your session by using the QUERY TERM command, and you can set it with the TERMINAL command.)

CP Command Structure

A CP command consists of a command name, usually followed by one or more positional operands. The general form for the CP command line is:

command name	[operand...]
--------------	--------------

You must use one or more blanks to separate each entry in the command line unless otherwise indicated.

The Command Name

The command name is an alphanumeric symbol of not more than eight characters. In general, the names are verbs that describe the function you want the system to perform. For example, you may want to find out whether or not a certain user is logged on the VM/SP HPO system. In this case, you would use the CP QUERY command.

The Command Operands

The command operands are keywords and positional operands of no more than eight alphanumeric characters each. The operands specify the information on which the system operates when it performs the command function. For the QUERY command, for example, you could use the USER or userid operand to find out whether the user is on the system.

Some commands require no operands; others require several. You can find each class G and class Any command with all of its operand requirements in "Section 5: Format of CP Commands" of this publication.

You must write the operands in the order in which they appear in the command formats in Section 5, unless otherwise specified.

Comments in the CP Command Language

You can write comments with CP commands of the following types:

- Commands with no operands
- Commands with a fixed number of operands
- Commands with a single optional operand

You should not write comments with commands that have a variable number of operands or with commands that have more than one optional operand. If you do, the comment could be interpreted as an operand.

You can enter comments on your console by using the CP * command.

Character Set Usage

CP commands are entered using a combination of characters from six different character sets. The contents of each of the character sets is described in Figure 2.

Character Set	Names	Symbols
Separator	Blank	
National	Dollar Sign Pound Sign At Sign	\$ # @
Alphabetic	Uppercase Lowercase	A - Z a - z
Numeric	Numeric	0 - 9
Alphanumeric	National Alphabetic Numeric	\$, #, @ A - Z a - z 0 - 9
Special		All other characters

Figure 2. Character Sets and Their Contents

Section 2. The CP Command Language

The CP commands represent a set of interactive console functions that are used (1) by the operator to control the VM/SP HPO system and (2) by a user to control a virtual machine. In this publication, only those commands that are used to control a virtual machine are fully documented. Other commands are referenced only to the extent necessary to show relationships and provide general knowledge.

Privilege Classes for CP Commands

Each user of VM/SP HPO is assigned one or more privilege classes as part of the directory entry of the virtual machine. Each user class is indicative of a specific function of a virtual machine and entitles the user to a specific subset of the CP command language.

Your installation may add to, or delete from, this set of commands in order to tailor the command set to your needs. Your installation administrator should make you aware of any changes of this type.

If your installation chooses to change user classes, you may record the changes in the blank column in Appendix A, Command Summary, in the back of the book.

This publication is a reference manual for class G and class ANY subsets of the CP command language.

Figure 3 lists each CP privilege class along with the associated type of user and function performed. Figure 3 also identifies the specific publications in which each class subset of CP commands is described in detail.

For a complete list of CP commands, refer to Appendix A in the back of this manual.

Although users are functionally divided into seven classes (A through G) with a corresponding privilege class of CP commands, there is another small group of commands available to all users. The eighth group consists of commands that do not belong to a privilege class because they are used to perform certain basic functions that are required by all virtual machines, such as logging on, logging off, and sending messages. These CP commands are in class Any.

The CP Command Language

IBM-Defined Class	Function Type	User And Function
A*	O	<p>Operations:</p> <p>The primary system operator can issue all Class A commands. The class A user controls the system. Class A is assigned to the user at the system console during IPL. The primary system operator is responsible for the availability of the system and its communication lines and resources. In addition, the class A user controls system accounting, broadcast messages, virtual machine performance options, and other command operands that affect the overall performance of VM/SP HPO. The system operator controls operation of the real machine using the system control panel and console device.</p> <p><i>Note: The class A system operator who is automatically logged on during CP initialization is designated as the primary system operator.</i></p> <p>If the system operator logs off and another class A user logs on, messages to the operator will be sent to this user, even if 'OPERATOR' logs back on, until the class A user logs off. If 'OPERATOR' logs off, then back on with an intervening logon of another class A user and the class A user then logs off, MSDMKMSG045E 'OPERATOR NOT LOGGED ON' will be received for messages to the operator until the system operator again logs off and back on. 'OPERATOR' must logon with no other class A users to be reestablished as the system operator.</p>
B*	R	<p>Resource:</p> <p>The system resource operator can issue all Class B commands. The class B user controls allocation and deallocation of all the real resources of the system, except those controlled by the primary system operator and spooling operator.</p>
C*	P	<p>Programmer:</p> <p>The system programmer can issue all Class C commands. The class C user updates certain functions of the system. The system programmer can modify real storage in the real machine.</p>
D*	S	<p>Spooling:</p> <p>The spooling operator can issue all Class D commands. The class D user controls spool data files and specific functions of the system's unit record equipment.</p>
E*	A	<p>Analyst:</p> <p>The system analyst can issue all Class E commands. The class E user displays the contents of real storage, performs the functions required to generate saved systems and discontinuous saved segments, and controls the collection and recording of performance measurement data. This class of user can display the contents of specified real storage areas on the virtual operator's console or on a spooled virtual printer, but cannot modify real storage.</p>

Figure 3 (Part 1 of 2). IBM-Defined and Function Type CP Privilege Class Descriptions

IBM-Defined Class	Function Type	User And Function
F*	C	<p>Customer Engineer:</p> <p>The service representative can issue all Class F commands. The class F user obtains, and examines, in detail, certain data about input and output devices connected to the system. The service representative can establish intensive recording mode for one I/O device at a time and can cause the recording of repressible machine check errors to be initiated or resumed.</p>
G+	G	<p>General:</p> <p>The general user can issue all Class G commands. The class G user controls functions associated with the execution of his virtual machine. A general user cannot display or modify real storage.</p>
ANY+		<p>The ANY classification is given to certain CP commands that are available to any user. These are primarily for the purpose of gaining and relinquishing access to the system.</p>
		<p>*Described in the <i>VM/SP HPO Operator's Guide</i>.</p> <p>+ Described in this publication.</p>

Figure 3 (Part 2 of 2). IBM-Defined and Function Type CP Privilege Class Descriptions

The Class G Commands

Commands in the general user class enable the user to control the operation of a virtual machine through using its virtual operator's console device much as a real machine is controlled through its system control panel.

For example, commands are included in the general user class that enable the user to perform the following operations: load an operating system (IPL command); start or restart program execution in the virtual machine (BEGIN command); stop virtual machine execution at a specified virtual storage address (ADSTOP command); reset and restart the virtual machine (SYSTEM RESET and SYSTEM RESTART commands); store and display data contained in virtual storage, registers (general, floating-point, and control), and status words such as the PSW (STORE and DISPLAY commands); display storage keys, the CSW, and the CAW (DISPLAY command); clear virtual storage to zeros (SYSTEM CLEAR command); and simulate an external interruption for the CPU timer, clock comparator, or external interruption button on the system console (EXTERNAL command).

The Class Any Commands

The class Any commands comprise those commands that are available, with the same format and operands, to all user classes. One of these commands, LOGON, is used to gain initial access to the system. The LOGON command identifies you to the system and sets up your virtual machine. The DIAL command allows you to establish a logical connection between your terminal and a previously logged-on multiple-access virtual machine. If you cannot log on for some reason, such as a faulty line connection or someone else using your user identification, the MESSAGE command allows you to communicate with the system operator or any other logged-on user.

Section 3. CP Command Usage

This section describes the CP commands provided for the general user that simulate the functions available on the operator's system console. The class G user can use these commands to control the system running in his virtual machine like an operator controls the entire VM/SP HPO system with console keys and switches. Figure 4 lists the steps a user must follow to prepare for a terminal session (steps 1 and 2), and operate a virtual machine (step 3).

Step Action	Function	Command Description
1. User must fill out request provided by installation ¹	Establish userid, user classes, passwords, storage requirements, disk space, and I/O devices	None
2. Installation builds directory entry for system user from information ¹	Directory entry provides CP with information necessary to create a virtual machine	CMS DIRECT command
3. User can activate terminal and begin work session	Enter CP commands to perform the function required	Available class G command
	a. Activate terminal	LOGON command
	b. Transfer and store data	LINK, SPOOL, STORE, and TRANSFER commands
	c. Record and control data	CHANGE, CLOSE, DISPLAY, DUMP, ORDER, PURGE, QUERY, and SPOOL commands
	d. Reconfigure the virtual machine	DEFINE and DETACH commands
	e. Communicating with the virtual machine	#CP, ADSTOP, ATTN, BEGIN, EXTERNAL, IPL, REQUEST, REQUEST, SET commands
	f. Handling errors	TRACE, DISPLAY, and DUMP commands
	g. End session	LOGOFF command
¹ Unless the installation or the users change system requirements, steps 1 and 2 do not have to be repeated every time a user activates a terminal after the initial session.		

Figure 4. Controlling the Virtual Machine During a Terminal Session

Controlling Input and Output Functions

The directory entry for your virtual machine includes, among other information, the I/O configuration required to run whatever operating system you are using in that virtual machine. This usually includes a virtual console, virtual printer, a virtual card reader, a virtual card punch, and virtual disks. A typical entry would look like:

```
(1)      USER VMUSER 123456 512K 2M G
(2)      ACCOUNT 87321 BIN14
(3)      OPTION ECMODE
(4)      CONSOLE 01F 3215
(5)      SPOOL 00C 2540 R
          SPOOL 00D 2540 P
          SPOOL 00E 1403
(6)      MDISK 130 3350 050 090 VDISK1 WR RWPASSWD
          MDISK 151 3330 001 020 VDISK2 WR RWPASSWD
          LINK CMSSYS 440 190 RR
```

where:

- (1) The userid of this virtual machine is **VMUSER**, and the password is 123456. The virtual machine's storage size is defined as 512K; however, this can be redefined up to 2 megabytes if, during an operating session, the need for additional storage arises (see the topic on "Reconfiguring the Virtual Machine"). The user privilege class is **G**.
- (2) **VM/SP HPO** generates accounting data for processor time and I/O device usage. This data is charged to a virtual machine by account number, which in this case is 87321. **BIN14** is a distribution code that is printed or punched on spooled printer and punch output. It usually denotes a location where the user can pick up spooled output of a terminal session.
- (3) Option **ECMODE** allows the virtual machine to use the complete set of virtual System/370 control registers and the dynamic address translation feature of the System/370.
- (4) The virtual address of the virtual machine console is 01F, and your operating system communicates with the terminal as if it were a 3215/3210.
- (5) The virtual unit record devices (reader, punch, and printer) are to be spooled and their addresses, as far as the virtual machine is concerned, are 00C, 00D, and 00E, respectively.
- (6) In this configuration, a 90-cylinder read/write minidisk with a virtual address of 130 is located on cylinders 50 through 139 of a real 3350 volume labeled **VDISK1**. Similarly, a 20-cylinder read/write minidisk with a virtual address of 151 is located on cylinders 1 through 20 of a real 3330 volume labeled **VDISK2**. The last entry provides a link to a shareable CMS system volume so the user can use such CMS functions as the CMS Editor, System Product Editor, **EXEC** or **EXEC2** processor. The **LINK** entry makes the virtual disk at virtual address 440 available to your virtual machine in read-only mode at virtual address 190.

Notes:

1. *The read password entry (RR) on the LINK command means that anyone with the proper password can share the use of the minidisk in read-only mode.*
2. *The write access mode (WR) on the MDISK command means that you can write to the minidisk as long as no one else has a link to it; if someone else links to it before you logon, you still have read-only access. Use the write access mode with great care. CMS does not protect a user from loss of data on a disk when multiple users have write access to the disk. More than one user writing to the same virtual device can result in a permanent loss of data.*
3. *For detailed information on directory entries including a discussion of the password in the control statements above, see the section "Directory Control Statements" in VM/SP HPO Planning Guide and Reference.*

Some devices such as unit record devices (readers, printers and punches) are usually defined as being spooled; in this way, a few real unit record devices can support a large number of virtual unit record devices.

Other devices such as magnetic tapes require a one-to-one virtual-to-real correspondence. This means that for its period of use, the device must be dedicated to one virtual machine. For this reason, these devices are not permanently assigned (that is, they are not defined in the user's directory entry). Rather, they are temporarily attached to a user as needed.

Virtual Disks

A single real direct access storage device (DASD) can be managed as if it were made up of a number of virtual disks (minidisks).

Virtual disks are cylinder or block extents on real DASD. To the virtual machine, they are functional equivalents of real disks. They can range from 1 cylinder on a count-key-data device or 1 block on an FB-512 device to all of the DASD space available on a real DASD volume. Virtual disks can be permanently or temporarily defined for your virtual machine.

Permanent Virtual Disks

Permanent virtual disks are defined in your virtual machine directory entry. They can be your own personal disks which you may or may not wish to have other users access; or, they may be common disks, owned by one user, but generally shared in read-only mode by any user on the system. Virtual disks, defined in your directory, are made available to you when you logon to the system.

CP Command Usage

Temporary Virtual Disks

If during a terminal session you require additional disk space, you can define a temporary minidisk via the CP DEFINE command:

```
DEFine T3330 [As] 133 [CYL] 15
```

In this example, a virtual 3330 disk comprising 15 cylinders is allocated to your virtual machine at virtual address 133. You can then notify your operating system of the additional storage space using the appropriate control statements or commands.

Because the previous structure or use of this temporary disk space is unknown, you must format it to conform to the operating system you are using. For CMS, use the CMS FORMAT command. For OS, DOS, or VSAM applications, use the Device Support Facility.

When you have no further need for temporary disk space, you can release the space to the system by issuing the command:

```
DETach 133
```

If you do not release it during your terminal session, it is automatically released to the system when you logoff.

Your installation's system programmer can specify an option during system generation time that allows CP to automatically format (clear to hexadecimal zeroes) all temporary disk space when you release that space or logoff the system. If your installation does not choose this option, you should, for security purposes, always format (clear) all temporary disk space before logging off or releasing temporary disk storage to free storage.

Sharing Virtual Disks

You can also gain temporary access to someone else's permanent virtual disk during your terminal session. You must know the userid of the disk's owner, as well as its virtual address in his system. If the owner is controlling the access to his disk, you have to obtain the read or write password. You can then issue the command:

```
LINK To smith 330 As 134 rr rpasswd
```

The virtual disk at address 330 in user SMITH's configuration is made available to your virtual machine at address 134. You have read-only access even if SMITH has the disk in write status.

Notes:

1. *For further information on virtual disks, consult VM/SP CMS User's Guide.*
2. *If your installation is using the password suppression facility, you cannot specify the password on the LINK command line. For further information, refer to discussion of "LINK" on page 113.*
3. *If your virtual machine is equipped to work with preferred machine assist, there are special restrictions on the use of virtual devices. For detailed information, refer to the discussion of "DEFINE" on page 72 and "LINK" on page 113.*

Virtual Unit Record Devices

The directory entries for your virtual machine usually define at least one reader, punch, and printer. These are the virtual unit record devices that your operating system addresses when performing unit record input and output. Rather than dedicate a real device to each virtual unit record device, VM/SP HPO uses the concept of spooling to provide a buffered interface between the few real unit record devices and the relatively large number of virtual unit record devices.

Virtual Unit Record Spooling

Whenever your operating system creates a punch or printer file, VM/SP HPO organizes the output data into a spool file. VM/SP HPO then places the file in a queue by device type and stores it on auxiliary DASD for eventual processing by the real unit record device. Similarly when card input data is read on a real card reader, VM/SP HPO organizes the data into a spool file and places it in the queue for the specified virtual card reader.

You can assign certain characteristics to spool files that enhance their manageability by VM/SP HPO. Some of these characteristics such as spooling class and distribution code are assigned in the directory entries. Other characteristics such as number of copies, concatenation of files, destination, and hold status are set to default values whenever you log on.

The CP SPOOL and CHANGE commands allow you to change some or all of the above characteristics. To change spool file characteristics by device, use the SPOOL command. To change characteristics by file, use the CHANGE command.

VM/SP HPO identifies individual spool files by assigning each file a spool identification number (spoolid). This number ranges from 0001 through 9900. After 9900, VM/SP HPO starts again from 0001. If CP has not yet processed spoolid 0001, CP will assign the next available spoolid to the new file. One series of spoolids covers the reader, printer, punch, and console spool files. VM/SP HPO normally assigns the spoolid to a spool file when that file is closed. An exception to this is a console spool file where VM/SP HPO assigns the number when the file is opened. Also, if you close a console file without stopping console spooling, VM/SP HPO automatically opens another console spool file with a new spoolid at the same time.

Spool File Characteristics

The spooling class of a virtual device logically groups its output with output from similar virtual devices belonging to other users. For example, you could spool all printer output that required two-part green striped forms as class G. The real printer could then be set up with two-part green striped forms and told to process only class G spool files. If you issue the command:

```
SPOOL 00e CLASS g
```

all subsequent output of the virtual unit record device at address 00E has a spool file class G. You can use the SPOOL command to set the spooling class of files before they are closed.

Similarly, you can spool output by form name or form number when all your output requires a particular form such as a tax form or inventory form. The form name is a one-to-eight character alphameric name assigned by your installation's system programmer at system generation time.

Using the FORM operand on certain class G commands, you can change, close, purge, query, order, spool, and transfer spool files by their form name.

For instance, you can issue:

```
CHANGE PRINTER FORM fedtax FORM statetax NOHOLD
```

to change all your printer files with a form name of FEDTAX to STATETAX. These files are then printed on a printer set up by your spooling operator to process state tax forms.

If you decide to change the class of a spool file that has been closed but not selected for processing by a real device, you can issue the command:

```
CHange Printer 1234 CLass m
```

and the printer spool file with a spool identification (spoolid) number of 1234 has its class changed to M. If you do not remember the spoolid of the spool file, you can issue the command:

```
Query Printer ALL
```

and all printer spool files that have not been processed or selected for processing have a one line description displayed at the terminal. The information displayed includes the spoolid number and the filename and filetype as well as other spool file characteristics.

The HOLD/NOHOLD status is a characteristic of a spool file that determines what happens to the file after it is closed. For example:

```
SPool Printer HOLD
```

prevents the release of all subsequent printer output spool files to the real printer. If many files are being generated, of which only few are required to be printed, you can hold all files with the HOLD option. You can then release only the required files via the NOHOLD option of the CHANGE or CLOSE commands.

If, when you are about to close a file, you know that the file is to be printed, you can issue the CLOSE command with the NOHOLD option:

```
CLose Printer NOHold
```

The HOLD status of the SPOOL command is overridden for that one file and it is placed on the real printer queue with a NOHOLD status.

If you do not know which files are to be printed until they are all closed, you can use the QUERY command:

```
Query Printer
```

to determine the spoolid numbers of all the closed printer files. Then issue the CHANGE command to alter the HOLD status of each file to be released. For example, if the file with a spoolid of 0246 is to be released, enter the command:

```
CHange Printer 0246 NOHold
```

To change your printer so that all subsequent files are automatically released to the real printer, enter:

```
SPool Printer NOHold
```

If your virtual reader has the default status NOHOLD, input spool files are deleted from the system after they have been read. To prevent this, enter:

```
SPool Reader Hold
```

CP Command Usage

Reader files are now held in the system until you issue:

```
SPool Reader NOHold
```

and reread the files. If you want to delete the files immediately, enter:

```
PURge Reader all
```

Again, you have the option of using the **CHANGE** command to control the **HOLD** status of specific reader files via the spoolid number.

The **CONT/NOCONT** status of a spooling device controls the concatenation of files. Since this is a characteristic of a spooling device, it can be changed only by the **SPOOL** command. When **CONT** is in effect, the spooling device ignores input end-of-file indicators and output **CLOSE** requests.

For output devices, the effect is to concatenate multiple output files into one logical spool file.

For input devices the effect is to read files and ignore end-of-file indicators until all files spooled to the virtual reader have been read. At that time, the end-of-file indicator that is reflected to the virtual machine depends on the **EOF/NOEOF** operand setting of the **SPOOL READER** command. **EOF** results in a unit exception that corresponds to pressing the end-of-file button on a real card reader. **NOEOF** results in a unit check/intervention required status.

Virtual Console Spooling

While spooling is usually associated with unit record equipment, **VM/SP HPO** also spools both input and output data that is displayed on your virtual console. To start console spooling, enter the command:

```
SPool CONsole STArt
```

If you wish to spool your console data and not have it displayed on the terminal, you can enter:

```
SPool CONsole STArt NOTerm
```

Note, however, that the **NOTERM** operand (or its default value, **TERM**) is not effective until console spooling has been started. Also, the **NOTERM** option does not prevent the printing of:

- CP commands entered from CP mode
- Commands entered on a display terminal in **CMS EDIT** mode

When you wish to stop console spooling, enter the command:

```
SPool CONsole STop
```

This stops any further spooling of console data but does not close the console spool files. For that, you must enter the command:

```
Close CONsole
```

The converse is also true—closing the console spool file does not automatically stop console spooling.

Reordering and Purging Spool Files

If you want to change the sequence of your closed spool files, issue the **ORDER** command to place your files in a specific order by device type. Files can be resequenced by spoolid or class, or both when using the **ORDER** command. For example:

```
ORDer Printer 2468 1357 CLass c
```

resequences your closed printer spool files as follows:

```
file 2468
file 1357
all class C files (in their original sequence)
other files in original sequence
```

To remove any of your closed spool files from the system, use the **PURGE** command. Files can be specified individually by spoolid or by class, or both. For example:

```
PURge 1234 CLass b
```

removes file 1234 and all class B files from the system.

```
PURge PUnch ALL
```

removes all punch files.

```
PURge ALL
```

removes all files from all devices.

*Note: The **CHANGE**, **ORDER**, and **PURGE** commands can be used only on closed spool files that belong to you and have not yet been selected for processing.*

Dedicated Devices

If a device's use is restricted to a single virtual machine, that device is a *dedicated* device. Some devices, such as disks and unit record equipment can function in dedicated mode at one time and in shared or nondedicated mode at other times. Devices such as magnetic tape drives can only be used as dedicated devices.

The directory program can dedicate a device via a **DEDICATE** control statement in the directory entry for a virtual machine at system generation or a device can be dedicated dynamically using the system operator's **ATTACH** command.

CP Command Usage

If the directory entry for your virtual machine includes a statement such as:

```
DEDICATE 495 295
```

the device at real address of 295 is made available to your virtual machine as virtual address 495 when you logon to the system. It remains dedicated to your virtual machine until you either logoff or release the device using the DETACH command:

```
DETach 495
```

The format of the DEDICATE control statement is described in detail in “Part 2: Defining Your VM/SP HPO System” of the *VM/SP HPO Planning Guide and Reference*. More than one user can have the same real device specified as being dedicated to his virtual machine. However, the first user to logon gains access to the device and others have to wait until the current user either logs off or releases the device.

If you are using the DEDICATE statement for a DASD, you can specify a particular disk volume by its volume serial number rather than by its real device address. This has the effect of not restricting you to a particular real device. In the event that a malfunction precludes the use of the volume on one DASD, the disk pack can be transferred to another DASD and still be accessed by its volume serial number.

If a device that cannot be shared, such as a magnetic tape device, is not required for the entire terminal session, it may be more practical to dedicate it as required. Since the ATTACH command is a class B command and not usually available to the general user, you can send the system operator a message:

```
MSG OPERator pls attach 281 to smith as 181
```

The operator issues the command:

```
ATTach 281 To smith As 181
```

If the device (assumed to be a magnetic tape device) was available and the command was completed successfully, you receive the verification response:

```
TAPE 181 ATTACHED
```

When you have no further use for the device, issue the command:

```
DETach 181
```

You receive the acknowledgment:

```
TAPE 181 DETACHED
```

and the device is now available to some other user. If you do not DETACH the device, it remains dedicated to your virtual machine until you log off.

Dedicated Channels

A user can have an entire channel with all its devices dedicated to his virtual machine. CP does not translate device addresses since the virtual addresses must be the same as the real device addresses. Contention for use of the channel is minimized, since all of the channel resources are dedicated to a single virtual machine. The ATTACH command is a class B command and is not usually available to the general user. However, you can send a message to the system resource operator:

```
MSG OPERator pls attach channel 2 to smith
```

If the channel is available, the operator issues:

```
ATTach CHANnel 2 To smith
```

when the command has been successfully completed, you are notified by the response:

```
CHANNEL 2 ATTACHED
```

When a dedicated channel path is no longer required for virtual machine I/O operations, you can issue the command:

```
DETach CHANnel 2
```

After the channel is detached, the following message is sent to you and the primary system operator:

```
CHANNEL 2 DETACHED userid
```

Controlling the Virtual Machine

VM/SP HPO provides several commands with which you can load an operating system into your virtual machine, simulate hardware interruptions to your virtual machine, enter CP commands while in a virtual machine environment, and reconfigure your virtual machine dynamically during a terminal session.

Loading an Operating System

At the completion of the LOGON procedure, you have a virtual machine of a predetermined configuration at your disposal. As with a real machine, its use is limited without an operating system. An operating system can be loaded using the IPL command or automatically at logon if an IPL control statement is included in your virtual machine's directory entry.

CP Command Usage

When a virtual machine runs the same operating system with very few exceptions, it may be expedient to use automatic loading. If the directory entries for your userid include an IPL control statement with the name or the virtual address of a specific system to be loaded, that system is automatically loaded when you logon unless you issue the LOGON command as:

```
Logon userid noipl
```

in which case, the automatic loading does not take place and you can IPL any supported system that is available.

A more flexible approach is loading an operating system using the IPL command. Again you have a choice. You can IPL by device address or by system name. If you IPL by the virtual address of the device containing the operating system, you can also clear virtual storage to binary zeros before loading the system:

```
Ipl 190 CLear
```

This facility is useful if your operating system does not automatically clear storage when it is loaded.

You can stop the IPL procedure at a point just before the initial PSW is loaded by issuing the IPL command with the STOP operand:

```
Ipl 190 CLear STOP
```

At this point, you can issue CP commands to display or alter data in your nucleus. To restart the virtual machine, issue the command:

```
Begin
```

You can load your operating system by name provided that the name refers to a system that has been previously saved by your installation's system programmer. As an example:

```
Ipl dosvs
```

If you do load by name, the options to clear storage or to stop before loading the initial PSW are not available. Whether you IPL by device address or by name, you can pass up to 64 bytes of data (including embedded and trailing blanks) to your operating system using the IPL command. The data is entered following the keyword PARM:

```
Ipl dosvs parm this data will be passed
```

VM/SP HPO loads the general registers, starting with register zero with the contents of the command line beginning with the first nonblank character after the keyword PARM and ending with the last character or blank entered. Embedded and trailing blanks are passed. For the previous example, the general registers contain the following:

```
GPR 0  THIS    DAT    A WI    LL B
GPR 4  E PA    SSED   xxxx   xxxx
GPR 8  xxxx    xxxx   xxxx   xxxx
GPR 12 xxxx    xxxx   xxxx   xxxx
```

where xx... denotes no change to previous contents. If you had entered three blank characters following the word "passed" in the command line, the registers would contain:

```
GPR 0  THIS    DAT    A WI    LL B
GPR 4  E PA    SSED   x     xxxx
GPR 8  xxxx    xxxx   xxxx   xxxx
GPR 12 xxxx    xxxx   xxxx   xxxx
```

For further information on the PARM operand, refer to usage note 4 on page 110.

Simulating Interruptions

The EXTERNAL command can be used to simulate an external interruption to the virtual machine and return control to that virtual machine. If you enter the command:

```
EXtErnal
```

a default interruption code of X'40' is assumed. This code is equivalent to pressing the external interruption key on the real system control panel. You can also specify an interruption code within the range of X'01' through X'FF':

```
EXtErnal A8
```

If your virtual machine has the ECMODE option specified in the directory, you can also code X'1004' (clock comparator interruption) and X'1005' (CPU timer interruption). The interruption code that you enter or default to, is placed in position 16 through 31 of the PSW if you are operating in basic control (BC) mode. If your machine is in extended control (EC) mode, the interruption code is placed in the two bytes at location X'84'. An external interruption is then presented to your virtual machine and subsequent action is determined by your operating system.

Note: If your virtual machine is operating in the basic control mode (ECMODE OFF), virtual addresses specified can be no higher than 5FF (channels 0 through 5). If it is operating in extended control mode (ECMODE ON), all addresses through FFF are valid.

CP Command Usage

The attention or request keys on the real console can be simulated by the CP commands:

ATTN

or

REQuest

In either case, the effect is to interrupt the running condition of the virtual machine and ready it for console input.

Entering CP Commands

There are several ways in which you can stop your virtual machine in order to issue CP commands. You can stop the virtual machine and place it in a virtual console read environment; you can go from a virtual console read environment to the CP environment; or, you can stop a virtual machine and place it directly into the CP environment.

Entering CP Commands in the Virtual Machine Environment

If the screen status area of your terminal indicates that you are in the virtual machine environment, you should press the attention or PA1 key (or its equivalent) to stop the virtual machine and place it into the virtual console read environment. This assumes that the SET RUN command has been set to ON.

You can now execute CP commands by using either the #CP command prefix or the CP command prefix before a CP command.

Using the #CP command prefix causes the immediate execution in the CP environment of the command following the #CP prefix. CMS is not involved. Because of this, you can use this prefix when an operating system other than CMS is running your virtual machine. CP processes your command and returns control immediately to your virtual machine. You can now execute CP commands by entering one or more command lines as operands of the #CP command. For example:

```
#CP Query Time
```

or

```
#CP Query Time#Query Users
```

where the latter example shows how multiple CP command lines can be entered.

Note: The pound symbol (#) both before this CP command and between the commands represents the logical line end symbol in effect for your virtual machine. Any other character can perform the same function

*if you so designate it with the **TERMINAL LINEND** command. To determine what logical line end symbol is in effect in your virtual machine, use the **QUERY TERMINAL** command.*

If you are using a 3270 display terminal where the keyboard is not activated when the virtual machine is running, you can enter either of the following commands:

```
#CP ATTN
```

or

```
#CP REQUEST
```

to both simulate an attention interruption for your virtual machine and place it in a virtual console read environment. On a 3270 terminal, you can also enter a command such as:

```
#CP Query Time
```

while the virtual machine is running. The control program (CP) interrupts the virtual machine, executes the CP command, and restarts the virtual machine.

The **#CP** command, entered without any command line operands, places the virtual machine in the CP environment. You can then enter CP commands directly. For example, the following sequence:

```
#CP
Query Time
Query Users
```

places you in CP mode and then executes the two CP commands.

Using the CP command prefix before a CP command is done like this:

```
CP Query Time.
```

This causes CMS to pass the command to CP for execution. Since you can issue CP commands while in CMS, this command prefix is usually unnecessary. It must be included, however, before a CP command when:

- **SET IMPCP OFF** is in effect,
- it is needed to distinguish it from user-written commands of the same name,
- it is issued from a CMS EXEC or EXEC 2 file and is not processed by the system product interpreter (that is, REXX commands).

Whether you reached CP using **#CP** or **CP** as a command prefix, you can leave that environment in the same way. In either case, use the **BEGIN** command or press the PA1 key.

CP Command Usage

Entering CP Commands from CP Console Function Mode

When you are in CP console function mode, you can enter CP commands such as:

```
Query Time
```

or

```
Query Users
```

directly, and you remain in CP mode until you issue the command:

```
Begin
```

The BEGIN command returns you to the environment from which you entered the #CP command that placed you in CP mode. This would be either the virtual machine executing or the virtual console read environment.

Reconfiguring the Virtual Machine

You can alter your virtual machine configuration (stored in the directory) to conform to specific situations that arise. Some examples are: a seldom used compiler may need additional disk work area or a larger virtual storage. A new application program in test status may need additional I/O devices or require a different channel mode of operation.

The CP DEFINE command allows you to change your configuration temporarily for the current terminal session. For example:

```
DEFine Reader 00b
```

temporarily adds a card reader at virtual address 00B. The entry:

```
DEFine T3330 As 291 CYL 125
```

adds a temporary virtual disk, containing 125 cylinders, at virtual address 291.

If your virtual machine is operating in basic control (BC) mode (that is, your virtual machine uses a maximum of 6 channels, 0 through 5), you can only define virtual devices with addresses up through 5FF; in extended control (EC) mode, you can use all addresses through FFF.

You can enter:

```
DEFine STORage As 768K
```

to temporarily change the virtual storage size of your virtual machine to 768K (K=1024 bytes). When you redefine storage, your virtual machine is automatically reset and you must reload your operating system.

Note: If the directory specifies a shared and a nonshared device on the same virtual control unit, an error message (DMKDIR768E) will be issued and the directory will not be updated. Define these devices on different virtual control units and reissue the DIRECT command, described in the VM/SP HPO Planning Guide and Reference.

Testing and Debugging of Programs

In addition to the testing and debugging facilities that are contained in your virtual machine's operating system, you can use several CP commands to locate, display, trace, and alter the program instructions that are executing in your virtual storage.

In the following discussions of the ADSTOP, DISPLAY, DUMP, STORE, and TRACE commands, it is important that you understand the levels of storage you can specify on the command line.

First level storage is real main storage. Only second level storage (storage that is real to the virtual machine) can be handled directly. This includes the V=R partitions or regions of DOS/VS, VSE systems, and OS/VS as well as storage in OS/PCP, MFT, and MVT. The V=V partitions or regions of DOS/VS, VSE systems, and OS/VS constitute third level storage and cannot be specified directly. The user, or the virtual operating system, is responsible for converting any third level storage addresses to second level storage addresses before including them in the specific command line. (This is equivalent to converting from second level to first level as seen by the virtual machine.)

Stopping Execution of Your Virtual Machine

To stop execution of your virtual machine at a given address in virtual storage, use the ADSTOP command and specify the hexadecimal address of a virtual instruction. The command:

```
#CP ADSTOP 3000
```

stops the virtual machine when the instruction at hexadecimal location 3000 is the next instruction to be executed. When the machine stops running, you receive the message:

```
ADSTOP AT 3000
```

and your terminal is placed in CP console function mode. At this point, you can enter other CP debugging commands to display and alter storage or to trace certain instructions. When you want to resume running your virtual machine, enter:

```
Begin
```

Unlike the hardware address stop, ADSTOP is turned off when:

- Execution reaches the specified hexadecimal address
- You issue another ADSTOP command (including ADSTOP OFF)
- You issue the IPL command
- You perform a system reset.

While ADSTOP is on, the SVC portion of virtual machine assist is not executed. When ADSTOP is turned off, SVCs are again handled by virtual machine assist.

Displaying Virtual Storage

The contents of virtual storage, storage keys, general registers, floating-point registers, control registers (if in EC mode), PSW, CAW, and CSW can be displayed on your terminal using the DISPLAY command. All but the CAW and CSW can be printed on the spooled virtual printer using the DUMP command. Using the VMDUMP command, you can also produce formatted storage dumps of guest virtual storage when your installation has the VM/IPCS Extension program product.

The DISPLAY, DUMP and VMDUMP commands are fully described in Section 5. The examples that follow do not attempt to show all possible uses.

If you are running MVS/SP Version 1 Release 3 or a later release in your virtual machine and if preferred machine assist is enabled (whether or not you are using control switch assist), you may use storage above 16 Mb. However, the DISPLAY, DUMP, and VMDUMP commands and CP diagnose and inter-user communication vehicle (IUCV) will not service the additional storage above the 16 Mb line.

Displaying Virtual Storage at Your Terminal

With the DISPLAY command, you can display virtual storage at your terminal in either of the following formats:

- Four byte groups, aligned on fullword boundaries, hexadecimal format, printed four fullwords per line.
- 16 byte groups, aligned on 16 byte boundaries, hexadecimal format, printed four fullwords plus EBCDIC translation per line.

For the first format, enter the DISPLAY command as:

```
Display 1026-102c
```

you receive the response:

```
001024 xxxxxxxx xxxxxxxx xxxxxxxx
```

For the second format, enter the command as:

```
Display t1026-102c
```

and the response is:

```
001020 xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx *.....*
                                         (EBCDIC trans.)
```

You can also specify the area of storage to be displayed by entering a hexadecimal byte count such as:

```
Display 1024.12
```

The response displays 20 bytes as follows:

```
001024 xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx
001034 xxxxxxxx
```

Byte Alignment on Terminal Output

The previous responses illustrate the byte alignment that takes place in each of the two display formats.

If the first location to be displayed is not on the appropriate 4 or 16 byte boundary, it is rounded down to the next lower boundary that applies.

If the last location to be displayed does not fall at the end of the appropriate 4 or 16 byte group, it is rounded up to the end of that group.

If you enter:

```
Display K1024-3200
```

the storage keys that are assigned to each segment of the specified storage area are displayed.

To display all storage keys, enter:

```
Display K
```

If your virtual machine is in extended control mode (ECMODE), you can inspect any of the control registers:

```
Display X1 4 A
```

and receive the response:

```
ECR 1 = xxxxxxxx
ECR 4 = xxxxxxxx
ECR 10 = xxxxxxxx
```

However, the same command entered while your virtual machine is not in extended control mode results in the response:

```
ECR 0 = xxxxxxxx
ECR 0 = xxxxxxxx
ECR 0 = xxxxxxxx
```


CP Command Usage

As each operand in the command line is processed, VM/SP HPO determines that ECMODE is OFF and replaces any reference to a control register with ECR 0, the only control register available in basic control (BC) mode.

Printer Output

With the DUMP command you can dump the contents of all registers, the PSW and the storage keys, along with any specified area of virtual storage, to the virtual machine's spooled printer. The printer format for storage locations is 8 fullwords per line plus the EBCDIC translation on the right.

To print only the registers, the PSW, and the storage keys, you need only enter:

```
DUMP 0
```

To also print an area of virtual storage, you can specify the beginning and ending hexadecimal locations:

```
DUMP 1064-10ff
```

You can also specify the beginning location and the number of bytes to be dumped; both values are entered in hexadecimal:

```
DUMP 1064.9b
```

If you are printing a series of dumps, you can identify each one by including its identification on the DUMP command line, following an asterisk:

```
DUMP 1000-2000 * dump no. 1
```

To print the dump data on the real printer you must first close the virtual printer. Issue the command:

```
Close Printer
```

and the dump data spool file is placed on an appropriate system printer queue.

In addition, if your installation has the VM/IPCS Extension program product installed, you can use the VMDUMP command that dumps storage for guest virtual machines. VMDUMP provides the VM/IPCS Extension with header information to identify the owner of the dump; it also maintains dump information, writes the dump to a spool file, and formats the dump.

When you enter at the terminal

```
VMDUMP 150-200
```

or

```
VMDUMP 400:500
```

CP dumps the contents of virtual machine storage at the hexadecimal addresses between 150 and 200 or between 400 and 500, respectively.

If you enter

```
VMDUMP 150.50
```

CP dumps the contents of virtual storage starting at hexadecimal address 150 for a total of X'50' bytes.

Altering Virtual Storage

You can alter the contents of your first level virtual storage, general registers, floating-point registers, control registers (if available), and the PSW with the STORE command.

Virtual storage can be altered in either fullword or byte units.

When using fullword units, the address of the first positions to be stored must have either an L or no prefix:

```
STore 1024 46a2
```

or

```
STore L1024 46a2
```

results in 000046A2 being stored in locations 1024 through 1027.

```
STore 1024 46 a2
```

on the other hand, implies storing 2 fullwords and results in the storing of 00000046000000A2 in locations 1024 through 102B.

If the starting location is not a multiple of a fullword, it is automatically rounded down to the next lower fullword boundary. Each fullword operand can be from one to eight hexadecimal characters in length. If less than 8 characters are specified, they are right justified in the fullword unit and padded to the left with zeros.

You can store in byte units by prefixing the start address with an S.

```
STore S1026 d1d6c5
```

stores D1D6C5 in locations 1026, 1027, and 1028. Note that the data storage is byte aligned. If an odd number of hexadecimal characters is specified, CP does not store the last character, you receive an error message, and CP terminates the function. For example, if you specify:

```
STore S1026 d1d6c
```

CP stores d1 at X'1026', and d6 at X'1027'; when CP attempts to store c at X'1028', it recognizes an incomplete hexadecimal character, and does not store the last character.

CP Command Usage

You can store data into one or multiple consecutive registers.

General and control registers are loaded in fullword units. For example,

```
STore G4 123456
```

loads general register 4 with 00123456.

```
STore G4 12 34 56
```

loads general registers 4, 5, and 6 with 00000012, 00000034, and 00000056, respectively.

Floating-point registers are loaded in doubleword units. Each doubleword operand can be from 1 to 16 hexadecimal characters in length. If less than 16 characters are specified, they are left justified in the doubleword unit and padded to the right with zeros. For example:

```
STore Y2 00123456789
```

loads floating-point register 2 with the value 0012345678900000.

You can use the STATUS operand of the STORE command to simulate the hardware store status facility. Selected virtual machine data is stored in permanently assigned areas in low storage. Your virtual machine must be in extended control mode for the command:

```
STore STATUS
```

to be accepted. To place your virtual machine in extended control mode, issue the command:

```
SET ECmode ON
```

Be aware that this command resets your virtual machine and you must reload (IPL) your operating system.

The data stored by the STORE STATUS command is summarized in the following table:

Virtual Address		No. of Bytes	Data
Dec.	Hex.		
216	D8	8	Processor Timer
224	E0	8	Clock Comparator
256	100	8	Current PSW
352	160	32	Floating-Point Registers (0, 2, 4, and 6)
384	180	64	General Registers (0-15)
448	1C0	64	Control Registers (0-15)

Note: If the operating system that is running in your virtual machine operates in the basic control mode, these areas of low storage may be used for other purposes. You should not use this facility under these conditions.

If you are running MVS/SP Version 1 Release 3 or a later release in your virtual machine and if preferred machine assist is enabled, you may be using storage above 16 Mb. However, you cannot use the STORE command to alter this additional storage.

Tracing Virtual Machine Activity

The TRACE command allows you to trace certain virtual machine activities and record the results on either your terminal or the virtual printer, or on both.

You can trace any one or more of the following virtual machine activities:

- SVC interruptions
- I/O interruptions
- Program interruptions
- External interruptions
- Privileged instructions
- PSW instructions
- Successful branches
- Instruction execution

Note: When MVS/System Extensions support is enabled, trace output may have gaps in the tracing sequences. Each gap represents a period when tracing was interrupted.

VM/SP HPO stops tracing an activity when a virtual-machine-extended-facility assist instruction results in a branch.

(Virtual-machine-extended-facility assist instructions are described in the publication *IBM System/370 Extended Facility, GA22-7022*.) Tracing resumes when these two conditions are met:

1. When CP gains control, such as for a real I/O interruption; and
2. When the virtual machine encounters one of the specified activities to be traced, except for successful branching.

Whenever you are recording trace output at your terminal, the virtual machine stops execution and enters the CP console read environment after each output line. This is the default mode of operation when, for example, you enter:

```
TRace ALL
```

or

```
TRace SVC PROgram BRanch
```

If you only want to record the trace and not stop after each output line, add the RUN operand as the last entry on the command line.

If, having specified multiple activities to be traced, you decide to stop tracing one or more of them, enter:

```
TRace PRoGram BRanch Off
```

and tracing is now confined to SVCs only.

To trace all activity with the output directed to the virtual printer, enter:

```
TRace ALL PRINter
```

When you stop tracing, you must also issue the **CLOSE** command to print the spooled trace output on the real printer:

```
TRace END  
Close Printer
```

If your virtual machine configuration contains only one printer, trace output is mixed with application output. You should define another virtual printer with an address lower than the previously defined printer. Application output is still directed to the original printer; however, trace output is always directed to the printer with the lowest address. For a complete description of the **TRACE** command along with the trace output formats, see discussion of the command in Section 5.

While trace is running, portions of virtual machine assist are disabled. When the trace is complete, they are enabled. For more information, see the **TRACE** command discussion in Section 5.

Using the PER Command

You can use the **PER** command to trace all instructions, all successful branches, all register alterations, and all instructions executed in your virtual machine that alter storage. The **PER** command has many options that allow you selectivity in choosing which events are to be monitored. Trace output for the **PER** command is always produced *after* the instruction executes.

The **RANGE** option of the **PER** command allows you to monitor events that occur after the execution of instructions within a specified range or ranges. In the following examples, assume that there is a program loaded at location 20000 and that the program is 500 bytes in length. If you wish to monitor all instructions within your program, you can issue:

```
PER Instruct Range 20000.500
```

There is no need to use the **ADSTOP** command first as is the case with **TRACE**. Only instructions in the range 20000-204FF are monitored.

When your program has been loaded and started, you will receive information at your terminal that might look like this:

```
020000 STM 90ECD00C 00DFAC CC=0
```

This line indicates that a STM instruction (located at address 20000) stored the registers beginning at location 00DFAC and that the condition code is now 0.

The QUERY command with the PER option can be used to determine what events are currently being traced. For example:

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

If in addition to instructions, you wish to trace instructions that alter registers, enter:

```
PER G Range 20000.500
```

The events that you are now monitoring (your current traceset) are:

```
Query PER
  1 INSTRUCT RANGE 020000-0204FF TERMINAL NORUN
  2 G RANGE 020000-0204FF TERMINAL NORUN
```

If you continue program execution by entering BEGIN you will receive information at your terminal that might look like this:

```
020004 BALR 05C0 000000 CC=0 G12=40020006
```

This line indicates that a BALR instruction at address 020004 changed register 12 to 40020006.

As with TRACE, it is possible to specify the printer and/or run for any event. To change *both* the instruction and register alteration (G) trace elements to *RUN* you can enter:

```
PER RUN
```

```
Query PER
  1 INSTRUCT RANGE 020000-0204FF TERMINAL RUN
  2 G RANGE 020000-0204FF TERMINAL RUN
```

To change *just* the instruction trace element to PRINTER, you can enter:

```
PER Instruct Range 020000-0204ff Printer RUN
```

```
Query PER
  1 INSTRUCT RANGE 020000-0204FF PRINTER RUN
  2 G RANGE 020000-0204FF TERMINAL RUN
```

Selectivity: You can use PER options to increase selectivity. Using PER, it is possible to limit tracing to a specific instruction or instructions. For example, to monitor only LR instructions (op code X'8'), enter:

```
PER Instruct DATA 18
```

CP Command Usage

When the NORUN option is in effect, program execution halts after each monitored event. When using the RUN option, program execution continues after each event. PER also has an execution rate between NORUN and RUN. This option is called STEP. STEP specifies the number of events that should be displayed before program execution halts and the CP command environment is entered. For example, to halt program execution after 5 instructions in the range 20000 to 204FF have been executed, enter:

```
PER Instruct Range 20000.500 Step 5
```

When your program has been loaded and started, you will receive information at your terminal that might look like this:

```
020000 STM 90ECD00C 00DFAC CC=0
020004 BALR 05C0 000000 CC=0
020006 ST 50DOC342 020348 CC=0
02000A LA 41DOC33E 020344 CC=0
020012 OC D607C3A2C3AA 0203A8 0204B0 CC=1
```

and then program execution would halt, and the CP command environment would be entered.

Although the STEP option allows you to step through your program more quickly without giving up all control, every monitored instruction is displayed. If many instructions are executed before the problem occurs, you will have to frequently clear your screen. However, you can change the frequency with which events are displayed by using the PASS option. Ordinarily, every successful event is displayed, but using the PASS option allows you to specify how many monitored events should be skipped before displaying one. For example, to skip the display of 100 instructions and display the 101st, enter:

```
PER Instruct Pass 100
```

Terminating PER: To end PER tracing with the current traceset, enter:

```
PER END Current
```

To end just the branch trace elements in the current traceset, enter:

```
PER END BRanch
```

It is not always desirable to end all the trace elements of a particular event type. When the current traceset is displayed using the QUERY PER command, each trace element is preceded by a number. This number can be used to selectively end PER tracing. For example:

```
Query PER
 1 INSTRUCT PRINTER RUN
 2 G5 DATA 0000023 TERMINAL NORUN
 3 G7 TERMINAL NORUN
 4 STORE INTO 020628-020630 TERMINAL NORUN
```

If you no longer want to monitor instructions that alter register 7 but want to continue monitoring register 5, enter:

```
PER END 3
```

Suspending PER: You can use the PER SAVE subcommand to save the current traceset. The saved traceset is saved only while the user is logged on.

To save the current traceset under the name TRACE1, enter:

```
PER SAVE trace1
```

The current traceset is still active. To suspend the PER tracing, enter:

```
PER END CURrent
```

PER tracing is no longer active. The traceset is saved under the name TRACE1. You can now execute normally, or create another current traceset using PER commands. This new traceset (or tracesets) can also be saved.

To resume PER tracing with the original traceset, enter:

```
PER GET trace1
```

A copy of traceset TRACE1 is still saved under the name of TRACE1. Changes made to TRACE1 while it is the current traceset have no effect on this saved copy.

To end the saved traceset TRACE1, enter:

```
PER ENd trace1
```

To terminate all PER tracing, enter:

```
PER ENd ALL
```

This ends the current and all saved tracesets.

The Branch Traceback Table: You can use TRACE to trace program interrupts. However, the trace information is displayed after the interrupt has occurred and cannot always be used to determine the cause of the problem. If you use PER in conjunction with TRACE, you can greatly reduce the difficulty of finding the cause of the problem. If the problem is an operation exception, it may have been caused by a bad branch instruction.

The first step is to trace program interrupts using TRACE:

```
TRace PROg
```

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

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


CP Command Usage

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

```
PER BRanch 24600
```

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

```
PER BRanch 245F0-24600
```

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

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

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

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

```
TRace PROg
```

followed by:

```
PER Instruct Range 20000.500 (assuming the program is 500 bytes in length)
```

and then:

```
PER Count
```

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

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

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

```
PER PASS 2144
```

the response is:

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

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

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

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

```
PER PASS
```

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

```
1 INSTRUCT RANGE 02000-0204FF TERMINAL NORUN
```

You can now trace the last 15 instructions, and use the DISPLAY command to display storage and register contents.

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

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

```
PER Instruct 0ACA Range 20000.500
PER Count
```

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

CP Command Usage

```
Query PER
  1 INSTRUCT OACA RANGE 020000-0204FF TERMINAL NORUN
    PER COUNT 623
```

You can then trace the program after using the PER PASS option as above to get close to the problem.

The PER Command Option: You can use the PER CMD option to execute any CP command (except SLEEP) whenever a particular event occurs. For example:

```
PER Instruct Range 20000.500 run
PER STore 204F0-204FF Range 20000.500 RUN Cmd Display 204F0-204FF
```

traces the execution of every instruction in the range 20000 through 204FF and displays the contents of storage at 204F0 through 204FF every time any storage within the range 204F0-204FF is altered by an instruction in the range 20000-204FF.

Also, you can use the CMD option to cause execution of a program to continue at a specific address whenever a particular event occurs. For example:

```
PER Instruct Range 20000.500 Printer
PER BRanch 0 RUN Cmd Begin 24f28
```

causes program execution to continue at location 24F28 whenever a branch to location 0 occurs. The execution continues after the instruction is displayed. If, when program execution is resumed at location 24F28, a subsequent branch to zero occurs, execution again begins at location 24F28. This can result in a loop. You can also use the CMD option to prevent this. For example, if LINEDIT is on, and the escape character is set to " and the line end character is #, then:

```
PER Instruct 20000.500 Printer
PER BRanch 0 RUN Cmd PER END BRanch"#Begin 24f28
```

turns off the branch trace element and causes program execution to continue at location 24F28 after the instruction is displayed. The current traceset would then be:

```
1 INSTRUCT RANGE 020000-0204FF PRINTER RUN
```

The commands associated with each trace element are executed whenever the event described by the trace element occurs. The commands are executed in the order in which they appear in the traceset. Therefore, if the current traceset is:

```
1 INSTRUCT TERMINAL RUN CMD cmd1#cmd2
2 BRANCH TERMINAL RUN CMD cmd3
3 G TERMINAL RUN CMD cmd4#cmd5
```

and an instruction is executed that alters a register, but does not cause a successful branch, then the CP commands cmd1, cmd2, cmd4, and cmd5 are executed (in that order).

Note: If a CP command is entered while commands are being executed by PER, the output from the commands may be interleaved.

Once the command option has been specified for a particular trace element, the command option remains in effect until the trace element is turned off. However, the command can be changed. For example, if the current traceset contains:

```
1 BRANCH TERMINAL RUN CMD DISPLAY G15
2 G TERMINAL RUN CMD DISPLAY PSW
```

the command associated with the branch trace element can be changed to DISPLAY G14-15 by issuing:

```
PER BRanch TERminal RUN Cmd Display G14-15
```

or both commands can be changed to DISPLAY G14-15 by issuing:

```
PER Cmd Display G14-15
```

Storage Alteration: You can use PER to trace the alteration of storage in the user's virtual machine. If PER STORE is specified, then whenever an instruction places a value into storage, that event is traced. It is *not* necessary that this value be *different* from the previous value. *stored in memory*

You can also monitor the alteration of storage to a specific value. For example:

```
PER STore INTO 20100 DATA 112757
```

monitors instructions that cause the storage at location 20100 to become 112757. Note that these instructions are traced even if the value at location 20100 was already 112757 before the execution of the instructions.

You can also monitor only alterations of storage when the storage value actually changes. For example:

```
PER Mask INTO 20100 DATA ffffffffffffffff
```

monitors instructions that actually change one or more bytes in the doubleword field starting at location 20100. You can also use PER MASK to monitor changes to specific bits in storage. For example:

```
PER Mask INTO 20100 DATA 8040
```

monitors instructions that change the status of the first bit in the byte at location 20100 or the second bit in the byte at location 20101.

GUESTR AND GUESTV: PER traces virtual machine activity in both second- and third-level storage. If you use its GUESTR and GUESTV options, you may limit the trace output to activity either on second- or third-level. Second-level storage is storage that appears real to your virtual machine. To limit your PER trace to this level of storage, use the GUESTR or DATOFF operand. Third-level storage is storage that appears virtual to your virtual machine. To limit your PER trace to this level of storage, use the GUESTV or DAT operand.

CP Command Usage

CP itself runs with dynamic address translation (DAT) off. When CP dispatches a virtual machine, the DAT bit in the PSW is on. Therefore, if CP is IPLed into a virtual machine, then that CP is executing in second-level storage. When CMS is IPLed on top of the second-level CP, then that CMS is executing in third-level storage.

If you issue the command:

```
PER I Range 20000-21000
```

then both second- and third-level activity is traced (that is, both CP and CMS).

To trace only the CP activity (second-level) in the range 20000-21000, enter:

```
PER I Range 20000-21000 GUESTR
```

To trace only the CMS activity (third-level) in the range 20000-21000, enter:

```
PER I Range 20000-21000 GUESTV
```

You may also trace selectively specific activity in both second- and third-level storage. For example, to trace successful branches and storage alterations of location X'024' in second-level storage and to trace branches to location X'20000', and alterations to register 7 in third-level storage, enter:

```
PER BRanch STore INTO 1024 GUESTR  
PER BRanch INTO 20000 G7 GUESTV
```

Part 2. Reference Information

Part 2 contains reference information. It consists of two sections:

“Section 4. How to Use Command Syntax Diagrams” describes the syntax that is used in this publication to display command formats.

“Section 5. Format of CP Commands” contains reference data for all the class G and Any commands in alphabetical sequence by command name. The command line format includes all operands and a description of each. Any limits or restrictions applicable to the values of variable operands are also defined.

The following commands and subcommands exist in the CP, CMS, and XEDIT environments:

CP

QUERY

SET.

The following command exists in the CP and XEDIT environments:

RESET.

Section 4. How to Use Command Syntax Diagrams

The notation used to define the command syntax is:

- Truncations and Abbreviations of Commands and Operands

Where truncation of a command name is permitted, the shortest acceptable version of the command is represented by uppercase letters. (Remember, however, that VM/SP HPO commands can be entered with any combination of uppercase and lowercase letters.)

The following examples show the format specifications as used:

Display INDicate DISConn

The format for Display means that D, DI, DIS, DISP, DISPL, DISPLA, and DISPLAY can be used. The format for INDicate means that IND, INDI, INDIC, INDICA, INDICAT, and INDICATE can be typed. The format for DISConn means that DISC, DISCO, DISCON, and DISCONN can be used. Each of the examples given are shown with the valid specifications for the command name.

Operands are specified in the same manner. Where truncation is permitted, the shortest acceptable version of the operand is represented by uppercase letters in the command format box. If no minimum truncation is noted, the entire word (represented by all capital letters) must be entered.

Abbreviations are shorter forms of command names and operands. Abbreviations for command names are shown below the full name in the format box. Abbreviations for operands are shown in the description of the individual operands that follows the format box. For example, the operand READER has both a minimum truncation and an abbreviation. In the format box, it is shown as:

Reader

indicating that the minimum truncation is R. In the discussion of the READER operand that follows the format block, it is shown as:

READER
RDR

indicating that the abbreviation is RDR. Thus, the acceptable specifications for the READER operand are: R, RE, REA, READ, READE, READER, and RDR.

- Lowercase letters, words, and symbols that appear in the command format box represent variables for which specific information should be substituted. For example, "hexloc" indicates that a hexadecimal storage location address is to be entered.

How to Use Command Syntax Diagrams

The following lowercase letters and symbols appear in the diagrams in this book:

***dumpid**

is a line of user input identifying the dump. It can contain up to 100 characters including imbedded blanks and asterisks.

cuu

is a three-digit channel address. The *c* represents a one-digit channel address. The first *u* represents the control unit. The second *u* represents the device address.

dsname

is a 1-to-24 character field suitable for specifying OS or DOS files.

fn

is the file name.

ft

is the file type.

hexloc

is a hexadecimal address.

msgtext

is the text of the message to be transmitted.

n

is one digit. For each *n* you can use one digit. If the syntax diagram uses *nnnn*, the maximum length of the number is four digits.

raddr

is the address of a real I/O device. The form of the address is *cuu* (see description above).

reg

is a register number.

resid

is a four-digit resource identification specified in the real I/O configuration module (DMKRIO).

spoolid

is a spool identification number that is automatically assigned by CP when the file is closed.

userid

is a user identification number, a 1-to-8 character alphanumeric symbol identifying each terminal user.

vaddr

is a virtual device address. Its form is *cuu*. (See description above.)

valid

is the volume identification label for a disk.

- The following symbols are used to define the command format and should never be typed when the actual command is entered.

underscore	
braces	{ }
brackets	[]
ellipsis	...

- Uppercase letters and words, and the following symbols, should be entered as specified in the format box.

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

- Choices are represented in the command format boxes by stacking.

A
B
C

- An underscore indicates an assumed default. If an underscored choice is desired, it need not be specified when the command is entered.

Example

The representation

A
B
C

indicates that either A, B, or C may be selected. However, if B is selected, it need not be specified. Or, if none is entered, B is assumed.

How to Use Command Syntax Diagrams

- The use of braces denotes choices, one of which *must* be selected.

Example

The representation

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

indicates that you *must* specify either A, or B, or C. If a list of choices is enclosed by neither brackets or braces, it is to be treated as if enclosed by braces.

- The use of brackets denotes choices, one of which *may* be selected.

Example

The representation

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

indicates that you may enter A, B, or C, or you may omit the field.

- In instances where there are nested braces or brackets on the text lines, the following rule applies; nested operand selection is dependent upon the selection of the operand of a higher level of nesting.

Example

Nesting levels are represented as follows:

```
[Level 1 [Level 2 [Level 3 ]]]
```

Where 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. As noted, our rule states: the higher level of nesting must be selected in order to select the next level (lower level) operand.

- An ellipsis indicates that the preceding item or group of items may be coded more than once in succession.

Example

The representation

```
spoolid1 spoolid2...
```

indicates that more than one spoolid may be coded.

Section 5. Format of CP Commands

The command formats are described in alphabetical order except for special characters such as the asterisk (*) and the pound sign (#), which appear first.

The command formats are presented in the following order:

- **Command Name:** Identifies the name of the command. The name is also included at the top of the page for easy reference.
- **Privilege Class:** States the privilege class of the command as Any or G.
- **Description of Function:** Describes the use.
- **Syntax:** Lists the syntax of the command with all the possible operands that you can use.
- **Operand Description:** Describes the function of each operand and any values that you can include in the operand.
- **Usage Notes** Contains notes about special uses of the command, its operands, or combinations of commands or operands.
- **Responses** Describes the CP responses sent to the terminal, caused by execution of the command. These responses are command responses and are not to be construed as system messages. The command responses are not prefixed and, therefore, they are not contained in *VM/SP HPO System Messages and Codes*. Only CP command responses applicable to the class G and class Any user are contained in this publication.

*

*

Privilege Class: Any

Use the asterisk (*) to annotate the terminal console sheet or terminal display screen data with a comment. This commentary also appears in the virtual console spool file (if the console spooling function is invoked for the virtual machine). The format of the * (comment) command is:

*	anycomment
---	------------

Responses

None.

#CP

Privilege Class: Any

Use the #CP command to execute a CP command while in a virtual machine command environment without first signaling attention to get to the CP command environment.

The format of the #CP command is:

#CP	[commandline1	[#commandline2 #...]]
-----	---------------	-----------------------

where:

commandline

specifies the name and operands for the CP command or commands you want to issue. You must precede the first commandline with at least one blank.

Usage Notes

1. The pound sign (#) shown in the preceding format represents the logical line end symbol and is the default supplied by VM/SP HPO. However, your installation may wish to redefine the logical line end symbol. If you have redefined the logical line end symbol, #CP is an invalid command; you must substitute your line end symbol for the pound sign when using this command. To find out your current line end character, issue the CP command QUERY TERMINAL described on page 165.
2. For the command to operate, the following conditions must be met:
 - The virtual machine must be operating with SET LINEDIT ON (a default).
 - The first three characters of the edited line must be #CP (uppercase or lowercase) with the “#” representing the logical line end symbol currently defined.
 - At least one blank must separate the #CP from any command line. Do not use attention interruption in any part of the line or to enter the line.
 - If in a virtual machine command environment, the virtual machine must have issued a READ for at least four bytes of data. If the READ is for less than three bytes, you must use the attention key to enter the CP environment.
3. You can enter multiple command lines as operands of the #CP command provided that (a) the virtual machine’s console READ is for at least the number of bytes required to enter the entire line and (b) all command lines to be executed must be separated by the current logical line end symbol.

4. If you enter #CP with no operands, the virtual machine enters the CP environment. CP cancels the virtual machine's console READ by returning a unit exception status for the virtual console. The virtual operating system then reissues the console READ to allow you to key in the appropriate response to a previous message from that machine's operating system.
5. If you issue the TERMINAL APL ON command on a keyboard equipped with APL characters, and then issue a #CP command during your APL session, CP does not translate the #CP command. When APL is ON, CP uses a different set of translation tables and does not recognize the pound sign (#) as a logical line end symbol.

Command	System Action
#CP	An attention interrupt is reflected to your virtual machine; your virtual machine enters CP environment.
#CP query files	QUERY command executed.
#CP query files#query users	Two separate QUERY commands executed.
data enteredϕ#CP msg op is tape available	"Data entered" is ignored. You send a message to the operator.
#CP data entered	You enter CP environment and CP interprets "data entered" as an invalid operand.
data enteredϕ#CP	"Data entered" is ignored. You enter CP environment.
#CP query filesϕdata entered	QUERY command is not executed; console input (data entered) passes to the virtual machine.

Figure 5. System Action to #CP Command

Responses

If you enter #CP without a command line, you receive this message:

CP

If you enter #CP with command lines, you receive the responses appropriate to the individual commands you entered.

ADSTOP

Privilege Class: G

Use the ADSTOP command to halt the execution of a virtual machine at a virtual instruction address. Execution halts when the instruction at the address specified in the command is the next instruction to be executed. Only one ADSTOP command can be in effect at a given time. If a second ADSTOP command is issued, the previous ADSTOP setting is canceled.

The format of the ADSTOP command is:

ADSTOP	$\left. \begin{array}{l} \text{hexloc} \\ \text{Off} \end{array} \right\}$
--------	--

where:

hexloc

is the six-character hexadecimal representation of the virtual instruction address where execution is to be halted.

OFF

cancels any previous ADSTOP setting.

Usage Notes

1. When execution halts, the CP command mode is entered and a message is displayed. At this point, you may invoke other CP debugging commands. To resume operation of the virtual machine, issue the BEGIN command. Once an ADSTOP location is set, it is turned off by the following:
 - Reaching the virtual storage location specified in the ADSTOP command
 - Performing a virtual IPL or SYSTEM RESET
 - Issuing the ADSTOP OFF command
 - Specifying a different location with a new ADSTOP *hexloc* command.
2. Since the ADSTOP function modifies storage by placing a CP SVC X'B3' at the specified location, you should not:
 - Designate an address within a shared segment. If a shared segment is modified by an ADSTOP command, a warning message is issued to the user and the virtual machine is unshared from (given a private copy of) the named system. Other users of the segment continue to operate with an unchanged copy of that segment.
 - Use the TRACE command with the INSTRUCT, BRANCH, or ALL operands when a traced instruction is located at the ADSTOP address.

3. Address stop (ADSTOP *hexloc*) is specified as a six-character virtual machine real address. For OS/VS or DOS/VS, the specified address must be in fixed storage. Movement of the page containing the ADSTOP can cause erroneous processing because the resulting SVC X'B3' is presented to the virtual machine for handling.
4. If the SVC handling portion of the virtual machine assist feature is enabled on your virtual machine, CP turns it off when an ADSTOP is set. When the address stop is removed, CP returns the assist feature SVC handling to its previous status.
5. If you plan to issue an ADSTOP command and you have issued a SET RUN ON command during your terminal session, you must first issue a SET RUN OFF command. This command allows the virtual machine to halt execution at the address specified on the ADSTOP command line. When you logon, RUN is set off.
6. You cannot issue the ADSTOP command if you are a V=R user running in single processor mode.
7. You cannot issue the ADSTOP command if you are a V=R user using the preferred machine assist feature whether you are using control switch assist or not using it.
8. When you are using the Group Control System (GCS), an ADSTOP at an address within the common GCS area can change the instruction at that address and affect the entire group. Therefore, use the PER command (described on "PER" on page 133) to debug rather than ADSTOP or TRACE.

Response

ADSTOP AT *nnnnnn*

The instruction whose address is *nnnnnn* is the next instruction scheduled for execution. The virtual machine is in a stopped state. Any CP command (including an ADSTOP command to set the next address stop) can be issued. Enter the CP command BEGIN to resume execution at the instruction location *nnnnnn*, or at any other location desired.

ATTN

Privilege Class: G

Use the ATTN command to make an attention interruption pending at your virtual console.

The format of the ATTN command is:

ATTN	
------	--

Usage Notes

1. The REQUEST command performs the same function as ATTN; the two commands can be used interchangeably.
2. The BEGIN command is not required after you issue ATTN.
3. CP ignores the ATTN and REQUEST commands if an I/O operation is in progress at the console or if other interruptions are pending for your virtual machine. This condition may result if you issue the ATTN or REQUEST command by means of DIAGNOSE X'8'.

Responses

None.

BEGIN

BEGIN

Privilege Class: G

Use the BEGIN command to continue or resume execution in the virtual machine at either a specified storage location or the location pointed to by the virtual machine's current program status word (PSW).

The format of the BEGIN command is:

Begin	[hexloc]
-------	----------

where:

hexloc

is the hexadecimal storage location where execution is to begin.

Usage Notes

1. When BEGIN is issued without *hexloc*, execution begins at the storage address pointed to by the current virtual machine PSW. Unless the PSW has been altered since the CP command mode was entered, the location stored in the PSW is the location where the virtual machine stopped.
2. When BEGIN is issued with a storage location specified, execution begins at the specified storage location. The specified address replaces the instruction address in the PSW, then the PSW is loaded.

Responses

None. The virtual machine begins execution.

CHANGE

Privilege Class: G

Use the CHANGE command to alter one or more of the external attributes of a closed spool file or files.

The format of the CHANGE command is:

CHange	$\left\{ \begin{array}{l} \text{[Reader]} \\ \text{[RDR]} \\ \text{[Printer]} \\ \text{[PRT]} \\ \text{[Punch]} \\ \text{[PCH]} \end{array} \right\} \left\{ \begin{array}{l} \text{Class c1} \\ \text{spoolid} \\ \text{FORM form1} \\ \text{ALL} \end{array} \right\} \left\{ \begin{array}{l} \text{Class c2} \\ \text{[HOLD]} \\ \text{[NOHOLD]} \\ \text{Dist distcode} \\ \text{COpy [*] nnn} \\ \text{FLash[name nnn]} \end{array} \right\} \left\{ \begin{array}{l} \text{[Name \{fn (ft)\}} \\ \text{\{dsname\}} \end{array} \right\}$
	$\left\{ \begin{array}{l} \text{MODify name [n]} \\ \text{CHars name1 [name2 [name3 [name4]]]} \\ \text{CHars name1 [CHars name2 [CHars} \\ \text{name3 [CHars name4]]]} \\ \text{FCB name} \\ \text{FORM form2} \end{array} \right\}$
<p>¹ One of these operands must be chosen; however, more than one may be specified. They may be combined in any order on the command line. NAME, if specified, must be the last entry in the command line. This is contrary to the notation normally used in this publication.</p>	

where:

[READER]
[RDR]

changes reader spool files.

[PRINTER]
[PRT]

changes printer spool files.

[PUNCH]
[PCH]

changes punch spool files.

CLASS c1

designates an existing class. The class, c1, is a one-character alphameric field from A through Z or from 0 through 9. Refer to the *VM/SP HPO Operator's Guide* for a detailed description of spool classes.

spoolid

is the *spoolid* number of the file that is to be changed.

FORM form1

is a one-to-eight character alphanumeric form name. Use the *form* name to select those files you want to change. All *form* names are defined by your installation at system generation time.

ALL

changes all your spool files.

CLASS c2

changes the spool class of the file to c2.

HOLD

prevents the file from being printed, punched, or read until it is released. The file is released when the CHANGE command is issued with the NOHOLD operand specified.

NOHOLD

releases the specified file from HOLD status.

DIST distcode

changes the distribution code specified in the directory to the *distcode* specified on the command line, for the specified file only. The distribution code appears on the output separators of the printer and punch output; it has no effect on reader files.

COPY [[*] nnn

specifies the number of copies of the file you want spooled to the virtual output device. The option is valid for printer and punch files only. The value of nnn (number of copies) must be a number from 1 through 255. Leading zeroes are optional. The optional asterisk (*) can be specified on files to be spooled to the 3800 printer. The asterisk (*) forces the 3800 printer to perform the file duplication internally one page at a time. This option avoids retransmission of the print file to the virtual output device every time a copy is desired.

FLASH name nnn

signifies that a forms overlay contained in the 3800 printer be superimposed onto certain pages of the output. The name, one to four characters, identifies the forms overlay to be used in the projection. The number nnn can be anything from 0 to 255. The number specifies that the first nnn copies of output are to be superimposed. For example, if you specify COPY 4 FLASH name 2, the first 2 copies of your file are flashed, while the last two copies remain unflashed. The default, 0, indicates no superimposing is desired.

MODIFY name [n]

assigns a copy modification module to be used when printing a file on the 3800. This function makes it possible to alter text by either preventing printing of certain information or adding labels to output. The name must be one to four characters. If the name is omitted, the file prints normally without modification. n selects one of the

character arrangement tables specified with the CHARS keyword to be used for the copy modification text. The values 0, 1, 2, or 3 correspond to the order in which the table names have been specified. If no n is specified, the first character arrangement table is assumed.

CHars name1 [name2 [name3 [name4]]]

CHars name1 [CHars name2 [CHars name3 [CHars name4]]]

specifies the character arrangement table used when printing a file. The name must be from one to four characters with a maximum of four names given per command. Note: The multiple use of 'CHars' need only be used if 'nameN' would conflict with an option name.

FCB name

controls the vertical spacing of output on a page. The name must be from one to four characters.

FORM form2

changes the spool form name of the file to form2.

NAME fn [ft]

assigns identification to the spool file in the CMS format filename and filetype. The field, fn, is a one- to eight-character alphameric filename assigned to the file for identification. The field, ft, is a one- to eight-character alphameric filetype assigned to the file for identification. If ft is not specified, the filetype is set to blanks.

NAME dsname

assigns identification to the spool file in a non-CMS format. The field, dsname, is a 1- to 24-character field suitable for specifying OS or DOS files (for example, SYS1.SYSLIB.MYMAC).

Usage Notes

1. Issue the QUERY command to determine the current attributes of the file.
2. If you want to change an output file, the file must have been closed but not yet selected for printing or punching.
3. You can change an input (READER) file at any time before it is opened; that is, before CP issues the first read for the file.
4. Use the QUERY command to verify the new attributes of the file or files.
5. Your system programmer can define certain spool classes to automatically print certain classification titles (for, example, TOP PRIORITY or CONFIDENTIAL). The title prints on the separator page, and, optionally, at the bottom of each printed page. Ask your system programmer if he has defined certain classes for this function, or request that classes be defined. For example, you may wish class P to print TOP PRIORITY on the separator page, and class X to print CONFIDENTIAL on the separator page and also on the bottom of each page of output.

CHANGE

6. In order to print on the 3800, the name you specify in the FLASH option must match the FLASH name the operator specifies in the START command, unless the FLASH number nnn is zero. If you specify a zero for the FLASH number or you omit the FLASH option, a file still prints on the 3800 as long as the spool file class matches the print class designated by the installation for the 3800. If you fail to meet these requirements, CP submits the spool file to another printer and the special 3800 print options are ignored.
7. When a spool file containing special 3800 options is spooled to a printer other than the 3800, the options are ignored and printing occurs normally. Options available only on the 3800 Model 3 (for example, 10 lines per inch FCB) may not be used on the 3800 Model 1 printer.
8. You can nullify any previous or existing settings of the CHARS, MODIFY, or FCB options by specifying NULL in the name field of these operands.
9. An FCB name of 6, 8, or 12 avoids an FCB forms mismatch. These special FCB specifications print the entire spool file at 6, 8, 12 or (for the 3800 Model 3 only) 10 lines per inch regardless of the paper size. VM/SP HPO calculates the proper FCB form to load.
10. When the operator activates the 3800 printer, the operator specifies the character arrangement table and the FCB value that prints the separator page for every output file. If you omit the CHARS option on a file spooled to the 3800, the printing defaults to the character arrangement table previously defined by the operator. This is also the case for the FCB option.
11. VM/SP HPO issues a message to you whenever a file spooled to the 3800 printer contains invalid or undefined option names. If the operator has not automatically purged the file, you can correct the situation using the CHANGE command or spool the file to another printer.
12. Customized 3800 Model 1 character sets are not interchangeable with the 3800 Model 3 character sets. Users may recode customized 3800 Model 1 character sets and build new modules through the use of the GENIMAGE command. See *VM/SP HPO Planning Guide and Reference* and *VM/SP HPO Operator's Guide* for this command. Use the MVS character Conversion Aid to convert existing customized character sets to the 3800 Model 3 PEL density.

Response

{nnn} FILES CHANGED
{NO }

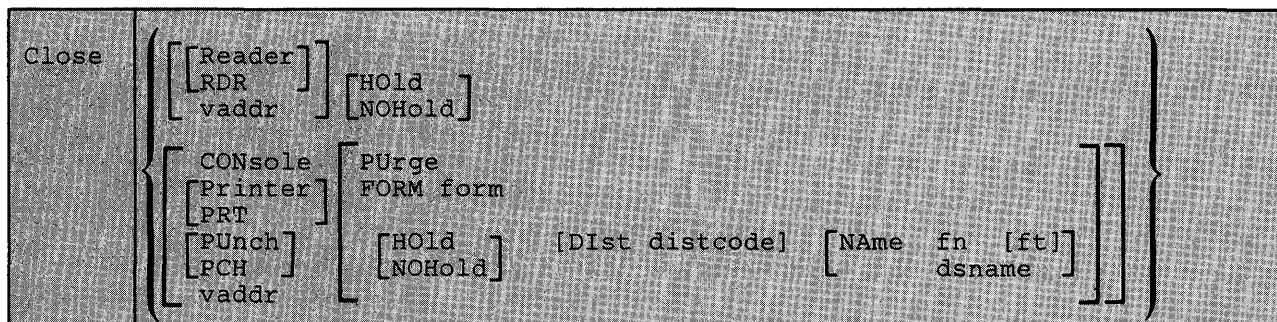
This is the response when you issue the CHANGE command. This is an indication of the number of files changed. It does not reflect individual alterations to a given file. This message does not appear if you have issued the CP SET IMSG OFF command line.

CLOSE

Privilege Class: G

Use the CLOSE command to terminate the spooling activity on any virtual spooled unit record or console device.

The format of the CLOSE command is:



where:

**[READER
RDR]**

closes all reader spool files.

CONSOLE

closes your virtual machine's console spool file. Once a virtual console spool file is closed, it becomes a printer spool file and can be manipulated in the same way as any printer spool file (for example, it can be purged or changed).

**[PRINTER
PRT]**

closes all printer spool files.

**[PUNCH
PCH]**

closes all punch spool files.

vaddr

is the virtual address (*cuu*) of the device to be closed. The address may represent a reader, console, printer, or punch.

HOLD

makes the output spool file being closed unavailable for further processing, until it is specifically requested or changed. However, input reader files are still available for processing (see Figure 6) This option, specified in the CLOSE command, overrides any previously specified HOLD or NOHOLD option for the files being closed.

NOHOLD

makes the output spool file being closed available for further processing, but the input reader file is no longer available for processing (see Figure 6). Specify NOHOLD if a HOLD established by the SPOOL command is still in effect and the current active file is not to be held.

You can release one of your own output files in HOLD status by using the CHANGE command. If an output file is spooled for another user (SPOOL FOR *userid*), only the receiving virtual machine user can change the file status. If an output file is spooled to another user as an input file (SPOOL TO *userid*), the HOLD option places the input file in HOLD status. The file then cannot be read by the virtual machine until it is changed to NOHOLD by the receiving virtual machine user.

If an input file is closed with the HOLD option, the file is saved and not purged from the system. The saved file is available for virtual machine and user processing and is not placed in a HOLD status. Input spool files that are closed are normally purged from the virtual machine.

PURGE

closes and immediately purges from the virtual machine the output spool files. No output file is produced.

FORM form

specifies the form name for a file. The form operand is a one- to eight-character alphanumeric form name defined by your installation during system generation. If FORM is not specified for the CLOSE command, the file created has the form associated with the last SPOOL command applied to it.

DIST distcode

uses the one- to eight-character alphanumeric identification (*distcode*) on the output separators of printer and punch instead of the identification specified in the directory. The distribution code is changed for this file only and does not affect other files or change the directory. If the file is transferred to another user, this option has no effect.

NAME fn [ft]

assigns identification to the spool file in the CMS format filename and filetype. The filename, *fn*, is a one- to eight-character alphanumeric name assigned to the file for identification. The filetype, *ft*, is a one- to eight-character alphanumeric type assigned to the file for identification. If *ft* is not specified, the filetype is set to blanks.

NAME dsname

assigns identification to the spool file in a non-CMS format. The *dsname* field is a 1- to 24-character field suitable for specifying OS or DOS files (for example, SYS1.SYSLIB.MYMAC). Only 18 characters of the 24-character *dsname* are displayed by QUERY, even though a name of up to 24 characters is valid.

Usage Notes

1. If the file is an input reader file, the file being processed is purged unless SPOOL READER HOLD was previously specified (see the SPOOL command). The effect of HOLD or NOHOLD for a particular file established by the SPOOL command can be overridden by specifying NOHOLD or HOLD, respectively, in the CLOSE command.
2. If an input spool file is read but the read is not completed (that is, the virtual machine does not get a last-card indication), you must issue a CLOSE READER command in order to be able to read that file again (or to read any other file). Unless you specify HOLD when you close a reader file, the file is purged.
3. If the file is an output file on a printer, punch, or console, the file is either queued for output on a real unit record device, or, if the virtual output device is transferred (by use of the SPOOL vaddr TO userid command), the file is queued for input to the receiving user.
4. You can specify a filename and filetype and an optional distribution code to aid in later identification of the file and its contents.
5. Figure 6 shows what happens to an input file in your virtual reader when the CLOSE command is issued with the HOLD or NOHOLD operand in effect. The CLOSE command can result in an input file being purged or saved.
6. Figure 7 shows what happens to a file in your virtual output device when the CLOSE command is issued with the HOLD, NOHOLD, CONT, and NOCONT operands in effect.
7. When you CLOSE a virtual 3800 printer that was not SPOOLED continuous (CONT), any attributes that were set via the SPOOL or TAG command are preserved for use with the next print file. In general, attributes such as CHAR and FCB set via the CMS SETPRT command, or loaded by a CCW from the virtual machine, are discarded after the print file is closed. An exception to this is that an FCB specified as 6, 8, or 12 on the CMS SETPRT command is preserved.

(CLOSE Command Setting)	SPOOL Command Options Set for a Virtual Input Device	
	NOHOLD	HOLD
Normal EOF (default CLOSE)	File purged	File saved
CLOSE	File purged	File saved
CLOSE HOLD	File saved	File saved
CLOSE NOHOLD	File purged	File purged

Note: The term "saved" means that the file is neither purged nor is it in HOLD status. A subsequent READ could read this file.

Figure 6. Action Taken on CLOSE for a Virtual Input Device

CLOSE

(CLOSE Command Setting)	SPOOL Command Options Set for a Virtual Output Device			
	NOHOLD NOCONT	HOLD NOCONT	NOHOLD CONT	HOLD CONT
CLOSE	File released for processing	File held	No action ¹	No action ¹
CLOSE HOLD	File held	File held	No action ¹	No action ¹
CLOSE NOHOLD	File released for processing	File released for processing	No action ¹	No action ¹

¹No action is taken until you issue a SPOOL...PRT|PUN NOCONT command. Then you can issue a CLOSE command to release the file for processing.

Figure 7. Action Taken on CLOSE for a Virtual Output Device

Responses

$\left\{ \begin{array}{l} \text{PRT} \\ \text{PUN} \\ \text{CON} \end{array} \right\}$ FILE spoolid $\left\{ \begin{array}{l} \text{TO} \\ \text{FOR} \end{array} \right\}$ userid COPY nnn $\left\{ \begin{array}{l} \text{HOLD} \\ \text{NOHOLD} \end{array} \right\}$

This response is received if: multiple copies of the file are being processed, the file is being transferred to another user, or the file is placed in a HOLD status. However, if the SET MSG option is OFF, this message is not issued even though the preceding conditions are met.

COMMANDS

Privilege Class: Any

Use the COMMANDS command to list the commands and diagnose codes you are authorized to use.

The format of the COMMANDS command is:

```
COMMANDS
```

Usage Note

Some commands belong to more than one functional type, but the virtual machine may not be authorized to use all the types of a particular command. If the system rejects a command that is listed in this response, it is probably because the virtual machine is not authorized to use the functional type (that includes the operands you entered.) Try the command again with the operand for one of the other functional types.

Responses

```
COMM1  COMM2  COMM3  COMM4  COMM5  COMM6  
COMM7  ....
```

```
DIAG1  DIAG2  DIAG3  DIAG4  DIAG5  DIAG6  
DIAG7  ....
```

COMM is the name of a command you are authorized to use.

DIAG identifies a diagnose code you are authorized to use.

COUPLE

Privilege Class: G

Use the COUPLE command to connect your virtual nondedicated channel-to-channel device (channel-to-channel adapter or 3088) to another user's virtual device of the same type, or to another one of your own virtual devices of the same type.

The format of the COUPLE command is:

```
COUPLE      vaddr1 [To] userid vaddr2
```

where:

vaddr1

is the virtual address (*cuu*) of your channel-to-channel device.

[TO] userid

is the user identification of the virtual machine to which *vaddr1* is to be connected. If *vaddr1* is to be connected to your own virtual machine, *userid* may be specified as an asterisk (*). The user must be logged on and have a virtual channel-to-channel device defined. If the keyword TO is omitted, the *userid* cannot be "T" or "TO".

vaddr2

is the virtual address (*cuu*) of the channel-to-channel device to be connected to *vaddr1*.

Responses

CTC vaddr1 COUPLE TO userid vaddr2

This is the response you receive when you issue the COUPLE command.

vaddr1

is the address of your channel-to-channel device.

userid

is the identification of the receiving virtual machine.

vaddr2

is the address of the channel-to-channel device of the receiving user (or a different channel-to-channel device in your own virtual machine).

CTC vaddr2 COUPLE BY userid vaddr1

This is the response that the user specified by *userid* receives when you issue the COUPLE command.

vaddr2

is the address of the channel-to-channel device of the receiving user.

userid

is the identification of the user who issued the COUPLE command.

vaddr1

is the address of the issuing user's channel-to-channel device.

CTC vaddr1 DROP FROM userid vaddr2

This is the response that the user who issued the COUPLE command receives if the virtual CTC *vaddr1* was already coupled when the COUPLE command was issued. The previous connection is terminated. This response is always followed by the response:

```
CTC vaddr1 COUPLE TO userid vaddr2
```

Usage Note

A 3088 channel-to-channel device cannot be coupled to a CTCA.

Privilege Class: Any

Use the CP command if you are a CMS user and want to execute a CP command when a CMS read has been issued to the console, without first signaling attention to get to the CP environment. This statement operates similarly when the console is in the CP environment.

The format of the CP command is:

```
CP [commandline1 [#commandline2 #...]]
```

where:

commandline

specifies the name and operands for the CP command or commands you want to issue. You must precede the first command line with at least one blank.

Usage Notes

1. The pound sign (#) shown in the format block represents the logical line end symbol currently in effect for your virtual machine. If you have redefined the logical line end symbol, you must substitute your line end symbol for the pound sign when using this command. To determine your current line end character, issue the CP command QUERY TERMINAL described on page 165.
2. The CP command is treated as a null line by the control program and therefore can precede any other command if one or more blanks separate CP from the other command. The CP command is useful because it lets the CMS user enter commands without having to know which command environment (CP or virtual machine) he is in.
3. You can enter multiple command lines as operands of the CP command, but you must separate each command line by the logical line end symbol (#). If you enter only CP with no operands, the virtual machine enters the CP environment. CP cancels the virtual machine's console READ by returning a unit exception status for the virtual console. The virtual operating system then reissues the console READ to allow you to key in the appropriate response to a previous message from that machine's operating system.
4. CP will not interrogate an attention key if you press it while in fullscreen mode (user mode) but will pass it on to the fullscreen application for processing. Therefore, to execute a CP command while your virtual machine is in fullscreen mode (user mode), first use the PA1 key or the break key to put the virtual machine into 'CP READ' mode.

Responses

If you enter CP without a command line, you receive this message:

CP

If you enter CP with command lines, you receive the responses appropriate to the individual commands you entered.

DEFINE

DEFINE

Privilege Class: G

Use the DEFINE command to alter your virtual machine configuration or channel operating mode.

The format of the DEFINE command is:

```
DEFine [Reader]
       [RDR]
       [Printer]
       [PRT]
       [PUnch] [As] vaddr
       [PCH]
       CONsole
       CTCa
       TIMer
       1403
       1443
       2501
       2540P
       2540R
       3088
       3203
       3211
       3262
       3289E
       3505
       3525
       4245
       4248

       3800 [As] vaddr [Size ww ll] [4Wcgm] [CFS] [NODatck]
       3800-1 [2Wcgm] [BTS] [Datck]
       3800-3

       [Printer] [As] vaddr [1403]
       [PTR] [1443]
       [3203]
       [3211]
       [3262]
       [3289E]
       [3800]
       [3800-1]
       [3800-3]
       [4245]
       [4248]
```

```

CHANnels [As] {SEL}
                {BMX}

Line      [As] vaddr [IBM[1]]
                [TELE[2]]

GRAF      vaddr [3036]
                [3066]
                [3138]
                [3148]
                [3158]
                [3270]

vaddr1 [As] vaddr2
}
TFB-512
T3310 [As vaddr BLK[nnnnnn]
T3370
}
T2305
T2314
T2319 [As vaddr [CYL] [nnn]
T3330
T3340
T3350
T3375
T3380

STORage [As] {nnnnnK}
                {nnM}

```

where:

[READER] [AS] vaddr
RDR

adds a spooling 2540 card reader with the address specified by *vaddr* to the virtual machine configuration.

[PRINTER] [AS] vaddr
PRT

adds a spooling 1403 printer with the address specified by *vaddr* to the virtual machine configuration.

[PUNCH] [AS] vaddr
PCH

adds a spooling 2540 card punch with the address specified by *vaddr* to the virtual machine configuration.

CONSOLE [AS] vaddr

adds a virtual system console to the virtual machine at the address specified by *vaddr*.

CTCA [AS] vaddr

adds a virtual channel-to-channel adapter with the address specified by *vaddr* to the virtual machine configuration.

TIMER [AS] vaddr

adds a pseudo timing device with the address specified by *vaddr* to the virtual machine configuration. To check the status of this a pseudo timing device, use the QUERY VIRTUAL UR command, described on pages 159 and 172.

1403 [AS] vaddr

adds a spooling 1403 printer with the address specified by *vaddr* to the virtual machine configuration.

1443 [AS] vaddr

adds a spooling 1443 printer with the address specified by *vaddr* to the virtual machine configuration.

2501 [AS] vaddr

adds a spooling 2501 card reader with the address specified by *vaddr* to the virtual machine configuration.

2540P [AS] vaddr

adds a spooling 2540 card punch with the address specified by *vaddr* to the virtual machine configuration.

2540R [AS] vaddr

adds a spooling 2540 card reader with the address specified by *vaddr* to the virtual machine configuration.

3088 [AS] vaddr

adds a virtual 3088 channel-to-channel device with the address specified by *vaddr* to the virtual machine configuration. The virtual 3088 does not support the online testing functions.

3203 [AS] vaddr

adds a spooling 3203 printer with the address specified by *vaddr* to the virtual machine configuration. The virtual 3203 printer supports the LOADVFCB command, but not the Index feature.

3211 [AS] vaddr

adds a spooling 3211 printer with the address specified by *vaddr* to the virtual machine configuration. The virtual 3211 printer supports LOADVFCB and the Index feature (right-index only).

3262 [AS] vaddr

adds a spooling 3262 printer to the virtual machine configuration; *vaddr* specifies the address to the virtual machine configuration. The virtual 3262 printer supports the LOADVFCB command but not the INDEX feature. If the INDEX feature is specified, CP processing ignores it.

3289E [AS] vaddr

adds a spooling 3289E printer; *vaddr* specifies the address to the virtual machine configuration. The virtual 3289E printer supports the LOADVFCB command but the INDEX feature is not supported. If the INDEX feature is specified, CP processing rejects it.

3505 [AS] vaddr

adds a spooling 3505 card reader with the address specified by *vaddr* to the virtual machine configuration.

3525 [AS] vaddr

adds a spooling 3525 card punch with the address specified by *vaddr* to the virtual machine configuration.

3800

3800-1 [As] vaddr [Size ww ll]

**[2WCGM
4WCGM]**

**[BTS
CFS]**

**[DATck
NODatck]**

3800-3

adds a spooling 3800 printer Model 1 (3800-1) or 3800 Model 3 (3800-3) printer to the virtual machine configuration. If 3800 only is specified, the 3800 model 1 is assumed. The *vaddr* specifies the address to the virtual machine configuration. The *SIZE* keyword specifies the size of the forms that CP loads into the virtual 3800. The *ww* specifies the width of the form as a hexadecimal code (see usage note 8 on page 80 for the available form width codes.) The *ll* specifies the length of the paper in half inches expressed in decimal notation. For example, if you want 8 1/2" by 11" paper loaded into your virtual 3800, specify:

```
define 3800 as 00e size 04 22
```

If no width and length are specified, the default is 14 7/8 x 11 inches.

The user can also specify the number of Writable Character Generation Modules (WCGMs) that CP assumes for the virtual 3800. A 3800 Model 1 has a standard two WCGMs with the option of adding two more. A 3800 Model 3 has a standard four WCGMs. Each WCGM stores a raster pattern for 64 characters. If the user specifies 2WCGM for the 3800 Model 1, he has enough storage for CP to load 128 characters in the 3800; if he chooses the default, 4WCGM, he can load up to 255 characters.

The user also has the option of specifying which stacker the 3800 uses — either the Continuous Forms Stacker (CFS) which is the default, or the Burster-Trimmed-Stacker (BTS).

In addition, the virtual 3800 user can choose whether he wants data checks for the 3800 reflected to his virtual machine. If he specifies the *DATCK* keyword, CP reflects all data checks for the 3800 to the user's virtual machine if the Block Data Check CCW has not been issued. If the user specifies *NODATCK*, CP only reflects those data checks that are the result of invalid translate tables or no FCB code match being found. *NODATCK* is the default. Use of the *NODATCK* option will also suppress the reflection of some load checks that are the result of invalid translate tables.

Note: The user should only specify the DATCK keyword when it is absolutely necessary since it severely increases the overhead associated with the simulation of Write and Skip CCWs to the virtual 3800. In general, the reflection of data checks due to overprinting and invalid EBCDIC codes is unnecessary.

4245 [AS] vaddr

adds a spooling 4245 printer to the virtual machine configuration; *vaddr* specifies the address to the virtual machine configuration. The virtual 4245 printer supports the LOADVFCB command but not the INDEX feature. If the INDEX feature is specified, CP processing rejects it.

4248 [AS] vaddr

Adds a spooling 4248 printer to the virtual machine configuration; *vaddr* specifies the address to the virtual machine configuration. The virtual 4248 printer supports the LOADVFCB command but not the INDEX feature. If the INDEX feature is specified, CP processing rejects it.

CHANNELS [AS] {SEL } {BMX }

redefines the channel mode of operation for the virtual machine to either selector or block multiplexer. Use of the SEL (selector channel) or BMX (block multiplexer channel) operand sets the mode of operation for all channels except virtual channel 0. Channel 0 always operates in byte multiplexer mode.

Block multiplexer mode may enhance the virtual machine's operating system by allowing the overlap of Start I/O operations. This is done by reflecting a channel condition code of 0 back to the virtual machine rather than a channel busy signal.

Note: You immediately reset the virtual machine when you issue the CHANNELS [AS] SEL or CHANNELS [AS] BMX commands. You also reset preferred machine assist (if active).

LINE [AS] vaddr [IBM1] [TELE2]

adds a virtual 2701/2702/2703 communication line with the address specified by *vaddr* to the virtual machine configuration.

IBM1 indicates that an IBM-type terminal (2741, 1050, or equivalent) is on the 2701/2702/2703 line. TELE2 indicates that a teletypewriter is on the 2701/2702/2703 line.

GRAF vaddr [3036] [3066] [3138] [3148] [3158] [3270]

defines a temporary 3036, 3066, 3270, 3138, 3148, or 3158 (the console used on the IBM System/370 Model 138, 148, or 158) for the virtual machine. The *vaddr*, in the form *cuu*, is the hexadecimal virtual address for the device. After you define the 3036, 3066, 3270, 3138, 3148, or 3158, you must issue the CP DIAL command from the device in order to use it. The device must be supported by the virtual machine's operating system.

vaddr1 [AS] vaddr2

redefines the device represented by *vaddr1* as *vaddr2*. The virtual address, *vaddr1*, must represent a defined device in the virtual machine configuration. See usage note 7 on page 79 for more information on potential conflicts with the virtual control unit that would support the new device.

If *vaddr1* is a *dedicated 2305*, both *vaddr1* and *vaddr2* must be specified as base addresses (xx0 or xx8). Also, when you redefine a dedicated 2305, all eight exposure addresses are redefined. For example, if 1B0 is the base address of a dedicated 2305, the command

```
define 1b0 as 1d0
```

redefines addresses 1B0 through 1B7 as 1D0 through 1D7, respectively.

When you redefine a virtual unit record device CP closes all open output files. The closed output files are then available for further processing according to previous spool commands.

A virtual machine that has a Mass Storage System's MSC port and 3330 virtual volume (3330V) addresses dedicated to it must ensure that the 3330V addresses used by the virtual machine are the same addresses as those specified in the mass table create control statements and in the input to the MVS system generation process because the control program does not perform device mapping for MSC messages or orders.

If desired, an installation may choose to use the same mass table create input for the multiple real processor environment as well as the single processor environment running VM/SP HPO with multiple virtual processors (virtual machines). Mass table create permits definition of the same set of device addresses for more than one CPUID. One set of addresses can be redefined later under VM/SP HPO so that VM/SP HPO can run two or more MSS virtual machines on a single real machine.

Example

Mass table create control statements may define 3330V addresses 120 through 12F for CPUID A associated with one virtual machine user. The same set of addresses may be defined for CPUID B. VM/SP HPO, on the other hand, may have defined addresses 120 through 12F and 220 through 22F as separate devices in DMKRIO.

All virtual machines must address the 3330V devices using 120 through 12F since these are the addresses predefined in the mass table create program and by the MVS system generation process. By means of the DEDICATE statement in USER1's directory, the installation may choose to DEDICATE real addresses 120 through 12F as virtual addresses 120 through 12F. In a similar manner, the installation may choose to DEDICATE real addresses 220 through 22F to USER2 as virtual addresses 120 through 12F. The same could be accomplished by attaching the string of addresses and then redefining them by means of the DEFINE command.

$\left. \begin{array}{l} \text{T3310} \\ \text{T3370} \end{array} \right\}$	$\left. \begin{array}{l} \text{TFB-512} \\ \text{T3310} \\ \text{T3370} \end{array} \right\}$	[AS] vaddr [BLK] [nnnnnn]
$\left. \begin{array}{l} \text{T2305} \\ \text{T2314} \\ \text{T2319} \\ \text{T3330} \\ \text{T3340} \\ \text{T3350} \\ \text{T3375} \\ \text{T3380} \end{array} \right\}$	$\left. \begin{array}{l} \text{T2305} \\ \text{T2314} \\ \text{T2319} \\ \text{T3330} \\ \text{T3340} \\ \text{T3350} \\ \text{T3375} \\ \text{T3380} \end{array} \right\}$	[AS] vaddr [CYL] [nnn]

adds a temporary virtual disk of the specified type to the virtual machine configuration. See usage note 6 on page 79 and 2 on page 79.

CYL nnn specifies the number of cylinders that the disk contains for count-key-data devices. Specify T3350 if a 3350 is used in native mode; specify T3330 if a 3350 is used in 3330 compatibility mode. Specify T3340 if a 3344 is used.

BLK nnnnnn specifies the number of blocks for FB-512 devices. Specify TFB-512 if you do not care what specific fixed-block device holds the temporary disk (allowing VM/SP HPO to select either 3310 or 3370).

When you define temporary disk space on a 2305 device, only the *vaddr* specified is defined (not all eight addresses associated with the device).

Note: Temporary disk space is assigned from a pool of DASD resources; therefore, you should always format your temporary disk space before you use it. If your installation's system programmer specifies the SYSCLR= Yes option on the SYSRES macro, and you release temporary disk space or logoff the system, CP clears all data areas and directory areas in that temporary space. If your installation chooses not to implement the SYSCLR option, you should clear any temporary disk space yourself before you release it to free storage or logoff the system. This eliminates the possibility of an unauthorized person accessing your data either accidentally or deliberately.

STORAGE [AS] $\left\{ \begin{array}{l} \text{nnnnnK} \\ \text{nnM} \end{array} \right\}$

redefines the size of the virtual storage for the virtual machine as nnnnnK (where K represents 1024 bytes) or nnM (where M represents 1,048,576 bytes). The value specified becomes the new virtual storage size. Sizes must be in 4K increments and are limited by the maximum value in the virtual machine directory entry. The minimum size you can specify is 8K. All entries not specified in a 4K increment are rounded up to the next 4K boundary. Increasing or decreasing the virtual storage size causes a reset of the virtual system and, if active, preferred machine assist. This reset also clears virtual storage to binary zeros.

Usage Notes

1. You may redefine the machine's device configuration during a terminal session to increase the number of virtual devices. The maximum number of devices can be as large as 410. If the MAXDEV xxxx option is part of your directory's OPTION control statement, you may redefine the machine's device configuration during a terminal session to increase the number of virtual devices up to the option-defined limit that is greater than 410 (the maximum being the xxxx of the control statement). (You can increase the number of virtual devices only when there is enough contiguous space available in main storage.)

The 410 limit should not be exceeded unless necessary since it requires additional storage space.

This limit may be changed as a result of modifications made by the installation.

For a description of the OPTION control statement see the *VM/SP HPO Planning Guide and Reference*. The *VM/SP HPO System Programmer's Guide* discusses the details of this option.

2. When you alter the configuration or channel operating mode of your virtual machine using the DEFINE command, the changes are temporary and are in effect for the current terminal session only.
3. If either storage or channel operating mode is altered, the virtual machine is immediately reset and the IPL procedure must be performed again.
4. If storage is redefined, any pages that have been locked using the LOCK command become unlocked.
5. If your virtual machine is operating in basic control mode (ECMODE OFF), virtual addresses specified in *vaddr*, *vaddr1*, and *vaddr2* can be no higher than 5FF (channels 0 through 5). If it is operating in extended control mode (ECMODE ON), all addresses through FFF are valid.
6. When defining FBA virtual devices (TFB-512, T3310, and T3370) as temporary storage, you can possibly get a temporary disk larger than requested. To determine the actual size of the defined disk, issue the command "QUERY VIRTUAL DASD".
7. Use caution in defining the hexadecimal addresses of virtual devices (*cuu*) in DEFINE statements in order to avoid a usage conflict caused by control unit I/O interface protocol. The following is an example of a virtual machine's DEFINE statements that can cause operational conflict:

```
define 3211 as 102
define graf 103 3270
```


DEFINE

The virtual addresses of both the 3211 printer and the 3270 terminal indicate the use of the same channel (1) and control unit (0). By definition, the devices are virtual and therefore share one virtual control unit in CP. A real 3211 printer operates on a nonshared subchannel, and the real 3270 console is designed for shared subchannel operations. Both of these real devices are mapped to the same virtual control unit.

Defining a shared and nonshared device on the same virtual control unit will result in an error message (DMKDEF331E)

```
vaddr NOT DEFINED -  
USE A SHARED|NONSHARED VCU INSTEAD.
```

Therefore, when defining devices, make sure the devices are defined (and separated) within their own control unit range and not shared with other devices.

8. Figure 8 shows the form width codes (in hexadecimal notation) that are available for the 3800 as a virtual printer. A virtual 3800 accepts form width code sizes for common use sizes (inches) or ISO (millimeters). Note that the default value is X'0F'.

X'01'	6-1/2 in.	(165 mm ISO)
X'02'	Reserved	(180 mm ISO)
X'03'	Reserved	
X'04'	8-1/2 in.	(215 mm ISO)
X'05'	Reserved	
X'06'	9-1/2 in.	(235 mm ISO)
X'07'	9-7/8 in.	(250 mm ISO)
X'08'	10-5/8 in.	(270 mm ISO)
X'09'	11 in.	(280 mm ISO)
X'0A'	12 in.	(305 mm ISO)
X'0B'	Reserved	(322 mm ISO)
X'0C'	Reserved	
X'0D'	13-5/8 in.	(340 mm ISO)
X'0E'	14-3/10 in.	(363 mm ISO)
X'0F'	14-7/8 in.	(378 mm ISO)

Figure 8. Available Form Width Codes

9. If you issue the DEFINE device [as] vaddr command while working with preferred machine assist, you must satisfy the virtual device restrictions for the preferred machine assist. This is true whether you are using control switch assist or not using it.

These restrictions are that a V=R user using preferred machine assist:

- a. May not define a T-disk
- b. Must ensure the following when assigning a new address to a dedicated device or full extent virtual disk:
 - 1) The virtual and real addresses are equal or there is no real device control block (RDEVBLK) specified for the virtual address; and
 - 2) The virtual channel address is a CP-owned channel
- c. Must use nondedicated devices (excluding full extent virtual disks) for which:
 - 1) There is no real device control block (RDEVBLK) for the virtual address; and
 - 2) The virtual channel address is a CP-owned channel

Note: When you are following these restrictions for defining a full extent virtual disk, the real address is the address of the pack's physical location--not its current virtual address.

- 10. If the optional keyword "AS" is omitted, the virtual address assigned cannot be "A".
- 11. You cannot use the DEFINE command to change the virtual base address or paging addresses of a 3880-11 or 3880-21 Storage Subsystem device.
- 12. If your virtual machine goes into V=R recovery, devices defined by this command will not be retained. For V=R machines, do not redefine storage or other elements that can cause problems if V=R recovery resets them to their original value.
- 13. You cannot define a 3880 Storage Subsystem device as anything other than a subsystem device using this command. Specifically you cannot redefine it as a CMS minidisk.
- 14. To find the status of the virtual devices that have been defined, use the QUERY [VIRTUAL] ALL command.
- 15. When using the virtual channel-to-channel adapter, you can receive a spurious attention interrupt after receiving attention and busy in response to a data transfer operation. The spurious attention can occur if both sides of the VCTCA are doing the same data transfer operation, such as both doing writes or both doing reads.

DEFINE

Responses

Responses are generated to confirm that the desired configuration change has taken place. These responses do not appear on your terminal if you have issued the CP SET IMSG OFF command line.

type vaddr DEFINED

where the possible values for type are:

Type	Meaning
------	---------

DASD	Direct access storage device/3330V(virtual volumes)
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
MSC	Mass Storage Control

CHANNELS = {SEL }
 {BMX }

is the channel mode of operation for the virtual machine. This response applies to all channels except channel 0 (always a byte multiplexer channel).

STORAGE = {nnnnnK }
 {nnM }

verifies the new storage size of your virtual machine. The minimum storage you may specify is 8K.

DETACH

Privilege Class: G

Use the DETACH command to remove a virtual device from the virtual machine. The CHANNEL operand detaches a dedicated channel from your virtual machine when that channel path is no longer required for virtual machine I/O operations.

The format of the DETACH command is:

DETach	{ vaddr [vaddr...] vaddr-vaddr CHANnel c }
--------	---

where:

vaddr [vaddr...]

vaddr-vaddr

is the virtual address or addresses (*cuu*) of the device to be detached from your virtual machine. To detach multiple devices on one command line, you must insert a blank between the addresses specified. To detach a range of devices, you must insert a hyphen (-) between the addresses specified. Be careful not to mix multiple addresses and a range of addresses on the same command line.

CHANNEL *c*

is the real address of the channel that you want to detach.

Usage Notes

1. You can detach a previously attached device even if the device is currently in use. You can also detach devices that were attached because of directory entries or during CP system initialization.
2. When you detach a virtual device, it becomes inaccessible to your virtual machine. If the device was previously attached to your virtual machine by an ATTACH command, it is released and becomes available for attachment to your virtual machine, another user, or the CP system. Tape devices are automatically rewound and unloaded when detached.
3. If you detach a device that was previously attached to your virtual machine by the operator, a message is sent to the operator informing him that the device is free.
4. If the device being detached is a dedicated 2305, the virtual address you specify must be a base address (xx0 or xx8).

5. If a detach operation is taking place for a 2305 device and a range is being used, processing for all 2305 exposures is performed when the base address is encountered. It is possible that the range operand may not include all the 2305 exposures. This is not an error situation; all exposures are detached. When multiple real device addresses are specified on the command line, only the base address for the 2305 should be used. Any device address other than the base exposure is considered an error condition.
6. Command processing does not terminate if an error (other than a syntax error) is encountered when processing a range of addresses or multiple addresses. Instead, an error message is issued for each device (address) that cannot be detached. In addition, an informational message identifies those devices that were successfully detached. Informational messages may appear several times in one command execution when a range of addresses or multiple addresses are being processed. Multiple informational messages appear only when range is being processed. You may specify a maximum of 48 multiple addresses on the DETACH command line. If you attempt to detach more than 48 multiple addresses, only the first 48 will be processed and the remaining addresses will be ignored with no error message issued for these device addresses.
7. It is possible for a class B user to DETACH real devices from another user or other users.
8. If you issue the CHANNEL c operand before all I/O operations are complete, those operations are aborted.

Responses

The DETACH command belongs to both the B and the G privilege class of user; the response or responses issued are dependent upon who issues the DETACH command. Depending upon the situation, responses are issued to:

- The general user whose virtual machine has the device to be detached
- The system operator informing him that the device is successfully detached
- Another B privilege class operator (not the system operator) that may have initiated the detach operation.

This publication documents only those responses that the general user would receive in the course of a virtual machine terminal session. For those responses that are received by the system operator as a result of a general users detach operation, refer to the DETACH command in the *VM/SP HPO Operator's Guide*.

Note: The 'type' variable in the following responses is omitted when a range of addresses or multiple addresses are being reflected in the response.

type vaddr DETACHED

vaddr...

vaddr-vaddr

The user receives one or more of these responses upon the completion of a successful detach operation that was self initiated.

type vaddr DETACHED BY operator

vaddr...

vaddr-vaddr

The user receives one or more of these responses upon the completion of the detach operation if the system operator or another user with the B privilege class had initiated the detach operation.

vaddr pagaddr1 pagaddr2 pagaddr3 DETACHED

The user receives this response when a 3880 Storage Subsystem device is detached after the user has initiated the detach operation. The virtual base address and virtual paging addresses of each detached device are shown.

vaddr pagaddr1 pagaddr2 pagaddr3 DETACHED BY operator

The user receives this response when a 3880 storage control device is detached by the operator. The virtual base address and virtual paging addresses of each detached device are shown.

CTC vaddr DROP FROM userid vaddr

This is the response if the device detached was a virtual channel-to-channel device connected (using the COUPLE command) to another virtual channel-to-channel device of the same type on the virtual machine specified by the *userid*. This response is always followed by the response:

CTC vaddr DETACHED

CHANNEL c DETACHED userid

is sent to the user who issued the command and to the primary system operator.

In the above responses, 'type' is one of the following:

Type	Meaning
DASD	Direct access storage device/3330V(virtual)volumes
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
DEV	Any other device
MSC	Mass Storage Control unit

DIAL

Privilege Class: Any

Use the DIAL command to logically connect a switched line, leased line, locally attached, or remote Binary Synchronous (BSC) terminal to a previously logged-on multiple-access virtual machine.

The format of the DIAL command is:

```
DIAL      userid  [vaddr]
```

where:

userid

is the identification of a virtual machine that is currently logged on.

vaddr

is the address of the virtual communication line to which the connection is to be made. If *vaddr* is not specified, the lowest GRAF address defined by the *userid* will be its connection to the virtual communication line.

Usage Notes

1. A DIAL command is accepted only at logon, and only as a substitute for a LOGON command. The type of terminal used must be supported by both VM/SP HPO and the multiple-access virtual machine. See the *VM/SP Terminal Reference* for details on running and gaining access to multiple-access machines.
2. The DIAL command is not supported for the 3066 system console, for the 3210/3215 system console, or for terminals that are using NCP lines in a 3704/3705/3725 control unit, but is supported for 3138, 3148, and 3158 display consoles.
3. Once the connection is made, your terminal operates entirely under the control of that virtual machine. The DIAL command matches your terminal to an equivalent type defined in the multiple-access virtual machine. If no matching terminal type exists, the connection cannot be made and an error message is issued.
4. The DIAL command supports both local non-SNA and remote BSC 3270 display devices. If you issue the DIAL command from either of these, the virtual system user must issue the CP command RESET to drop the dialed connection. In addition, there must be enough space on the screen to display the DIAL request and its resulting message. If enough space is not available, the message DMKDIA017E will be issued and the command will terminate. The user should then clear the screen and reissue the command.

5. The DIAL command is not supported for SNA terminals logically connected to the host VM/SP HPO system using the VTAM Communications Network Application (VM/VCNA) program product.

Support is provided for VM SNA Console Services (VSCS) graphics device users to access logged on guest virtual machines through use of the DIAL command. DIALED to machines that use a "write for position" operation will receive an OPCHECK.

"Write for position" is a write CCW where the data is a start buffer address with a command chained Read CCW.

6. The maximum amount of data that can be transferred from a DIALED remote 3270 is 65,503 bytes.
7. The DIAL command does not reset a 3278-Mod 4 to default mode (a 24 line screen) after the system has switched the terminal to alternate mode (a 43 line screen).

Responses

DIALED TO userid vaddr

is the message sent to the user indicating that a logical connection has been made. The term *vaddr* is a three-digit virtual device address in the form *cuu*.

$$\left. \begin{array}{l} \text{GRAF } raddr \\ \text{LINE } raddr \\ \text{DEV } resid \end{array} \right\} \text{DIALED TO userid DIALED} = nnn$$

is the response to the primary system operator, indicating a successful connection to the virtual machine (*userid*) and the total number of lines (*nnn*) currently connected to other virtual machines. The term *raddr* in this response is a three-digit real device address in the form *cuu*. DEV *resid* indicates the resource identification of a 3704/3705/3725 line.

Note: The terminal remains connected to and under the control of the virtual machine until that virtual machine terminates the communication.

DROP FROM userid vaddr

is the message sent to the user when the line is disabled. The term *vaddr* is a three-digit virtual device address in the form *cuu*.

$$\left. \begin{array}{l} \text{GRAF } raddr \\ \text{LINE } raddr \\ \text{DEV } resid \end{array} \right\} \text{DROP FROM userid DIALED} = nnn$$

is the message sent to the primary system operator when the line is disabled. The term *raddr* in this response is a three-digit real device address in the form *cuu*. DEV *resid* indicates the resource identification of a 3704/3705/3725 line.

COMMAND REJECTED; DEVICE *raddr* PENDING OFFLINE

is the message sent when a command has been issued that would cause the device *raddr* (*cuu*) to be activated. Since this device is offline, the command is rejected.

DISCONN

Privilege Class: Any

Use the DISCONN command to disconnect your terminal from the system while the virtual machine continues operation. If you have a secondary *userid*, *secuser* specified in your directory entry, you can now use the console of that logged on secondary user as your virtual console after you disconnect.

The format of the DISCONN command is:

DISConn	[HOLD]
---------	--------

where:

HOLD

specifies that the communication line is not to be disabled. This option allows you to disconnect your terminal, and, at the same time, to avoid the process of telephone dialing into the system to access your virtual machine again.

Usage Notes

1. The terminal remains disconnected until it is reconnected using a LOGON command. The virtual machine is logged off 15 minutes after an attempt is made to read from the terminal or if the virtual machine goes into a disabled WAIT state. The user should be aware that certain operating systems frequently issue 'reads' to the console, which can cause the logoff time-out situation.
2. If your terminal connection is broken because of terminal, line, or teleprocessing control unit errors, CP places the virtual machine in disconnect mode for up to 15 minutes and your virtual machine does not continue to run. If you logon within 15 minutes, your virtual machine can continue operating. Since many operands of the SET and TERMINAL commands default to OFF when you logon again, you have to reissue the SET and TERMINAL commands. If you do not logon within the 15-minute interval, the virtual machine is logged off.
3. Unless the CP command, SPOOL CONSOLE START, is issued to spool the virtual console output, all "writes" or output messages to the virtual console are ignored. When the terminal is reconnected through the normal logon procedure, the terminal is placed in CP console function mode. To resume execution of the virtual machine, enter the BEGIN command.
4. If you have a secondary *userid* specified in your directory entry, but that *secuser* is not logged on or is running disconnected, you receive a message stating that the secondary user is unavailable.
5. The HOLD option of the DISCONN command is invalid if you are on a terminal running under control of the GCS or the VCNA interface.

6. Local console options, 'APL', 'ATTN', 'CONMODE 3270', 'HIGHLIGHT', 'LINESIZE', 'SCRNSAVE', 'SCROLL', 'TEXT', and 'TYPE 3101', for the **TERMINAL** command are session-valid options only. They are not supported across **DISCONNECT** of the virtual machine. You must reset these options when they are reconnected if you want to use they again.

Responses

When the **DISCONN** command is issued, the disconnect time message is issued.

DISCONNECT AT hh:mm:ss zone weekday mm/dd/yy
is the response to the user who issued the command.

$\left. \begin{array}{l} \text{GRAF } raddr \\ \text{LINE } raddr \\ \text{DEV } resid \end{array} \right\} \text{DISCONNECT userid USERS} = nnn$

is the response to the primary system operator informing him that the user represented by *userid* has been disconnected from the system. The response, *raddr*, is a three-digit real device address in the form *cuu*. The response, **USERS = nnn**, displays the total number of users remaining in the system. **DEV resid** indicates the resource identification of a 3704/3705/3725 line.

DISPLAY

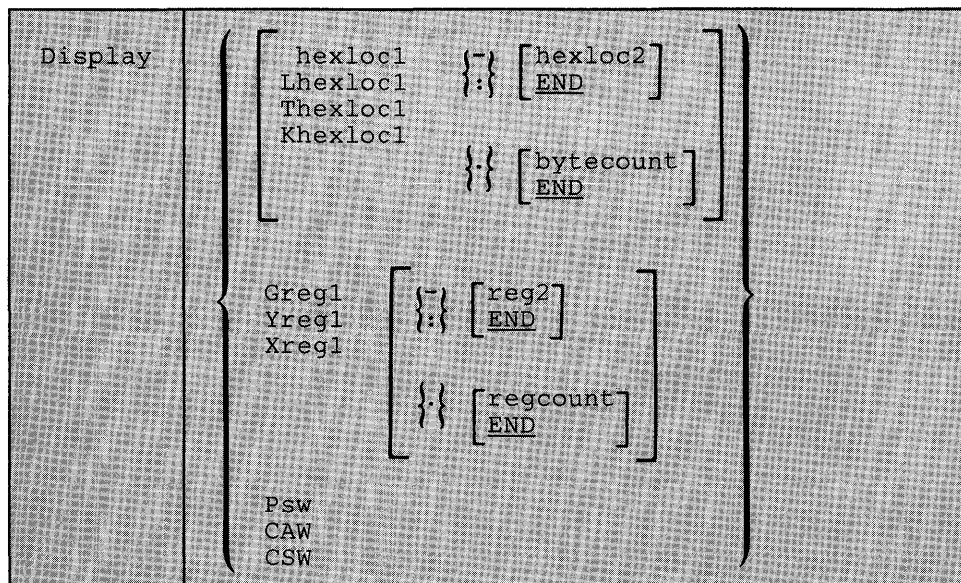
Privilege Class: G

Use the DISPLAY command to display the following virtual machine components at your terminal:

- Virtual storage locations (second-level virtual storage only; see usage note 1 on page 94.)
- Storage keys
- General registers
- Floating-point registers
- Control registers
- Program status word (PSW)
- Channel address word (CAW)
- Channel status word (CSW)

Note: Use the NETWORK DISPLAY command to display the contents of 3704/3705/3725 storage. This command is described in the (VM/SP HPO Operator's Guide).

The format of the DISPLAY command is:



where:

hexloc1
Lhexloc1
Thexloc1
Khexloc1

is the first, or only, hexadecimal storage location that is to be displayed at the terminal. If L or no letter prefix is specified, the storage contents are displayed in hexadecimal. If T is specified, the storage contents are displayed in hexadecimal, with EBCDIC translation beginning at the next lowest X'0' address. If K is specified, the storage keys are displayed in hexadecimal.

If *hexloc1* is not on a fullword boundary, it is rounded down to the next lower fullword.

If L, T, or K is entered either without any operands, or followed immediately by a blank, the contents of all storage locations or all the storage keys are displayed. The address, *hexloc1*, may be one to six hexadecimal digits; leading zeros are optional.

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

is the last of the range of hexadecimal storage locations whose contents are to be displayed at the terminal. Either a - or a : must be specified to display the contents of more than one location by storage address. If - or : is the first character of the operand, then *hexloc1* defaults to zero. If *hexloc2* is not specified, the contents of all storage locations from *hexloc1* to the end of virtual storage are displayed.

If specified, *hexloc2* must be equal to or greater than *hexloc1* and within the virtual storage size. (See usage note 8 on page 95 for a discussion on discontinuous saved segments.) The address, *hexloc2*, may be from one to six hexadecimal digits; leading zeros are optional.

If you wish to display the storage from the first storage location to the end, use END instead of *hexloc2*.

{ . } [**bytecount**]
 [**END**]

is a hexadecimal integer designating the number of bytes of storage (starting with the byte at *hexloc1*) to be displayed at the terminal. The period (.) must be specified to display the contents of more than one storage location by *bytecount*. The sum of *hexloc1* and *bytecount* must be an address that does not exceed the virtual machine size. (See usage note 8 on page 95 for a discussion on discontinuous saved segments.) If this address is not on a fullword boundary, it is rounded up to the next higher fullword. The value, *bytecount*, must be at least one and may have from one to six hexadecimal digits; leading zeros are optional.

If you wish to display the storage from the first storage location to the end, use END instead of *bytecount*.

Greg1

is a decimal number from 0 to 15 or a hexadecimal integer from 0 to F representing the first, or only, general register whose contents are to be displayed at the terminal. If G is specified without a register number, the contents of all the general registers are displayed at the terminal.

Yreg1

is an integer (0, 2, 4, or 6) representing the first, or only, floating-point register whose contents are to be displayed at the terminal. If Y is specified without a register number, the contents of all of the floating-point registers are displayed at the terminal.

Xreg1

is a decimal number from 0 to 15 or a hexadecimal number from 0 to F representing the first, or only, control register whose contents are to be displayed at the terminal. If X is specified without a register number, the contents of all of the control registers are displayed at the terminal. If Xreg1 is specified for a virtual machine without extended mode operations available, only control register 0 is displayed.

{-} [reg2]
{:} **END**

is a number representing the last register whose contents are to be displayed at the terminal. The dash (-) or colon (:) following reg1 is used to specify a range of register values to be displayed. If reg2 is not specified, the contents of all registers from reg1 through the last register of this type are displayed. The operand, reg2, must be equal to or greater than reg1. If Greg1 or Xreg1 is specified, reg2 may be a decimal number from 0 to 15 or a hexadecimal number from 0 to F. If Yreg1 is specified, reg2 may be 0, 2, 4, or 6. The contents of registers reg1 through reg2 are displayed at the terminal.

If you wish to display the storage from the first storage register to the end, use END instead of reg2.

{.} [regcount]
END

is a decimal number from 1 to 16 or a hexadecimal number from X'1' to X'10' specifying the number of registers (starting with reg1) whose contents are to be displayed at the terminal. If the display type G or X is specified, regcount can be a decimal number from 1 to 16 or a hexadecimal number from X'1' to X'10'. If display type Y is specified, regcount must be 1, 2, 3, or 4. When you display Greg or Xreg, the sum of reg1 and regcount must not exceed 16. When you display Yreg, regcount plus twice that regcount must not exceed 8.

If you wish to display the storage from the first storage register to the end, use END instead of reg2.

PSW

displays the current virtual machine PSW (program status word) as two hexadecimal words.

DISPLAY

CAW

displays the contents of the CAW (channel address word at hexadecimal location 48) as one hexadecimal word.

CSW

displays the contents of the CSW (channel status word at hexadecimal location 40) as two hexadecimal words.

Usage Notes

1. First-level storage is real main storage. Only second-level storage (storage that is real to the virtual machine) can be displayed. Operating systems running in a virtual machine such as DOS/VS and OS/VS have virtual storage of their own. This third-level virtual storage cannot be displayed directly. The user or the virtual operating system is responsible for converting any third-level storage locations to second-level storage locations before issuing the command.
2. If a command line with an invalid operand is entered, the DISPLAY command terminates when it encounters the invalid operand; however, any previous valid operands are processed before termination occurs. Multiple storage locations, registers, and control words can be displayed using a single command line.
3. When multiple operands are entered on a line for location or register displays, the default display type is the same as the previous explicit display type. The explicit specification of a display type defines the default for subsequent operands for the current display function. Blanks are used to separate operands or sets of operands if more than one operand is entered on the same command line. If a dash or a colon is omitted between the *hexloc1* and *hexloc2* operands, and the two operands are separated by one or more blanks, CP only displays the storage contents at those two specific addresses. Blanks must not be used to the right or left of the range or length delimiters (: or - or .), unless the user wants the default value of the missing operand defined by the blank. For example:

```
Display 10 20 T40 80 G12 5 L60-100
```

displays the following, respectively:

```
hexadecimal location 10
hexadecimal location 20
hexadecimal location 40 with EBCDIC translation
hexadecimal location 80 with EBCDIC translation
general register 12
general register 5
hexadecimal locations 60 through 100
```

4. To terminate the DISPLAY function while data is being displayed at the terminal, press the attention key (or its equivalent). When the display terminates, another command may be entered.

5. The DISPLAY command does not distinguish between shared and nonshared storage; it displays any of the virtual machine's addressable storage whether shared or not.
6. Use the DISPLAY command to display the contents of various storage locations, registers, and control words at the terminal. By examining this type of information during the program's execution, you may be able to determine the cause of program errors. Usually, an address stop is set to stop the program execution at a specified point. The system enters the CP environment and you may then issue the DISPLAY command.
7. When you must examine large portions of storage, use the DUMP command rather than the DISPLAY command. Because the terminal operates at a much slower speed than the printer, only limited amounts of storage should be printed (using the DISPLAY command) at the terminal.
8. When running with a discontinuous saved segment (DCSS), you can display storage locations outside the range of your virtual machine size if they are within the DCSS. If there exist locations between the upper limit of your virtual machine and the address at which the DCSS was saved, an attempt to display those locations (or associated keys) results in a nonaddressable storage message.
9. If you are running MVS/SP Version 1 Release 3 or a later release in your virtual machine, you may be using storage greater than 16 Mb. However, you cannot issue the DISPLAY command to show this additional storage.

Responses

One or more of the following responses is displayed, depending upon the operands specified.

Displaying Storage Locations

xxxxxx word1 word2 word3 word4 [key] *EBCDIC TRANSLATION*

This is the response you receive when you display storage locations; xxxxxx is the hexadecimal storage location of word1. Word1 is displayed (word-aligned) for a single location specification. Up to four words are displayed on a line, followed, optionally, by an EBCDIC translation of those four words. Periods represent nonprintable characters. Multiple lines are used (if required) for a range of locations. If translation to EBCDIC is requested (*Thexloc*), alignment is made to the next lower 16-byte boundary; otherwise, alignment is made to the next lower fullword boundary. If the location is at a 2K page boundary, the key for that page is also displayed.

Displaying Storage Keys

xxxxxx TO xxxxxx KEY = kk

This is the response you receive when you display storage keys; xxxxxx is a storage location and kk is the associated storage key.

Displaying General Registers

GPR n = genreg1 genreg2 genreg3 genreg4

This is the response you receive when you display general registers; n is the register whose contents are genreg1. The contents of the following consecutive registers are genreg2, genreg3, and so on. The contents of the registers are displayed in hexadecimal. Up to four registers per line are displayed for a range of registers. Multiple lines are displayed if required, with a maximum of four lines needed to display all 16 general registers.

Displaying Floating-Point Registers

FPR n = xxxxxxxxxxxxxxxx .xxxxxxxxxxxxxxxxxx E xx

This is the response you receive when you display floating-point registers; n is the even-number floating-point register whose contents are displayed on this line. The contents of the requested floating-point registers are displayed in both the internal hexadecimal format and the E format. One register is displayed per line. Multiple lines are displayed for a range of registers.

Displaying Control Registers

ECR n = ctlreg1 ctlreg2 ctlreg3 ctlreg4

This is the response you receive when you display control registers; n is the register whose contents are ctlreg1. The contents of the following consecutive registers are ctlreg2, ctlreg3, and so on. The contents of the requested control registers are displayed in hexadecimal. Up to four registers per line are displayed. Multiple lines are displayed if required.

Displaying the PSW

PSW = xxxxxxxx xxxxxxxx

The contents of the PSW are displayed in hexadecimal.

Displaying the CAW

CAW = xxxxxxxx

The contents of the CAW (hexadecimal location 48) are displayed in hexadecimal.

Displaying the CSW

CSW = xxxxxxxx xxxxxxxx

The contents of the CSW (hexadecimal location 40) are displayed in hexadecimal.

DUMP

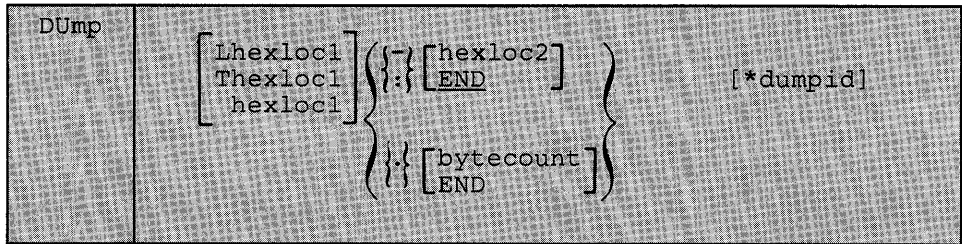
Privilege Class: G

Use the DUMP command to print the contents of various components of the virtual machine on the virtual spooled printer. The following items are printed:

- Virtual program status word (PSW)
- General registers
- Floating-point registers
- Control registers (if you have the ECMODE option specified in your directory entry)
- Storage keys
- Virtual storage locations (second-level virtual storage only; see usage note 1 on page 98)

Note: Use the NETWORK DUMP command to dump the contents of 3704/3705/3725 storage. This command is described in the VM/SP HPO Operator's Guide.

The format of the DUMP command is:



where:

- Lhexloc1**
- Thexloc1**
- hexloc1**

is the first or only hexadecimal storage location to be dumped. If you enter L or T without operands, the contents of all virtual storage locations are dumped. If L or no letter prefix is specified, the storage contents are displayed in hexadecimal. If T is specified, the storage contents are displayed in hexadecimal, with EBCDIC translation. The address, *hexloc1*, may be one-to-six hexadecimal digits; leading zeros are optional.

If *hexloc1* is not on a boundary divisible by 32, it is rounded down to the next lower such boundary.

$\{-\} \left[\begin{array}{l} \text{hexloc2} \\ \text{END} \end{array} \right]$

is the last hexadecimal storage location whose contents are to be dumped to the printer. If - or : is the first character of the operand, then *hexloc1* defaults to zero. The operand, *hexloc2*, must be equal to or greater than *hexloc1* and within the virtual storage size. To dump

to the end of storage, you can specify END instead of *hexloc2* or you can leave the field blank, since the default is END. If you specify :END or -END, the contents of storage from *hexloc1* to END are dumped. The contents of storage locations *hexloc1* through *hexloc2* are printed with EBCDIC translation at the printer. The operand, *hexloc2*, may be from one to six hexadecimal digits; leading zeros are optional.

{ . } [**bytecount**
END]

is a hexadecimal integer designating the number of bytes of storage (starting with the byte at *hexloc1*) to be dumped to the printer. The period (.) must be specified to dump the contents of more than one storage location by *bytecount*. The sum of *hexloc1* and *bytecount* must be an address that does not exceed the virtual machine size. If this address is not on a fullword boundary, it is rounded up to the next highest fullword. The value, *bytecount*, must be one or greater and can be no longer than six hexadecimal digits. Leading zeros are optional.

*dumpid

can be entered for descriptive purposes. If specified, it becomes the first line printed preceding the dump data. Up to 100 characters, with or without blanks, may be specified after the asterisk prefix. No error messages are issued if more than 100 characters are used including asterisks and embedded blanks.

Usage Notes

1. First-level storage is real main storage. Only second-level storage (storage that is real to the virtual machine) can be dumped. Operating systems running in a virtual machine such as DOS/VS, VSE operating systems, and OS/VS have virtual storage of their own. This third-level virtual storage cannot be dumped directly. The user or the virtual operating system is responsible for converting any third-level storage locations to second-level storage locations before issuing the command.
2. The CP DUMP command executes in an area of storage separate from your virtual machine storage and does not destroy any portion of your storage.
3. If the DUMP command is used in CMS mode, use the #CP DUMP command or depress the PA1 key before entering the command. This procedure maintains the virtual machine's registers for the dump. Otherwise, the registers are those of the CMS command processor.
4. The DUMP command prints the virtual PSW and the virtual registers (general, floating-point, and control). If only this information is desired, at least one virtual address must be specified, such as

```
DUMP 0
```

5. The output format for the virtual storage locations is eight words per line with the EBCDIC translation on the right. Each fullword consists of eight hexadecimal characters. All the rest of the information (PSW, general and floating-point registers, and storage keys) is printed in hexadecimal. If you have the ECMODE option in your virtual machine directory entry, the control registers are also printed. To print the dump on the real printer, a CLOSE command must be issued for the spooled virtual printer.
6. Normally, you should define beginning and ending dump locations in the following manner:

```
dump Lhexloc1-hexloc2
dump Lhexloc1.bytecount
dump Lhexloc1-hexloc2 hexloc1.bytecount * dumpid
```

If, however, a blank follows the type character (L or T) or the character and the *hexloc*, the default dump starting and ending locations are assumed to be the beginning and/or end of virtual storage. Blanks are used to separate operands or sets of operands if more than one operand is entered on the same command line. If a dash or a colon is omitted between the *hexloc1* and *hexloc2* operands, and the two operands are separated by one or more blanks, CP only dumps the storage contents at those two specific addresses. Blanks must not be used to the right or left of range or length delimiters (: or - or .), unless it is intended to take the default value of the missing operand defined by the blank. Thus, all of the following produce full storage dumps:

dump l	dump t:	dump 0-end
dump t	dump l.	dump l:end
dump -	dump t.	dump t:end
dump:	dump 0-	dump 0:end
dump .	dump 0:	dump l.end
dump l-	dump 0.	dump t.end
dump t-	dump l-end	dump 0.end
dump l:	dump t-end	

The following produces three full dumps:

```
dump l . t
dump - . :
```

7. When running with a discontinuous saved segment (DCSS), you can dump storage locations outside the range of your virtual machine size if they are within the DCSS. If there exist locations between the upper limit of your virtual machine and the address at which the DCSS was saved, an attempt to dump those locations (or associated keys) results in a "nonaddressable storage" message appearing in the printer output.
8. If you are running MVS/SP Version 1 Release 3 or a later release in your virtual machine, you may be using storage greater than 16 Mb. However, you cannot issue the DUMP command to dump this additional storage.

Responses

As the dump progresses, the following message is displayed at the terminal, indicating that the dump is continuing from the next 64k boundary:

DUMPING LOC *hexloc*

where *hexloc* is the segment (64K) boundary address for the dump continuation, such as 020000, 030000, or 040000.

If you press the attention key, or its equivalent, on the terminal while the message is being displayed, the dump function is terminated.

COMMAND COMPLETE

is the response indicating normal completion of the dump function.

ECHO

Privilege Class: G

Use the ECHO command to place the terminal in the echo environment. When in the echo environment, any input line entered is transmitted unchanged back to the terminal a specified number of times.

The format of the ECHO command is:

ECho	[nn <u>1</u>]
------	-------------------

where:

[nn
1]

is the number of times that the line is to be sent. The default is 1. An invalid entry (that is, one that is greater than 99 or contains nonnumeric characters) is treated as 1.

Usage Notes

1. When the specified number of lines has been displayed, another read to the terminal is issued to accept another data line.
2. To terminate the display before completion (for example, when you want to change the data line), press the attention key.
3. No line editing is done; the output line is the same as the input line and may contain any of the logical line editing symbols.

Responses

ECHO ENTERED; TO TERMINATE TEST, TYPE END

is displayed after the ECHO command is invoked to indicate that the echo environment has been entered.

ENTER LINE

is displayed to prompt for an input line to be entered. A reply of END returns the terminal to the CP command environment.

EXTERNAL

EXTERNAL

Privilege Class: G

Use the EXTERNAL command to simulate an external interruption to the virtual machine and return control to that machine.

The format of the EXTERNAL command is:

EXtErnal	[code] <u>40</u>
----------	---------------------

where:

code

40

is the interrupt code, a hexadecimal number to be associated with the external interrupt. All codes within the range X'01' through X'FF' are valid. If your virtual machine has the ECMODE option specified in the directory entry, you can also code X'1004' (clock comparator interruption) or X'1005' (CPU timer interruption). The default is the external interruption button on the system console, X'40'.

Usage Note

This simulates pressing the interruption key on the real computer console, or other functions which cause an external interruption. Control is given to the virtual machine immediately.

Responses

None. Since control is given to the virtual machine, any response is from virtual machine processing.

INDICATE

Privilege Class: G

Use the INDICATE command to display, at your terminal, the use of and contention for major system resources.

The format of the INDICATE command is:

INDicate	[LOAD USER]
----------	----------------

where:

LOAD

provides an indication of the operating load on the system by displaying values on: the number of users in queue 1 and queue 2, the usage of real storage, and the ratio of active users to users being serviced.

USER

displays the amounts of system resources used by your virtual machine in the current terminal session.

Usage Notes

1. Use INDICATE LOAD to display system load conditions.
2. Use INDICATE USER to display the total amount of certain resources used by your virtual machine during the current terminal session. Use the INDICATE USER command before and after the execution of a program to indicate the execution characteristics of that program in terms of the resources used.

Response for INDICATE LOAD

```
{CPU-nnn% [APU-nnn%] } Q1-nn Q2-nn STORAGE-nnn% EXPAN-nnn
{PROC xx-nnn% PROC yy-nnn%}
PAGING-nnnn/SEC STEAL-nnn% LOAD-nnn% SWAPPING-nnnn/SEC
```

where:

CPU-*nnn*%

(in AP and UP modes) *nnn* is a value rounded to the nearest whole number; it indicates the percentage of time that the main processor is executing.

APU-*nnn*%

(in AP mode) *nnn* is a value rounded to the nearest whole number; it indicates the percentage of time that the attached processor is executing. In uniprocessor mode of operation, the response field is zero.

PROC xx-nnn%

(in MP mode only) *nnn* is a value rounded to the nearest whole number; it indicates the percentage of time that the system is running on the IPL processor; *xx* represents the processor ID of the IPL processor. If the system was generated for MP operation, but it is currently executing in UP mode, PROC *xx* is the only value displayed (PROC *yy* is omitted from the response).

PROC yy-nnn%

(in MP mode only) *nnn* is a value rounded to the nearest whole number; it indicates the percentage of time that the system is running on the non-IPL processor; *yy* represents the processor ID of the non-IPL processor. If the system was generated for MP operation, but it is currently executing in UP mode, this value is not displayed.

Q1-nn Q2-nn

indicates the contention for CP. *nn* is a value rounded to the nearest whole number; it represents the numbers of users in queue 1 and queue 2. The values are maintained by the scheduler.

STORAGE-nnn%

is a measurement of the use of real storage. *nnn* is a value rounded to the nearest whole number and represents the ratio of the sum of the estimated working sets of the users in queue 1 and queue 2 (the runlist) to the number of pageable pages in the system, expressed as a percentage. Because the criterion for allowing a user on the eligible list to enter a queue is that at least 75 percent of his working set size must fit in the available page frames, the value of STORAGE can be more than 100 percent.

EXPAN-nnn

indicates the average elapsed time ratio for the contention of processor and storage resources. As EXPAN increases above the ratio of 1, there is a relative increase in the contention for these resources. For an explanation of the algorithm used, see the *VM/SP HPO System Programmer's Guide*.

SWAPPING-nnnn/SEC

is the smoothed average number of pages read and written per second by swapping.

PAGING-nnnn/SEC

is the smoothed average number of pages read and written by paging. It does not include pages read and written by swapping.

STEAL-nnn%

indicates the ratio of pages stolen to pages read by both swapping and paging, as expressed as a percentage.

LOAD-nnn%

is the estimated percentage of total elapsed time required to handle paging activity associated with reading pages that require stealing pages from in-queue users.

Response for INDICATE USER

PAGES: RES-nnnn WS-nnnn READS=nnnnnn WRITES=nnnnnn PG-nnnn PP-nnnn
VTIME=mmm:ss TTIME=mmm:ss RDR-nnnnnn PRT-nnnnnn PCH-nnnnnn
SWAPS: SWAPOUT=nnnnnn SWAPIN=nnnnnn SW-nnnn

where:

RES-nnnn

is the current number of your virtual storage pageable pages resident in main storage. This number is taken at an instant of time during the execution of the INDICATE command.

WS-nnnn

is the most recent system estimate of your projected working set size.

READS = nnnnnn

is the number of pages read from the DASD paging areas (defined as TYPE = PP, PG, PM, and PS on the SYSPAG macro) for your virtual machine. This counter may wrap around.

WRITES = nnnnnn

is the number of pages written to the DASD paging areas (defined as TYPE = PP, PG, PM, and PS on the SYSPAG macro) for your virtual machine. This counter may wrap around.

PG-nnnn

is the number of pages allocated for your virtual machine in the general paging areas, the page migration areas, and the spooling areas (defined as TYPE = PG, PM, and PS on the SYSPAG macro.)

PP-nnnn

is the number of pages allocated for your virtual machine in the preferred paging area (defined as TYPE = PP on the SYSPAG macro.)

VTIME = mmm:ss

is your total virtual machine time since you logged on or since the last ACNT command was issued for your virtual machine.

TTIME = mmm:ss

is your total virtual machine time plus total processor time (virtual plus overhead) you have used since you logged on or since you last issued an ACNT command for your virtual machine.

SIO = nnnnnn

is the total number of nonspoiled I/O requests you have issued since you logged on or since you last issued an ACNT command for your virtual machine. This counter may wrap around.

RDR-nnnnnn

is the total number of virtual cards read since you logged on or since the last ACNT command was issued for your virtual machine. This counter may wrap around.

PRT-nnnnnn

is the total number of virtual lines printed since you logged on or since the last ACNT command was issued for your virtual machine. This counter may wrap around.

PCH-nnnnnn

is the total number of virtual cards punched since you logged on or since the last ACNT command was issued for your virtual machine. This counter may wrap around.

SWAPOUT = nnnnnn

is the number of pages written to the DASD swapping area (defined as TYPE = SW on the SYSPAG macro) for your virtual machine.

SWAPIN = nnnnnn

is the number of pages read from the DASD swapping area (defined as TYPE = SW on the SYSPAG macro) for your virtual machine.

SW-nnnn

is the current number of pages allocated for your virtual machine in the swapping area (defined as TYPE = SW on the SYSPAG macro.)

IPL

Privilege Class: G

Use the IPL command to simulate an initial program load function for a virtual machine.

The format of the IPL command is:

Ipl	{	[vaddr cyln nnnnn]	[Clear NOClear]	[STOP ATTN PMA PMAV]	}	[PARM p1 [p2...]]
	{	systemname				

where:

[vaddr cyln] [CLEAR] [STOP]
 nnnnn NOCLEAR

simulates the IPL function when loading by device address.

vaddr

is the virtual address (*cuu*) of the device that contains the nucleus to be loaded.

cyln

is the cylinder containing the IPL data. When this operand is specified, CP loads the IPL data from the specified virtual cylinder instead of from virtual cylinder zero. Cyln defaults to virtual cylinder zero when no cylinder is specified. This operand is valid only for virtual direct storage devices and only if you do not specify systemname in the IPL command line.

nnnnn

is the block address containing the IPL data. When nnnnn is not specified, virtual block zero is the default. This operand is valid only for FB-512 devices and only if you do not specify a systemname in the IPL command line.

CLEAR

sets the virtual storage space to binary zeros before the operating system is loaded. This operand is invalid if you specify a systemname in the IPL command line.

NOCLEAR

allows the contents of your virtual storage space to remain unchanged prior to program load. This operand is invalid if you specify a systemname in the IPL command line. NOCLEAR is the default if you IPL by device (*vaddr*).

STOP

halts the virtual machine during the IPL procedure just before the initial PSW is loaded. It provides the virtual simulation of the IPL procedure for a real machine in instruction step mode. Issue SET RUN OFF to ensure that the stop occurs at the proper time. The

STOP

halts the virtual machine during the IPL procedure just before the initial PSW is loaded. It provides the virtual simulation of the IPL procedure for a real machine in instruction step mode. Issue SET RUN OFF to ensure that the stop occurs at the proper time. The STOP operand is invalid if PMA, PMAV, or systemname is specified in the IPL command. When the virtual machine stops, you can issue CP commands. For example, if you are loading OS or OS/VS into your virtual machine, you can use CP commands to store data into low storage, to load an alternate nucleus, or to alter the apparent size of virtual storage. To restart the virtual machine, issue the BEGIN command.

ATTN

generates an attention interrupt to the virtual machine during the IPL procedure. When the user specifies this parameter for an OS/VS1 IPL, the FASTNIP feature of OS/VS1 is automatically activated. Note that the ATTN operand is invalid if the user specifies the systemname, PMA, or PMAV operand on the IPL command.

PMA

enables the preferred machine assist feature for the V=R virtual machine running MVS/SP Release 1 with Release 1 Enhancement or subsequent releases. With this feature enabled, the V=R virtual machine (the preferred virtual machine) runs in real supervisor state, handles most of its own privileged operations and interruptions, and can use any storage unused by the system. This reduces CP overhead by performing some functions that ordinarily require simulation. If your processor is any but a 3033, it is capable of I/O interrupt support. If you are running MVS/SP Version 1 Release 3 or a later release, the preferred virtual machine may use storage greater than 16 Mb. However, you cannot use certain commands to service this additional storage (see the DISPLAY, DUMP, STORE, and VMDUMP commands).

In order to issue the IPL command with the PMA operand, you must meet the following requirements:

- Preferred machine assist must be installed on your 3033 or 308x processor
- The CP load list must include module DMKPMA
- You must be running in the V=R area
- You must set ECMODE ON
- PMA must be specified on the OPTION statement in your directory entry
- You must have affinity set to one processor if the system is running in AP/MP mode.

You cannot issue the IPL command with the PMA operand and either ATTN or STOP. These operands are mutually exclusive. You cannot specify the PMA operand more than once on the same command line. You cannot specify PMA on the same command line with an IPL systemname.

See the DEFINE and LINK commands for the preferred machine assist virtual device related restrictions.

Note: You may only issue the IPL command with the PMA operand to load MVS/SP Release 1 with Release 1 Enhancements, or later enhancements. The results are unpredictable if you issue this command to load any other system. A CP system abend may result.

PMAV

enables the control switch assist that provides CP Diagnose and IUCV (inter-user communication vehicle) capabilities in addition to the preferred machine assist described above for the PMA parameter. To use control switch assist, you must meet the requirements listed under the description of the PMA parameter above.

When using the PMAV parameter, do not use the PMA, STOP, or ATTN parameters.

systemname

simulates the IPL function when loading a named system that was previously saved via the SAVESYS command. It is loaded into virtual storage and given control. For information about saved systems, see the *VM/SP HPO System Programmer's Guide*.

Note: You cannot load a shared system or one that uses discontinuous saved segments into a virtual machine running in the V=R area. An attempt to do so results in an error message.

You can load a system into the V=R area that was stored by the VMSAVE function. When loading a VMSAVE system, the date and time that the system was stored is displayed. In addition, the page ranges that were saved are displayed. The page-range display can be suppressed by using the SET IMSG OFF command.

PARM p1 [p2...]

passes up to 64 bytes of data (excluding all leading blank characters after the keyword, PARM, but including all other imbedded and trailing blanks) to your virtual machine's general registers (4 bytes per register), starting with the high order byte of general register 0.

Note: The registers are not cleared before use.

Whenever you specify PARM, the remaining characters on the command line are treated as parameters to be passed to your virtual machine; PARM must be the last operand on the command line. For restrictions and more detailed information on parameters, refer to reference material specific to the system the parameters are being passed to.

Usage Notes

1. You cannot use this command with the 3880 Storage Subsystem because subsystem devices are not supported as system residence devices.
2. IPL simulates the LOAD button and the device address switches on the real computer console. The specified virtual address is accessed and the required input/output operations are performed to retrieve the IPL data.
3. Optionally, the IPL procedure can be stopped just before loading the virtual PSW except when loading a named system. Also, parameters can be passed to the virtual machine's general registers. When the simulated load function is complete, CP initiates execution of the virtual machine by loading the IPL PSW which was stored during the simulation process.
4. Care must be used when passing parameters to a named system (systemname). Named systems expect certain registers to be initialized when they are given control. Indiscriminate use of the PARM operand could overlay a previously initialized register causing unpredictable results.
5. The IPL simulator program occupies one page (4096 bytes) of storage in your virtual machine. Therefore, if the system being loaded utilizes data remaining in storage by the previous system, care should be taken as to where that data is located. The starting address of the overload virtual storage is:

Virtual Storage Size	Location of IPL Simulator Page
256K and above	= 20,000
Less than 256K	= 1/2 the storage size

For example:

Storage size = 200K then the IPL Simulator is loaded at 100K.

6. To load a VMSAVE system, a number of conditions must be met:
 - The user must be specified in module DMKSNT as either the owner (USERID =) or the receiver (RCVRID =).
 - The area must contain a valid saved system.
 - The system must have been saved under the same system name expressed in the IPL command.
 - The area cannot be the target area for other users currently enabled for VMSAVE.

When these conditions are not met, an appropriate message is issued, and the IPL is terminated.

7. Any logical lines following the IPL command are ignored since the execution of the IPL command transfers control from the CP environment to the virtual machine environment.
8. Some CMS routines and programs are stored on disks and loaded into storage as needed. Since this storage area is not contiguous with your virtual storage, the saved segments that are loaded in this area are called discontinuous segments (DCSS). However, the only discontinuous saved segment a CMS user can replace is the CMSSEG segment. The user can indicate an alternate segment to be loaded by issuing an IPL command with this format:

```
IPL {cuu          } PARM[SEG=segmentname]
    {systemname } [SEG=NULL
```

where:

SEG = segmentname

indicates the name of the saved segment to be loaded whenever the CMS Editor, System Product Editor, EXEC or EXEC2 interpreters, or OS simulation routines are needed. The segmentname can be a maximum of eight characters long. If you specify a segmentname of less than eight characters, you must specify a blank character after the last character of the segmentname to act as a delimiter. You must do this even if the segmentname is the last operand on the command line.

SEG = NULL

indicates that a CMSSEG segment name is not to be loaded on the IPL command. If you issue SEG = NULL, the IPL command will not load the CMSSEG segment.

The CMS batch facility loads whatever segment is specified on the first IPL command issued for the batch virtual machine. Thus, if the first IPL command for a CMS batch facility machine is:

```
IPL CMS PARM SEG=CMSSEG02
```

all subsequent IPL commands issued by the CMS batch facility specify the same segment name (CMSSEG02).

9. The BATCH option in the PARM operand indicates that the CMS initial program load is being performed in a batch instead of an interactive virtual machine. This option causes the CMS nucleus to issue the CMSBATCH command to load the batch processor. Specifying the BATCH option eliminates the need for operator intervention between jobs. Consult *VM/SP CMS Command and Macro Reference* for information on the CMSBATCH facility.

10. CMS users can use the PARM operand to pass the AUTOOCR (automatic carriage return) option to the CMS initialization routine. AUTOOCR simulates the pressing of the ENTER key as input to the virtual machine. This results in automatic execution of the user's PROFILE EXEC, if it exists on virtual minidisk 191.

Note: For a disconnected CMS service virtual machine, this mechanism can prove to be very useful. The name of a CMS restart routine may be specified in the CMS nucleus. In the event of an ABEND, the specified restart routine receives control instead of the service virtual machine entering a disabled wait state. If this routine issues an IPL CMS PARM AUTOOCR command, the PROFILE EXEC executes initial setup procedures that allow the service virtual machine to reestablish itself.

11. An AUTOLOG user should not use AUTOOCR if SET AUTO READ OFF is in his profile.
12. If the user issues an 'IPL vaddr ATTN' command to a virtual machine other than OS/VS1 with the FASTNIP feature, results are unpredictable. The results depend on how the object operating system handles ATTN interrupts during IPL.
13. The preferred machine assist feature is reset (turned off) if:
 - You issue an IPL systemname on the IPL command line
 - You issue an IPL vaddr cylno/nnnnnn command without specifying the PMA operand
 - You issue one of the following commands:
 - SYSTEM RESET
 - SYSTEM CLEAR
 - SET ECMODE
 - DEFINE STORAGE or DEFINE CHANNELS
14. When you IPLed CMS by name and the virtual storage size is too small to contain all of the saved CMS system pages, CP treats the CMS saved system as a discontinuous saved segment. If there are shared segments in CMS beyond the VMSIZE, any subsequent IPL causes the saved CMS system to be purged in its entirety. This means that if data is stored into CMS saved pages in low storage, the pages will be cleared, even if the 'NOCLEAR' option is specified for the IPL. To have the data available for the IPL, either define storage large enough to accommodate all of CMS or use the STOP option on the second IPL, store the data required for the IPL, and continue with the BEGIN command.
15. When IPLing the loader from your virtual machine in order to create a CP nucleus, set CONMODE to 3215 mode. Otherwise, console messages generated by the loading process will not be displayed at the terminal.

Responses

After a successful IPL, any responses you receive are those from the operating system that was loaded and initialized.

LINK

Privilege Class: G

Use the LINK command to make a device that is associated with another virtual machine available to your virtual machine configuration, based upon information in that user's directory entry.

The format of the LINK command is:

```
LINK [To]userid vaddr1 [As] vaddr2 [mode] [[PASS=] password1]
```

¹If your installation uses the password suppression facility, the password for a DASD device cannot be entered on the LINK command line. See the full description of the password operand for the steps needed to link to DASD with the password suppression facility in effect.

where:

TO userid

is the name of the user whose directory is to be searched for device *vaddr1*. An asterisk (*) is used to specify that the device is in your own directory. If the keyword TO is omitted, the *userid* may not be "TO" or "T".

vaddr1

is the virtual device address (*cuu*) in the directory entry for that *userid*.

AS vaddr2

is the virtual address (*cuu*) that is to be assigned to the device for your virtual machine. If the keyword AS is omitted, *vaddr* may not be "A". If your virtual machine has the ECMODE option, any address up to X'FFF' is valid; otherwise, any address up to X'5FF' is valid.

mode

is the access mode which consists of up to two letters. The first letter specifies the primary access mode (read-only, write or multiple). The optional second letter indicates the alternate access mode (read-only or write access) desired if the primary access is not available. Valid modes are:

Mode Meaning

- R** Primary read-only (R/O) access. The read-only link is established as long as no other user has the disk in write status. If there is an existing write link to the disk, no link is given. R is the default mode if the link is to another userid.
- RR** Primary read-only (R/O) access or alternate read-only access. The read-only link is established even if another user has the disk in write status. The alternate access of R assures the user will get the read link no matter what links currently exist to the disk.

- W** Primary write access. The write link is established only if there are no other current links to the disk. If another user has the disk in read or write status, no link is given.
- WR** Primary write access or alternate read-only access. If write access is available, then the write link is established. Otherwise, the alternate access of read-only is given.
- M** Primary multiple access. A write link is established unless another user has write access to the disk, in which case no link is given.
- MR** Primary multiple access or alternate read access. A write link is established unless another user has write access to the disk, in which case, a read link is given.

Note: Unpredictable results can occur when one user has a R/O (R or RR) link to a device that is being updated by a user who has the device in write/read status.

- MW** Primary multiple access or alternate write access. A write link is given in all cases.

Warning: CMS does not protect a user from loss of data on a disk when multiple users have write access to the same disk. More than one user writing to the same virtual device can result in a permanent loss of data. Users should not be linking with MW mode to obtain the M or MR function. (The M or MR access modes will allow only one write link to a disk.)

If the mode is omitted, the default is R if the userid is another user. If you are linking to one of your own disks, the default is the "user access mode" as specified in the directory entry for your disks (on the MDISK or LINK control statement).

PASS = password

is a one-to-eight character string that must match the access mode password for device *vaddr1* in the directory for the user (*userid*) specified. The password should be specified only when the LINK is executed by a virtual machine (for example, from CMS), since printing of the password is not suppressed when included with the LINK command. The access mode password cannot be the same as any of the access modes (R, RR, W, WR, M, MR, or MW) if the default mode is to be used.

Notes:

1. *The access mode password should not be confused with a user password.*
2. *If your installation is using the password suppression facility, an INVALID FORMAT message is issued when a user attempts to enter the password for a DASD device on the LINK command line. The user must wait for the ENTER PASSWORD response before typing in the password. This facility improves system security because the password is automatically masked.*
3. *PASS= must be followed by a blank.*

Usage Notes

1. You may redefine the machine's device configuration during a terminal session to increase the number of virtual devices. The maximum number of devices can be as large as 410. If the MAXDEV xxxx option is part of your directory's OPTION control statement, you may redefine the machine's device configuration during a terminal session to increase the number of virtual devices up to the option-defined limit that is greater than 410 (maximum being the xxxx of the control statement). You can increase the number of virtual devices only when there is enough contiguous space available in main storage.

The 410 limit should not be exceeded unless necessary since it requires additional storage.

This limit may be changed as a result of modifications made by the installation.

For a description of the OPTION control statement see the *VM/SP HPO Planning Guide and Reference*. The *VM/SP HPO System Programmer's Guide* discusses the details of this option.

2. If you link to one of your own disks, no password is required. If another user has write access to one of your disks when you log on, your access will be forced read-only (R/O). You can obtain write access by issuing a LINK command with an access mode of MW. Now two write links exist to the disk and the integrity of the data cannot be guaranteed by CP or CMS.
3. If the link is to a device whose password is ALL, (meaning that any user can access the device), the password is not required. However, if the link is to a device whose password is not ALL, a user must provide password. If a LINK command is issued from a CMS batch job, a password must be provided even if the access password is ALL.
4. When linking to a 2305 device, you gain access only to the virtual address specified and not to all eight addresses associated with the device.

LINK

5. The access allowed by the LINK command to the *vaddr1* device belonging to *userid* is summarized below. You read the columns down to determine the type of link that results. The first row indicates the primary (and, optionally, the alternate) access mode requested. The second row indicates whether read, write, or multiple passwords exist in the directory for the disk being linked. The third row indicates whether the disk is already being used, and if so, the mode of its access. The last row indicates the type of link established. For example, the third column is interpreted as follows: if you request a read access link (R) to a disk that has a read password defined and that already is accessed in read mode, you can establish a read link.

Primary access requested:	R	R	R	R	R	W	W	W	W	W	W	M	M	M	M	M	M			
Alternate access (if any):				R					R		R					R	W			
Read password accepted: (If in directory)	N	Y	Y	Y	Y															
Write password accepted: (If in directory)						N	Y	Y	Y	Y	Y									
Mult. password accepted: (If in directory)												N	Y	Y	Y	Y	Y			
Any existing links:		N	R	W	W		N	R	R	W	W		N	R	W	W	W			
Access established:		N	R	R	N	R		N	W	N	R	N	R		N	W	W	N	R	W
<i>where:</i> N=no or none; R=read; W=write; M= multiple; Y=yes																				

Figure 9. Password Requirements on the LINK Command

6. Virtual reserve/release processing cannot be requested by appending a V to the mode operand of the LINK command line. Virtual reserve/release processing can only be requested by means of the MDISK statement of the directory program.
7. The LINK command, when issued with the preferred machine assist must meet the preferred machine assist virtual device restrictions as follows:

If you are the V=R user working with the preferred machine assist feature enabled, you must:

- LINK only to a permanent full extent virtual disk (not a T-disk)
- Use devices or full extent virtual disks for which:
 - a. The virtual and real addresses are equal or there is no real device control block (RDEVBLK) specified for the virtual address; and
 - b. The virtual channel address is a CP-owned channel.

Notes:

- a. *These restrictions hold whether or not preferred machine assists control switch assist is enabled.*
 - b. *When following these restrictions for linking to a full extent virtual disk, the real address is the address of the pack's physical location—not the disk owner's current virtual address.*
8. You cannot link to a minidisk located on a 3880-11 or 3880-21 Storage Subsystem device. You can link to a minidisk located on a 3880-13 or 3880-23.
 9. When issuing the LINK command, be aware that VM/SP HPO does not support a mixture of SHARED and NONSHARED device types on the same virtual control unit. If not specified correctly, *vaddr2* will cause a conflict in the user's virtual machine. If this happens, the command is not executed and an error message is sent. Refer to Appendix B in the *VM/SP HPO Planning Guide and Reference* for a listing of the Virtual Device Characteristics (shared and nonshared) associated with each device. The table indicates the proper device specification, the command or statement, the internal device type and class, and the protocol.

Responses**ENTER READ, WRITE, OR MULT PASSWORD:**

■ ■ ■ ■ ■ ■ ■ ■

Type the read, write, or mult password over the mask to obtain read access to the desired disk.

ENTER WRITE OR MULT PASSWORD:

■ ■ ■ ■ ■ ■ ■ ■

Type the write or mult password over the mask to obtain write access to the desired disk.

ENTER MULT PASSWORD:

■ ■ ■ ■ ■ ■ ■ ■

Type the multiple password over the mask to obtain write access to a disk for which other users may already have access.

Note: If the LOGON/AUTOLOG/LINK journaling facility is activated, unsuccessful link attempts, due to invalid passwords, are recorded. When a predefined system threshold value for attempts is exceeded, an installation userid receives a message. There is another system threshold value that disables the LINK command when a user continues to enter trial passwords to obtain a link to a minidisk. The default is ten incorrect passwords before the command is disallowed for the remainder of the terminal session.

DASD vaddr2 LINKED R/O

This response indicates that a read-only link to the given disk is established for a LINK request with a mode of R or RR, and that no other users are linked to the same disk in read/write mode.

DASD vaddr2 LINKED R/W

This response indicates that a read/write link to the given disk is established for a LINK request with a mode of W, WR, M, MR, or MW, and that no other users are linked to the same disk.

DASD vaddr2 LINKED R/O; R/W BY {nnn USERS} [; R/O BY {nnn USERS}] userid

This response indicates that a read-only link to the given disk is established for a LINK request with a mode of RR, but warns that the disk is in read/write use by some users and possibly in read use by some users. If only one user has access, the number of users (*nnn USERS*) is replaced by *userid*.

DASD vaddr2 LINKED R/W; R/O BY {nnn USERS} userid

This response indicates that a read/write link to the given disk is established for a LINK request with a mode of M, MR, or MW, and informs you that the disk is also in read-only use by *userid* or by *nnn* users. (No other users have a read/write link to the disk.)

DASD vaddr2 LINKED R/W; R/W BY {nnn USERS} [; R/O BY {nnn USERS}] userid

This response indicates that a read/write link to the given disk is established for a LINK request with a mode of MW, but warns you that the disk is also in read/write use by some users and possibly in read use by some users. If only one user has access, the number of users (*nnn USERS*) is replaced by *userid*.

Note: In the above responses, a link is indicated for those users whose virtual devices have the same starting cylinder or block on the same real device.

LOADVFCB

Privilege Class: G

Use the LOADVFCB command to specify the forms control buffer image for a virtual spooled 3203, 3211, 3262, 3289E, 4245, or 4248 printer.

The format of the LOADVFCB command is:

```
LOADVFCB      vaddr      Fcb  name  [Index [nn]]
```

where:

vaddr

is the virtual device address (*cuu*) of the virtual spooled 3203, 3211, 3262, 3289E, 4245, or 4248 printer.

FCB

is a required reserved keyword meaning Forms Control Buffer.

name

is a system-defined name for the 3203, 3211, 3262, 3289E, 4245, or 4248 FCB image which is to be the controlling virtual FCB image.

INDEX nn

for the 3211 only, is the number of the print position that is the first print position. The value, *nn*, must be a number from 1 through 31; a leading zero need not be specified. If the keyword INDEX is specified without a value, the index defaults to the value specified in the FCB macro. See the *VM/SP HPO System Programmer's Guide* for a discussion of the FCB macro and forms control images.

The 3203, 3262, 3289E, 4245, and 4248 do not support the index feature. CP rejects any attempt to issue the INDEX operand.

Usage Notes

1. The LOADVFCB command can be used with installations that do not have a 3203, 3211, 3262, 3289E, 4245 or 4248 printer. The virtual machine's directory entry must indicate a 3203, 3211, 3262, 3289E, 4245 or 4248 even though the program and operating system have a 1403 printer defined. Then the LOADVFCB command can be used to specify a virtual FCB image for 1403 printers so that programs that use printer overflow sensing may be spooled to disk.

LOADVFCB

2. VM/SP HPO provides two FCB image (FCB1 and FCB8) with the following format:

Line spacing = 6 lines/inch												
Length of page = 66 lines												
Page line	1	3	5	7	9	11	13	15	19	21	23	64
Channel skip	1	2	3	4	5	6	7	8	10	11	12	9

Line spacing = 8 lines/inch FCB8												
Length of page = 68 lines												
Page line	1	4	8	12	16	20	24	28	32	36	63	66
Channel skip	1	2	3	4	5	6	7	8	10	11	12	9

Your system programmer may provide you with additional FCB images.

3. The virtual machine operating system may issue a Load FCB CCW to the spooled printer. This loads a virtual FCB into the spooled printer replacing the previous virtual FCB, which may have been loaded with the LOADVFCB command. Note that CMS does not use the Load FCB CCW in normal printer operations, thus the LOADVFCB command may be used with CMS to control the printer.
4. If no LOADVFCB command or Load FCB CCW is issued to a spooled printer, the virtual printer uses a default FCB image for its processing but does not load this FCB into the spool file. This default image specifies 6 lines per inch, 66 lines per page, channel 1 in line 1, channel 2 in line 2, channel 3 in line 3, channel 4 in line 4, channel 5 in line 5, channel 6 in line 6, channel 7 in line 7, channel 8 in line 8, channel 9 in line 9, channel 10 in line 10, channel 11 in line 11, and channel 12 in line 60. The real printer uses the FCB loaded into it by the system.
5. If the default FCB image has been modified, the virtual FCB image from the spooled printer is stored in any spool file created on the spooled printer. When the spool file is printed on a real printer, the FCB is also sent to the real printer. The spool file will then print properly on the real printer.

Note: This does not apply if either the virtual or real printer is a 1403.
6. If an attempt is made to send an FCB that is too long, an error message is generated.
7. The LOADVFCB command simulates sending a LOAD FCB CCW to a 3211 type printer. The LOAD FCB CCW when sent to the real hardware resets the FCB line position to 1. Therefore, it is necessary for the application to ensure that the paper is at the top of the page when an FCB is loaded.

8. The LOADVFCB command allows specification of the extended FCB image for a virtual 4248 Model 1 printer. If an attempt is made to send the extended FCB image to a printer that does not support the extended FCB format, an error message is generated.

Responses

None.

LOGOFF

Privilege Class: Any

Use the LOGOFF command to terminate a virtual machine session and disconnect your virtual machine from the system.

The format of the LOGOFF command is:

$\left. \begin{array}{l} \text{LOGoff} \\ \text{LOGout} \end{array} \right\}$	[HOLD]
---	--------

where:

HOLD

retains the connection for a switched communication line to enable you to logon without redialing the system. This operand is invalid if you are on a terminal running under control of the GCS or VCNA interface.

Usage Notes

1. This command causes all active spool files to be closed, temporary disks to be relinquished, dedicated devices to be detached, and an accounting record to be created for the user.
2. You should always logoff of your terminal in addition to turning power off on the terminal. Terminal power off is not synonymous with logoff.

If you turn power off at the terminal instead of logging off, logoff occurs by one of the following methods:

- *Typewriter Terminal and Remote 3270 Display Terminal*—Logoff takes place after a 15-minute interval has elapsed. This occurs if no attempt is made to turn terminal power on and re-establish communications with the still logged-on virtual machine during this 15-minute period.
 - *Local 3270 Display Terminal*—Logoff only takes place 15 minutes after VM/SP HPO discovers that the terminal has been turned off (that is, VM/SP HPO attempts to send a message to the terminal, but gets back an error code indicating that the terminal is turned off).
3. If an I/O device (such as a disk or tape drive) drops ready while it is processing virtual I/O activity, any virtual machine users performing I/O on that device are unable to continue processing or to log off. The LOGOFF command is not effective in this case because it does not complete until all waiting I/O is finished. The system operator should determine which I/O device is involved and make the device ready once more to allow the user to log off.

Responses

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

where:

CONNECT = hh:mm:ss

is the actual clock time spent in the current terminal session in hours:minutes:seconds.

VIRTCPU = mmm:ss.hs

the virtual processor time used in the current terminal session in minutes:seconds.hundredths of seconds.

TOTCPU = mmm:ss.hs

the total processor time (including virtual and overhead) used in the current terminal session in minutes:seconds.hundredths of seconds.

These times are either the elapsed time for the entire terminal session or the elapsed time since the ACNT command was entered for this user.

LOGOFF AT hh:mm:ss zone weekday mm/dd/yy

is the response for a logoff.

This system then responds with one of the following, depending on the type of terminal you are using:

Type of Terminal	Message
327x, 3290	PRESS ENTER OR CLEAR KEY TO CONTINUE
SNA	PRESS ENTER KEY TO CONTINUE

Pressing the appropriate key on your terminal causes the logo to appear on the terminal which readies the terminal for the next session. For terminals not listed above, control automatically returns to CP and the "vm/370 online" message printed.

$$\left. \begin{array}{l} \text{(GRAF raddr)} \\ \text{(LINE raddr)} \\ \text{(DEV resid)} \end{array} \right\} \text{LOGOFF AS userid USERS = nnn}$$

is the normal response to the primary system operator. The term *raddr* is a three-digit real device address in the form *cuu*. *DEV resid* specifies the resource identification of a 3704/3705/3725 line.

LOGOFF

**{ GRAF raddr }
LOGOFF AS userid USERS = nnn FORCED
{ LINE raddr }
{ DEV resid }**

is the response to the primary system operator if the logoff is forced by a line timeout or a terminal power-off. The term *raddr* is a three-digit real device address in the form *cuu*. *DEV resid* specifies the resource identification of a 3704/3705/3725 line.

USER DSC LOGOFF AS userid USERS = nnn

is the response to the primary system operator when logoff occurs for a user who had previously disconnected using the DISCONN command.

LOGON

Privilege Class: Any

Use the LOGON command to identify yourself to the system and to access that system.

The format of the LOGON command is:

$\left. \begin{array}{l} \text{Logon} \\ \text{Login} \end{array} \right\}$	userid [password] [Noipl]
---	---------------------------

where:

userid

is the identifier assigned to you in the system.

password

is your password. Specify this field if no protection (that is, masking characters) is desired.

Note: If your installation is using the password suppression facility an INVALID FORMAT message is issued when a user attempts to enter the system LOGON password on the command line. The user must wait for the ENTER PASSWORD response before typing in the password. This facility improves system security because the password is automatically masked.

NOIPL

specifies that the IPL device or name in the directory should not be used for an automatic IPL.

Usage Notes

1. If your terminal is already powered on when you want to use it, you can enhance security by turning power off and then on again, before using the LOGON command. This ensures that the terminal is logically connected to your first level VM/SP HPO system and not to some other user's virtual machine.

If your terminal is a SNA terminal supported through either GCS/VTAM/VSCS or VS1/VTAM/VCNA, you can enhance security by pressing the system request key on the SNA terminal. This switches you to VTAM Subsystem Services Control Point (SSCP). After you have logged on to the VTAM application, you can then log on to VM and be assured that you are connected to your first level VM/SP HPO system with a new session.

2. Upon successful logon, VM/SP HPO creates a virtual machine configuration from information in that user's directory entry. The LOGON command name may not be entered using any line-editing symbols, but the operands may use these symbols. See the *VM/SP Terminal Reference* for a detailed description of logon procedures.

3. If you use LOGON because a teleprocessing line or terminal error disconnected you from your virtual machine, you have 15 minutes to logon again. If you do not logon within 15 minutes, your virtual machine automatically logs off. In this case, you may have to reconstruct files and restart jobs interrupted by the teleprocessing line or terminal error.
4. If you previously issued the DISCONN command to run your virtual machine without a console and then issue the LOGON command to reconnect your console, certain SET and TERMINAL command operands are forced off even if they were on prior to issuing the DISCONN command. For example, after a reconnect, CP resets the line delete character specified in the directory to the system default value. Issue the QUERY SET and QUERY TERMINAL command lines to check which functions are currently active.
5. During the virtual machine logon process on a system generated for attached processing operation but operating in uniprocessor mode, a message is sent to the user with attached processor AFFINITY set (using the user directory's OPTION statement) that the AFFINITY setting of his virtual machine is nullified by the system's uniprocessor mode of operation.
6. During the virtual machine logon process, you cannot link to a minidisk located on a 3880 Storage Subsystem device.
7. If the LOGON/AUTOLOG/LINK journaling facility is activated, accounting records are produced for unsuccessful logon attempts with incorrect passwords. If system defined values are exceeded attempting to logon with trial passwords, a message is issued to an installation *userid* and any further LOGON attempts by the user are rejected.
8. CP automatically masks your password on typewriter terminals to insure installation security. For installations migrating from a VM/370 operating system to VM/SP HPO, and for compatibility purposes, CP still accepts the MASK operand on the LOGON command as valid. However, CP automatically masks out your password should you forget to issue a LOGON command without the MASK operand.
9. When a terminal user is logging on but has not yet been validated, he has a temporary identification. This temporary identification takes one of the following forms:
 - If the user is at a local terminal, this is LOGONnnn in which nnn is a local address
 - If the user is at a logical terminal, this is LOGOLnnn in which nnn is a logical address
 - If the user is at a remote terminal, this is LOGNnnnn in which nnnn is a remote resource identification (resid).

Responses**ENTER PASSWORD:**

indicates that the *userid* has been accepted. You should now type in the password.

LOGMSG- hh:mm:ss mm/dd/yy

indicates the time and date at which the system log message was generated or most recently revised. All lines of the log message for which the first character is an asterisk are displayed at this point. If you wish to see all of the system log messages, you must issue the CP command QUERY LOGMSG.

**FILES: {nnn}RDR, {nnn} PRT, {nnn} PUN
{NO} {NO} {NO}**

This message is omitted if all counts are zero; otherwise, it indicates the number of spool files that exist for you at logon time.

LOGON AT hh:mm:ss zone weekday mm/dd/yy

or

RECONNECTED AT hh:mm:ss zone weekday mm/dd/yy

indicates the time, day of the week, and date at which the LOGON or RECONNECT is complete.

$\left. \begin{array}{l} \text{luname} \\ \text{GRAF} \quad \text{raddr} \end{array} \right\}$	$\left. \begin{array}{l} \text{LINE} \quad \text{raddr} \\ \text{DEV} \quad \text{resid} \end{array} \right\}$	LOGON AS <i>userid</i>	USERS = <i>nnn</i>
or			
$\left. \begin{array}{l} \text{luname} \\ \text{GRAF} \quad \text{raddr} \end{array} \right\}$	$\left. \begin{array}{l} \text{LINE} \quad \text{raddr} \\ \text{DEV} \quad \text{resid} \end{array} \right\}$	RECONNECT <i>userid</i>	USERS = <i>nnn</i>

is the response to the primary system operator. DEV *resid* specifies the resource identification of a 3704/3705/3725 line. A *luname* response is a one-to-eight character name given for a user who logs on to VM/SP HPO through a SNA network. The form *raddr* in this response is a three-digit real device address in the form *cuu*.

MESSAGE

MESSAGE

Privilege Class: Any

Use the MESSAGE command to transmit message text to a specified *userid* or to the primary system operator.

The format of the MESSAGE command is:

$\left. \begin{array}{l} \text{Message} \\ \text{Msg} \end{array} \right\}$	$\left. \begin{array}{l} \text{userid} \\ * \\ \text{OPERator} \end{array} \right\}$	msgtext
---	--	---------

where:

userid

is the identification of the user who is to receive the message.

*

specifies that you are sending a message to yourself.

OPERATOR

sends the message to the primary system operator regardless of his *userid*.

msgtext

is the text of the message that is to be transmitted. The length of the message is limited to the number of characters remaining on the input line after entering the command and the appropriate operand in most cases. On a 3178, however, though the input line is 139 characters, but CP will process only the first 130 characters, omitting the last nine.

Usage Notes

1. If the user designated to receive the message is not logged on or has suppressed the receiving of messages (SET MSG OFF), the message is not transmitted and the sender receives a diagnostic message to this effect. A message which is not received by a user is not saved and must be sent at a later time when the user is receiving messages.
2. The message is displayed at the terminal when the terminal is ready to receive output. If a typewriter terminal (or a display terminal having AUTOREAD set ON) is entering data, the message is not displayed until an end-of-line (carriage return or ENTER) signal is received.
3. If the message is received by the primary system operator console and SET MSG ON is in effect, the alarm at the central computer console sounds.
4. If the user receiving the message is logged on with a 3277 virtual console that has the alarm feature, the alarm sounds.

5. MSG * is useful in identifying the current *userid* of an abandoned terminal, or it can be used in a program intended to be executed under several *userids*.
6. To be established as the system operator, the operator must be the only class A user logged on. If the system operator logs off and another class A user logs on, SYSOP in DMKSYS will contain that class A user. Messages to the operator will go to this user - even if the operator logs back on. If the operator does log back on and the class A user logs off, DMKSGO45E 'OPERATOR NOT LOGGED ON' will be received for messages to the operator. In order to be recognized as system operator by the system, the system operator must log off and then log back on again.

Responses

hh:mm:ss

MSG FROM OPERATOR: msgtext

is the message received by the user from the system operator.

hh:mm:ss

MSG FROM $\left. \begin{array}{l} \text{LOGONnnn} \\ \text{LOGNnnnn} \\ \text{LOGOLnnn} \\ \text{userid} \end{array} \right\}$ **: msgtext**

is the format of the message sent to another user or to the system operator, where *userid* is that of the sender. If the user sending the message is not logged on to VM/SP HPO, LOGONnnn, LOGNnnnn, or LOGOLnnn and the line number are displayed instead of *userid*.

hh:mm:ss

is the time in hours:minutes:seconds when the message was sent to the user.

NOTREADY

NOTREADY

Privilege Class: G

Use the NOTREADY command to cause a virtual device to appear as if it had changed from ready to not ready status.

The format of the NOTREADY command is:

NOTready	vaddr
----------	-------

where:

vaddr

is the virtual device address (*cuu*) of the unit to be removed from ready status.

Usage Notes

1. This command is for use with spooled unit record devices and virtual consoles only.
2. Any I/O operation to the specified device, in progress at the time the command is issued, is completed. On the next Start I/O (SIO) instruction, the not ready condition is in effect.

Response

None.

ORDER

Privilege Class: G

Use the ORDER command to place your closed spool files in a specific order by device type.

The format of the ORDER command is:

ORDER	$\left\{ \begin{array}{l} \text{Reader RDR} \\ \text{Printer PRT} \\ \text{Punch PCH} \end{array} \right\}$	$\left\{ \begin{array}{l} \text{Class c1 Class c2...} \\ \text{FORM form1 FORM form2...} \\ \text{spoolid1 spoolid2...} \end{array} \right\}^1$
<small>¹ A combination of CLASS, FORM, and spoolid specifications may be entered.</small>		

where:

[READER]
RDR

specifies that the reader spool files are to be placed in order.

[PRINTER]
PRT

specifies that the printer spool files are to be placed in order.

[PUNCH]
PCH

specifies that the punch spool files are to be placed in order.

CLASS c1 CLASS c2...

are the class(es) to be rearranged and the sequence in which they are to be processed. CLASS is a required reserved word and c1, c2,... are one-character alphameric fields (with values from A to Z and from 0 to 9) that represent spooling classes.

FORM form1 FORM form2...

are the forms you want rearranged and the sequence in which you want CP to process them. The form1 and form2 specifications are one-to-eight character form names defined during system generation.

spoolid1 spoolid2...

are the *spoolids* of the files to be rearranged and the sequence in which they are to be processed.

Usage Notes

1. You may order only your own files.
2. Use the QUERY command to determine spool file attributes such as form, class, and *spoolid* for files you want to rearrange.

ORDER

3. You can use a combination of class, form, and *spoolid* specifications to rearrange files. For example:

```
Order PRinter CLass A 1963 CLass C FORM CATALOG4
```

specifies that CP queues printer files in the following order: all class A files, the file with spoolid 1963, all class C files, all files with form name CATALOG4, followed by any other printer spool files not specified in the ORDER command line.

4. When duplicate *spoolids* are present, the ORDER command resequences only the first duplicate spool file found.
5. The order of your spool files may change without warning between terminal sessions if the system was restarted with a checkpoint restart.

Response

```
{nnnn} FILES ORDERED  
{NO }
```

indicates the number of files that were placed in order. It is not displayed if you issued the CP SET IMSG OFF command.

PER

Privilege Class: G and all classes except class ANY.

Use the PER command to monitor certain events as they occur during program execution in your virtual machine. The events that can be monitored by the PER command are:

1. The fetching and execution of an instruction.
2. The execution of a successful branch instruction.
3. The execution of an instruction that alters a specific general purpose register.
4. The execution of an instruction in the virtual machine that alters storage.

For examples on how to use the PER command for debugging, refer to “Testing and Debugging of Programs” on page 29.

PER

The format of the PER command is:

PER	<p><u>EVENT TYPES:</u></p> <pre> Instruct [DATA] hex-data Branch [INTO] into-addr-range Store [[INTO] storage-addr-range [INTO] addr [DATA] hex-data] Mask [INTO] addr [DATA] mask-field G[reg1] [{ - } [reg2]] [[DATA] hexword] [{ : } [regcount]] [{ . }] </pre> <p style="text-align: right;">[options]</p> <p><u>OPTIONS:</u></p> <pre> Range instruction-addr-range FRom instruction-addr-range PAss [0 n] CMd {text} [Printer RUN Terminal [NORun Both [RUN [STEP [1 [n]]]]] GUESTR GUESTV DATOFF DAT <u>SUB-COMMANDS:</u> Count Table SAve traceset-name [APPend] GET traceset-name [APPend] ENd { ALL { Count { Current { element-number { event-type { traceset name </pre>
-----	---

where:

Event Types

The PER command supports 5 different EVENT keywords. You can specify more than one EVENT keyword on a single line, and you can specify them in any order. If an into-range or data is specified following an event, then these parameters apply only to that event. They do not apply to other events on the same command line.

Instruct [[DATA] hex-data

specifies the tracing of the execution of instructions within the virtual machine. The hex-data is optional. The DATA keyword is optional. If you specify the DATA keyword, then you must also specify hex-data. Hex-data must be expressed in complete bytes. If you do not specify hex-data, then all instructions are traced. If hex-data is provided, then only instructions that match the hex-data specified are displayed.

Examples:

PER Instruct	traces all instructions
PER I DATA 18	traces LR instructions (op code X'18')
PER I DATA 1845	traces all LR 4,5 instructions
PER I DATA 1967 I DATA 078F	traces both CR 6,7 and BZR 15 instructions separately
PER I DATA 1967078F	traces CR 6,7 instructions only if they are followed by a BZR 15 instruction

BRanch [[INTO] into-addr-range]

specifies the tracing of successful branch instructions. A successful branch occurs whenever one of the following 10 instructions causes control to be passed to the instruction designated as the branch address:

BAL	BAS	BC	BCT	BXH
BALR	BASR	BCR	BCTR	BXLE

Note: LPSW and SVC are not detected by the PER hardware as a successful branch.

The INTO keyword is optional. If you specify the INTO keyword, then you must specify an into-addr- range.

The into-addr-range is optional. See usage note 1 on page 147 for the various ways that you can express this address range. If you do not specify an into-addr-range, then all successful branch instructions are displayed. If you specify an into-addr-range (other than the default range 0-FFFFFF), then a traceback table of the five previous successful branch instructions is displayed when a branch occurs into the address range specified.

Examples:

PER BRanch	traces all successful branches
PER BR INTO 0	traces successful branches to location 0
PER BR INTO 20000.100	traces successful branches to any instruction whose first byte is between location 20000 and 200FF inclusive

STore **[**[INTO] storage-addr-range **]**
[[INTO] addr [DATA] hex-data **]**

specifies the tracing of the execution of instructions within a virtual machine that alter storage. The two positional operands are optional.

The INTO and DATA keywords are optional. If you specify the INTO keyword, then an address range must be specified. If you specify the DATA keyword, then you must also specify hex-data.

If you don't specify any operands, then all instructions causing alterations to storage are traced.

If you specify the storage-addr-range, trace output will be produced only when an instruction alters storage in the indicated range.

If you specify addr hex-data, then you must specify addr as a single address. It indicates the address where the data comparison will begin. The hex-data that follows must consist of complete bytes. The length of the hex-data will determine the number of bytes to be tested. In this case, trace output will be produced whenever the storage starting at the given address matches the data.

Note: Whenever an instruction places a value into storage, then that instruction has altered storage. It is not necessary that this value be different from the previous value.

Examples:

PER STore INTO 200-210	traces any instruction that changes storage anywhere in the storage range between 200 and 210 (inclusive)
PER ST INTO 200 DATA c1c2c3	traces instructions that alter storage in the range 200 - 202 and result in these storage locations containing the character string ABC

Mask [INTO] addr [DATA] mask-field

specifies the tracing of the execution of instructions that cause an alteration to specific bits at a specific storage location. The operands are positional and required. The INTO and DATA keywords are optional. The first operand (addr) is the address of the first byte in storage that will be tested. addr must be a single address.

The mask-field consists of complete bytes. The length of the mask-field determines the number of storage bytes involved in the test. The binary ones in the mask-field indicate the specific bits in storage that are to be checked. If any alteration of these bits in storage occurs, the instruction that caused the change will be displayed as well as the new contents of the storage location.

Note: When a MASK trace element is created, the current contents of the specified storage location are saved for later comparison. As a result, if the address indicated is outside the user's virtual machine, an error will result.

Examples:

PER Mask INTO 20100 DATA 80	traces a change in the first bit of a byte at location 20100
PER M INTO 20100 DATA ff01	traces a change in any of the 9 indicated bits in a two byte field starting at location 20100
PER M INTO 20100 DATA e0	traces a change in the first, second, or third bits of the byte at location 20100

G[reg1] [{-} [reg2]] [[DATA] hexword]
[{:}]
[{.} [regcount]]

specifies the tracing of the execution of instructions that cause an alteration to general registers. The two operands are optional and positional. The first operand indicates the register(s) to be traced. It follows immediately after the G keyword *without* intervening blanks. You can express this value in either decimal or hexadecimal. If you don't supply a register number, then the default is all 16 general purpose registers. If you specify the register number reg1, then reg1 is a decimal number from 0 to 15 or a hexadecimal integer from 0 to F representing the first, or only, register to be monitored. If you specify reg2, then all the registers from reg1 to reg2 will be monitored. reg2 must be greater than or equal to reg1. If you specify regcount, then regcount is a decimal number from 0 to 15 or a hexadecimal integer from 0 to F representing the number of registers (starting with reg1) to be monitored. The sum of reg1 and regcount must not exceed 15. If -, :, or . is followed by a blank, then all the registers from reg1 to 15 will be monitored. G-reg2 and G:reg2 specify the registers from 0 to reg2 and G.regcount specifies the registers from 0 to regcount-1. G-, G:, and G. are all invalid.

You can further increase the selectivity of the register trace by specifying the data operand (hexword). The DATA keyword is optional. If you specify the DATA keyword, you must also specify hexword. Hexword consists of one to eight hexadecimal characters that will be padded on the left with zeros, as necessary, to complete a fullword. Trace information will be produced when the contents of the specified registers are set equal to that data.

Note: PER recognizes a register alteration event only when the new register contents are different from the previous register contents.

Whenever an alteration occurs, the instruction that altered the register(s) and the new contents of the register(s) are displayed.

Examples:

PER G	traces any change to the contents of registers 0 - 15
PER G5.3	traces any change to the contents of register 5, 6, or 7
PER G DATA 6	traces any register whose value changes to a value of 00000006
PER GC DATA 0	traces register 12 if its value changes to 00000000
PER G12-f DATA 6	traces register 12, 13, 14, or 15 if any of their values change to 00000006

Options

The PER command supports 10 options that can be placed anywhere on the command line, except for CMD (see CMD option).

These options are:

Range	TErminal	GUESTR
FRom	BOth	GUESTV
PAss	RUN	DATOFF
CMd	NORun	DAT
Printer	STEp	

Multiple options can be specified on the command line. Each option on the command line applies to all the events specified on that command line.

Range instruction-addr-range

causes the tracing for a given event type to be limited to instructions executed from within the specified range. See usage note 1 on page 147 for the various ways that you can specify the range. The instruction is considered to be within the specified range if either the first byte of the instruction is within the range or the first byte of the

EX instruction that executes the instruction is within the range. If you do not specify RANGE, then the full range of virtual storage is assumed. If you do specify the RANGE keyword, then the instruction-addr-range is required, and follows the RANGE keyword. You can specify the RANGE keyword on the same command line as an event type. In this instance, the instruction-addr-range will apply to *all* event types on the same command line. If you specify multiple ranges in this case, then each range will apply to all the event types on that command line. The maximum number of ranges that you can specify on a single command line is 8.

Examples:

```
PER STore G Range 20000-20100 Range 20200:20250
```

produces trace information when an instruction (whose first byte is in either of the indicated address ranges) results in an alteration of storage or an alteration of registers.

```
PER BRanch INTO 20250-20300 Range 20100-20200
```

produces trace information when an instruction (whose first byte is between 20100 and 20200) causes a successful branch to a location between 20250 and 20300.

You can specify the RANGE keyword on a command line without specifying an event type. In this case, the instruction-addr-range applies to all elements in the current traceset. It will replace all the instruction-addr-ranges currently in effect. See “Changing Options on Trace Elements” on page 145.

You can also use the RANGE keyword to set multiple address stops. However, unlike the ADSTOP command, the program execution halts *after* the execution of the instruction at the given address. You should note that address stops set using the PER command remain in effect until you end the trace element.

You can set up multiple address stops with PER by using RANGE as an option to the INSTRUCT keyword. The instruction-addr-range, in this case, is a single value corresponding to the address of the instruction where program execution is to be halted.

Examples:

```
PER Instruct Range 20000
```

causes program execution to halt after the instruction at location 20000 executes

```
PER Instruct Range 20000 range 20400
```

causes program execution to halt after the execution of an instruction at either location 20000 or 20400

FRom instruction-addr-range

FROM and RANGE are synonyms and provide identical functions. See the discussion of the RANGE option for details. All output produced for the QUERY display will have the RANGE keyword.

PAss $\left[\begin{matrix} 0 \\ n \end{matrix} \right]$ *— see page 38*

suppresses the display of *n* successful events between displays (*n* must be an integer number between 0 and 32,767). For example, PASS 1 causes every other event to be displayed. If PASS *n* is not specified, PASS 0 is assumed. PASS 0 turns off the PASS option. You cannot specify PASS more than once on a command line.

CMd {text}

specifies a CP command or commands to be executed when the given event occurs. If you use the CMD option, it must be the last option on the line because everything following CMD will be treated as a CP command. The text that follows CMD can be either a single command or multiple commands. However, note that linedit must be on (SET LINEDIT ON) when the PER command is issued in order to set up multiple commands. The escape character ("), followed by the line end character (#) is used to separate multiple commands.

You should not use the CMD option with the SLEEP command. (See usage note 22 on page 152.)

Regardless of the PER options in effect, the output from the CP commands will always be directed to the terminal.

Examples:

```
PER STore INTO 22000.D CMd Display t22000.d
PER BRanch INTO 0 RUN CMd PER END ALL "# Begin 21000
```

Printer
TERminal
BOTH

PRINTER causes trace output to be directed to the virtual printer.

TERMINAL causes trace output to be directed to the terminal. This is the default condition.

BOTH causes trace output to be directed to both the terminal and the virtual printer.

PRINTER and NORUN or PRINTER and STEP are an invalid combination of options. You cannot specify them on the same command line unless you also specify TERMINAL. If you specify PRINTER for any trace elements when NORUN or STEP are already in effect, then the PRINTER option is applied to that trace element and the RUN option is turned on.

Note: The combination BOTH and NORUN and the combination BOTH and STEP are valid combinations. If you specify BOTH on the command line, PRINTER and TERMINAL appear in the QUERY display.

RUN
NORun

STEP $\left[\begin{array}{c} \underline{1} \\ n \end{array} \right]$

RUN causes program execution to continue after trace output has completed. The CP command environment is not entered. RUN is the default if PRINTER (without TERMINAL) is specified.

NORUN causes program execution to stop after the trace output to the terminal. The CP command environment is entered. NORUN is the default if either TERMINAL or BOTH is specified.

STEP causes program execution to stop and the CP command environment to be entered after n displays of trace output. n must be an integer between 1 and 32,767. If you do not specify n , then 1 is the default value for n . You can turn off STEP by specifying either the RUN or NORUN option. You cannot specify STEP more than once on a command line.

STEP, RUN, and NORUN are conflicting options. You cannot specify them on the same command line.

NORUN and PRINTER or STEP and PRINTER are invalid combinations of options and you cannot specify them on the same command line unless you also specify TERMINAL. If you specify NORUN or STEP for any trace element when PRINTER is already in effect, then the NORUN or STEP option is applied to the trace element and the TERMINAL option is turned on in addition to the PRINTER option.

The RUN, NORUN, and STEP options determine whether or not CP mode will be entered regardless of the outcome of any command that may have been executed as a result of the PER CMD option. Different trace elements can exist at the same time with the RUN, NORUN, and STEP options. When an event occurs, if any trace element that is a success indicates that CP mode should be entered, then CP mode is entered.

GUESTR
GUESTV
DATOFF
DAT

PER traces virtual machine activity in both second- and third-level storage. The GUESTR, GUESTV, DATOFF, and DAT options allow the user to selectively limit tracing to just second-level activity or just third-level activity.

GUESTR (or **DATOFF**) causes only second-level activity to be traced. Second-level storage is storage that appears real to your virtual machine

GUESTV (or **DAT**) causes only third-level activity to be traced. Third-level storage is storage that appears virtual to your virtual machine.

If both **GUESTR** and **GUESTV** are specified, or if neither is specified, then both second- and third-level activities will be traced.

Output produced for the **QUERY** display will have the **GUESTR** or **GUESTV** keyword even if **DATOFF** or **DAT** was specified. If both second- and third-level activity is being traced, then neither **GUESTR** nor **GUESTV** will be included in the **QUERY** output. See usage note 20 on page 152.

Sub-Commands

The **PER** command supports five sub-command keywords. You can specify only one sub-command on a command line, and you cannot specify a sub-command on a command line that contains event or option keywords. A sub-command must be the first keyword after **PER** on the command line.

COUNT

causes the successful events to be counted. While the **PER COUNT** option is in effect, no trace output is produced. Whenever you enter a valid **PER** command, the value of the current count is displayed and then the **PER COUNT** is terminated.

If **COUNT** is in effect and you enter **PER count**, the current count is displayed. The **PER COUNT** is not terminated, it is reset to 0.

The **QUERY** command with the **PER** option does not end or reset the **PER COUNT**. The **COUNT** is displayed along with the current traceset.

The **PER COUNT** sub-command is valid only when a current traceset exists.

When the **PER** count reaches 16 million, this count is displayed, and the count will continue from zero.

Table

causes the traceback table containing the last six successful branch instructions to be displayed on the terminal. A traceback table is maintained *only* if a branch trace is in effect. If the traceback table is currently empty, then the message **TRACEBACK TABLE IS EMPTY** is displayed.

SAve traceset-name [APpend]

causes a copy of the current traceset to be saved under the given name until ENDED or the user logs off. A traceset name must be 1-8 characters in length. A traceset name cannot be a number. In addition, the following keywords and their abbreviations cannot be used as traceset names:

ALL	G
APPEND	INSTRUCT
BRANCH	MASK
COUNT	NAMES
CURRENT	STORE

If you specify APPEND, then the current traceset is merged with the named saved traceset. (See "Merging Trace Elements" on page 144.)

If you do not specify APPEND, then the current traceset is saved as the named traceset. In this case, it replaces trace elements that may already be saved under that name.

Example:

If you had the following tracesets defined:

```
query per all
 1 INSTRUCT RANGE 020000-030000 TERMINAL NORUN
 2 BRANCH TERMINAL RUN
SET1
STORE INTO 000250-000300 RANGE 070000-075000 TERMINAL RUN
MASK INTO 020400 DATA FF TERMINAL NORUN
G TERMINAL RUN
```

and you issue the following PER command:

```
PER SAve SET1
```

The saved traceset (SET1) now contains:

```
Query PER SET1
  INSTRUCT RANGE 020000-030000 TERMINAL NORUN
  BRANCH TERMINAL RUN
```

GET traceset-name [APpend]

causes the current traceset to be replaced by a copy of the saved traceset. The saved traceset is unchanged.

If you specify APPEND, then the named traceset is merged with the current traceset. (See "Merging Trace Elements" on page 144.)

If you do not specify APPEND, then the current traceset is replaced by the named traceset.

ENd { ALL
 COunt
 CUrrent
 traceset-name
 element-number
 event-type }

causes the trace elements specified by the operand to be ended. At least one operand is required and must follow the END keyword. You can specify more than one operand on a single command line with END.

ALL eliminates all trace elements in the current traceset and all trace elements in the saved tracesets.

COUNT turns off the PER COUNT option and displays the current count. The current traceset and saved tracesets are not changed.

CURRENT eliminates all trace elements in the current traceset. The saved tracesets are not affected.

traceset-name eliminates all trace elements in the saved traceset.

element-number eliminates trace elements in the current traceset that correspond to the element number. This number is supplied to you as a result of the QUERY PER command.

event-type eliminates any trace elements in the current traceset that match the event-type specified. The event-type can be any of the possible EVENT keywords or their abbreviations (for example, BR, I, G, ST, M).

Note: The elements in the current traceset are renumbered following the execution of the PER END element-number or PER END event-type sub-command.

Merging Trace Elements

Trace elements within a traceset are merged whenever the following conditions are satisfied:

- The event types are the same,
- The options PASS, STEP, RUN, NORUN, PRINTER, TERMINAL, CMD, GUESTR and GUESTV match,
- The DATA (if any) for both trace elements are equal,
- The INTO and FROM ranges, or the registers and FROM ranges are such that the trace elements can be merged to form a trace element that is equivalent to the original two.

For example, if the current traceset contains:

```
1 STORE INTO 023048 DATA FF88 RANGE 020000-021000 TERMINAL NORUN
2 BRANCH INTO 021000-021008 RANGE 020000-021000 TERMINAL NORUN
```

and you enter the following:

```
PER STore INTO 23048 DATA ff88 r 21000-22000
```

the ranges on the STORE are merged and the current traceset becomes:

```
1 STORE INTO 023048 DATA FF88 RANGE 020000-022000 TERMINAL NORUN
2 BRANCH INTO 021000-021008 RANGE 020000-021000 TERMINAL NORUN
```

Changing Options on Trace Elements

If you specify any of the options (RANGE, FROM, PASS, CMD, PRINTER, TERMINAL, BOTH, RUN, NORUN, STEP, GUESTR, GUESTV, DATOFF, or DAT) on a PER command line without any event types, then the options specified replace the options currently in effect for the current traceset.

For example, if the current traceset contains:

```
1 G2 G4 RANGE 020000-024000 TERMINAL NORUN
2 INSTRUCT 41 RANGE 020000-021000 PRINTER TERMINAL RUN
3 STORE INTO 023034-023100 RANGE 020000-024000 PRINTER RUN
```

and you issue the command:

```
PER Range 24000-25000 PAss 3 Printer RUN
```

then the range 24000-25000 replaces the from ranges 20000-21000 and 20000-24000 (however, the storage address range 23034-23100 is not changed). The PASS 3 option is applied, the PRINTER option replaces the TERMINAL option and the RUN option replaces the NORUN option. The resulting current traceset is:

```
1 G2 G4 RANGE 024000-025000 PRINTER RUN PASS 3
2 INSTRUCT 41 RANGE 024000-025000 PRINTER RUN PASS 3
3 STORE INTO 023034-023100 RANGE 024000-025000 PRINTER RUN PASS 3
```

Changing Specific Trace Elements

The options in effect for a specific trace element in a traceset are replaced by the options specified on a new trace element whenever the following conditions are satisfied:

- The event types are the same,
- The DATA (if any) for both trace elements are equal,
- The FROM ranges, INTO ranges (if any), and registers specified (if any) are equal.

For example, if the current traceset contains:

```
1 G2 G4 RANGE 020000-024000 TERMINAL NORUN
2 INSTRUCT 41 RANGE 020000-021000 PRINTER TERMINAL RUN
3 STORE INTO 023034-023100 RANGE 020000-024000 PRINTER RUN
```

and you issue the command:

```
PER STore 23034-23100 Range 20000-24000 PAss 4
```

then the resulting current traceset is:

```
1 G2 G4 RANGE 020000-024000 TERMINAL NORUN
2 INSTRUCT 41 RANGE 020000-021000 PRINTER TERMINAL RUN
3 STORE INTO 023034-023100 RANGE 020000-024000 TERMINAL NORUN PASS 4
```

Note that although not explicitly specified, the default options **TERMINAL** and **NORUN** have also been applied to the third trace element.

For the **PER GET** subcommand with the **APPEND** option, whenever a trace element in the saved traceset matches a trace element in the current traceset as described above, the options from the trace element in the saved traceset replace the options in the trace element in the current traceset (that is, the two tracesets are merged).

For the **PER SAVE** subcommand with the **APPEND** option, whenever a trace element in the current traceset matches a trace element in the saved traceset as described above, the options from the trace element in the current traceset replace the options in the trace element in the saved traceset (that is, the two tracesets are merged).

Example:

If you have the following tracesets defined:

```
Query PER ALL
  1 INSTRUCT RANGE 020000-030000 TERMINAL NORUN
  2 BRANCH TERMINAL RUN
SET1
  INSTRUCT RANGE 030000-040000 TERMINAL NORUN
  BRANCH TERMINAL NORUN
```

and you issue the following PER command:

```
PER SAve SET1 APpend
```

the current traceset would merge with the elements already in SET1 to produce:

```
query per set1
  INSTRUCT RANGE 020000-040000 TERMINAL NORUN
  BRANCH TERMINAL RUN
```

Usage Notes

1. You must specify all address range values in hexadecimal. You can do this in any of the following ways:

- `addr`

specifies a single byte in storage.

- `addr1-addr2` or `addr1:addr2`

specifies all storage locations from `addr1` to `addr2` inclusive.

Address ranges can wrap around. If the second address in the range is lower than the first, the storage locations specified are from the first address to `FFFFFF` and from 0 to the second address.

- `addr.hexlength`

specifies a hexlength number of contiguous storage locations starting at `addr`. (Hexlength is a hexadecimal number.)

Addresses can wrap around. If the hexlength (when added to `addr`) goes beyond `FFFFFF`, the storage locations specified are from `addr` to `FFFFFF` and from 0 to the remainder of the hexlength.

- `addr-addr`, or `addr`.
specifies all storage locations from `addr` to `FFFFFF` inclusive.
- `-addr`, or `:addr`
specifies all storage locations from 0 to `addr` inclusive.
- `.hexlength`
specifies all storage locations from 0 for the number of bytes specified.

If you don't specify a range, a range of 0-FFFFFF is the default.

It is invalid to only specify a separator (`.` or `:` or `-`).

2. When tracing storage alterations within a specific range (concurrently with other storage alteration ranges, mask, or instruction trace elements), the execution of a MVCL instruction can cause the MVCL instruction to be displayed even though it did not cause a storage alteration in the range specified.
3. The execution of the interruptable instructions MVCL, CLCL, and TB can result in multiple PER interrupts. This can result in multiple displays of the same instruction.
4. If a storage area is being monitored by a mask trace element, then the use of the STORE command to change that storage area after the creation of the mask trace element can cause erroneous trace output.
5. Self-modifying instructions or instructions that change the contents of registers used to compute operand addresses cause erroneous processing of trace output.

Examples:

MVCINST MVC	MVCINST(6),DATA	self-modifying instruction
L	12,0(6,12)	this instruction changes register 12 which is used to compute the operand address

6. STORE traces only instructions executed in the virtual machine that the PER hardware recognizes as altering storage. Other alterations are not traced.

For example:

- Execution of a channel program that reads data from an I/O device into storage is not traced as a storage alteration event.
- Alteration of low storage due to an interrupt (for example, SVC, PROG, I/O, External) is not traced. These interrupts can be traced using the TRACE command.
- The storing of the CSW for I/O operations is not traced.
- The updating by CP of the interval timer is not traced.
- The alteration of storage caused by the following CP simulated instruction is not traced: DIAGNOSE and IUCV.

However, you can trace the alteration of low storage caused by instructions executed in your virtual machine.

7. A register alteration caused by the following CP simulated instructions are not traced by PER: DIAGNOSE and IUCV.

When the instruction display is produced for the next instruction that alters a monitored register, the contents of the register(s) altered by the DIAGNOSE or IUCV instruction can also be displayed.

8. The default range 0-FFFFFF or its equivalent (for example, RANGE 5-4) is not displayed in the QUERY PER output even if you explicitly specified it in the PER command. The default PASS 0 is not displayed in the QUERY PER output even if explicitly specified.
9. An address range with beginning and end addresses equal to each other is considered to be a single address value. For example, the range 20-20 is the same as the single address 20.
10. For MASK, the storage value is saved when the original command is issued and updated whenever an instruction alters that storage. If a saved traceset containing a MASK trace element is retrieved using the PER GET command, then the storage values used for the next comparison may not be current.
11. Message DMKPER182I (PER TRACE IS ACTIVE) informs you each time 10,000 PER interrupts have occurred without output to the terminal. You can prevent this message by issuing the CP command SET IMSG OFF.

12. When tracing in a virtual machine with only one printer, the trace data is intermixed with other data sent to the virtual printer. To separate trace information, define another printer with a lower virtual address than the previously defined printer. In this case, the PER output will go to the printer with the lower virtual address. However, there is no way to prevent output from PER and TRACE from being mixed.
13. In order to use the line end character for multiple commands, line edit must be on before you issue the PER command.

When you have entered multiple commands properly, then QUERY PER always includes the current line end character in the display of multiple commands.

14. One command buffer is created for each PER command line with the command option specified. For example:

```
PER I BR Cmd Display g0
```

creates one command buffer shared by the instruction and branch trace elements, and

```
PER I Cmd Display g0  
PER BR Cmd Display g0
```

creates 2 separate command buffers. If a branch occurs, then both the instruction and branch trace elements will be successes. In the first example, the single command buffer is executed resulting in register 0 being displayed once. In the second example, both command buffers are executed, resulting in register 0 being displayed twice.

When multiple trace elements with the command option are successes at the same time, then the commands are executed in the order specified by the QUERY PER command.

15. If you issue the command:

```
PER G2 G4 DATA 78
```

the data only applies to G4.

The resulting traceset would be displayed by the QUERY command as follows:

```
1 G2 TERMINAL NORUN  
2 G4 DATA 00000078 TERMINAL NORUN
```

To have the data apply to both G2 and G4, you would issue

```
PER G2 DATA 78 G4 DATA 78
```

The QUERY command would display this as:

```
1 G2 G4 DATA 00000078 TERMINAL NORUN.
```

16. The execution of each instruction will only be counted once, regardless of how many trace elements are successes. For example:

```
PER I BR R 20000.1000
PER COUNT
```

The execution of a successful branch instruction in the range 20000-20FFF would add 1 to the count even though both the INSTRUCT and BRANCH trace elements were successes.

17. PASS suppresses the display of successful events. PASS *does not* make the event unsuccessful. Therefore, the PASS option has no effect on the count.
18. Trace elements are merged after the options of matching trace elements are changed. For example, if the current traceset contains:

```
1 INSTRUCT RANGE 020000-022000 TERMINAL RUN
```

and you issue the command:

```
PER I R 20000-22000 R 22000-24000 TERminal NORun
```

then the resulting current traceset would be:

```
1 INSTRUCT RANGE 020000-024000 TERMINAL NORUN
```

However if the current traceset contains:

```
1 INSTRUCT RANGE 020000-022000 TERMINAL RUN
```

you issue the command:

```
PER I R 20000-21000 R 21000-22000 TERminal NORun
```

then the resulting current traceset would be:

```
1 INSTRUCT RANGE 020000-022000 TERMINAL RUN
2 INSTRUCT RANGE 020000-022000 TERMINAL NORUN
```

See the explanation of the RUN, NORUN, and STEP options on page 141.

19. PER uses the real control registers 9, 10, and 11. Therefore, these control registers are not available to the virtual machine while that virtual machine is using PER. Programs running in the virtual machine can still load virtual control registers 9, 10, and 11, but the real control registers 9, 10, and 11 are not loaded with the contents of the virtual control registers when the virtual machine is dispatched. While PER is active, all PER interrupts are interpreted by CP. No PER interrupts are reflected to the virtual machine.

Note: When running in single processor mode or as a preferred machine assist guest, do not use the PER command because MVS has control of real registers 9, 10, and 11.

20. The PER command allows you to indicate address ranges as operands for certain keywords. PER traces virtual machine activity in both second- and third-level storage. Second-level storage is storage which appears real to your virtual machine. Third-level storage is storage which appears virtual to your virtual machine.

For example, if you are running a BC mode virtual machine (for example, CMS) or an EC mode virtual machine with the DAT (data address translation) feature off (for example, second-level CP), then the addresses are all second-level addresses. If you are running an EC mode virtual machine with the DAT feature on, then the addresses specified are third-level addresses.

21. The maximum number of trace elements that you can establish is 255. An error condition results if the intermediate processing of a PER command would require more than 255 trace elements to exist even if the end result would be less than 255 elements.
22. You should not use the SLEEP command with the PER CMD option. If the SLEEP command is issued from the CMD option while PER is active, then the SLEEP command is ignored. If the SLEEP command is issued from the CMD option while PER is being ended, the SLEEP command may or may not be ignored depending on how PER is terminated. While PER output is being produced the SLEEP command will be executed only if entered while in the CP command environment.
23. If the current traceset is altered during instruction display, then the current instruction display may be halted before it completes.

Responses

The following symbols are used in the responses received from PER:

Symbol	Meaning
vvvvvv	virtual storage address
ttttt	virtual transfer address
xxxxxxxxxxxx	virtual instruction
zzzzzzzz	register contents
oooooo	virtual operand address
yy	low order byte of register in an execute instruction
dd	hex-data
mnem	mnemonic for instruction
CC = n	condition code number
:	traceback table entry
= = >	sequence indicator
n	an integer
l	level indicator; if l is blank, then vvvvvv is a second-level storage location. If l is "V", then vvvvvv is a third-level storage location.

BRANCH INSTRUCTIONS:

lvvvvvv mnem xxxxxxxx ttttt CC = n

RR INSTRUCTIONS:

lvvvvvv mnem xxxx CC = n

RX, RS, or SI INSTRUCTIONS:

lvvvvvv mnem xxxxxxxx oooooo CC = n

SS TYPE INSTRUCTIONS:

lvvvvvv mnem xxxxxxxxxxxxxx oooooo oooooo CC = n

EXECUTE INSTRUCTIONS:

lvvvvvv EX xxxxxxxx yy (instruction output as above)

BRANCH TRACEBACK:**TRACEBACK TABLE:**

:lvvvvvv mnem xxxxxxxx ttttt n TIMES

:(up to 6 entries are included)

Note: Entries are put at the bottom of the traceback table. When the table is full, the new entry is put at the bottom of the table, and the top entry in the table is eliminated.

n TIMES is the number of times in a row that a branch instruction was successfully executed with no other successful branch instruction in between. This value is not displayed when n equals 1. The maximum value of n is $2^{16}-1$.

GENERAL REGISTER ALTERATION:

The following is appended to the instruction output that is displayed:

Gnn=zzzzzzzz

This appears for each register that is to be displayed.

STORAGE ALTERATION OUTPUT:

When storage alterations are being monitored and a storage alteration event occurs, then *STORE* is appended to the instruction output. If you specified hex-data in the PER STORE or PER MASK command, then the following is appended to the instruction output that is displayed:

DATA AT vvvvvv=dddddd

This appears for each data trap that is a success.

SEQUENCE INDICATOR:

A sequence indicator (= = >) is displayed when the displayed instruction is not in sequence. This occurs if:

1. A successful branch occurred and the displayed instruction was branched to.
2. The PASS option was in effect and the display of the previous instruction was suppressed as a result.
3. The RANGE option was in effect and the display of the previous instruction was suppressed as a result.

When using the PASS or RANGE options, the sequence indicator is not displayed on the instruction following a branch if the instruction branched to is the next sequential instruction following the last displayed instruction.

PER TRACE ENDED

All PER traps associated with the current traceset and any saved tracesets have been eliminated.

TRACEBACK TABLE IS EMPTY

There are no entries in the traceback table.

PER COUNT n

The number of successful PER events (based on the current traceset) is n (a decimal number).

PER COUNT ENDED

The PER COUNT option has been terminated. Successful events are no longer being counted. Output is now determined by the options in the current traceset.

PER COUNT RESET TO 0

The PER COUNT has been reset to 0. Successful events are still being counted.

PURGE

Privilege Class: G

Use the PURGE command to remove your own closed spool files from the system before they are printed or punched by the spooling devices, or before they are read by a user.

The format of the PURGE command is:

PURge	{	Reader RDR Printer PRT Punch PCH ALL	}	¹
		CLASS c1 CLASS c2... FORM form1 FORM form2... spoolid1 spoolid2... ALL		
¹ A combination of CLASS, FORM and <i>spoolid</i> specifications may be entered.				

where:

[READER]
[RDR]

purges reader files.

[PRINTER]
[PRT]

purges printer spool files.

[PUNCH]
[PCH]

purges punch spool files.

ALL

purges all reader, printer, and punch spool files. When ALL is specified instead of a device type, all other operands are ignored.

CLASS c1 CLASS c2...

are the class(es) of files to be purged. CLASS is a required reserved word and c1,c2,... are one-character alphameric fields (with values from A to Z and 0 to 9) that represent spooling classes.

FORM form1 FORM form2...

specifies that CP should purge all files with the specified form names defined during system generation. The *form1* and *form2* options are one-to-eight character form numbers.

spoolid1 spoolid2...

are the *spoolids* of specific files to be purged.

ALL

purges all files of the specified type (reader, printer, or punch).

PURGE

Usage Notes

1. You can purge any closed spool file you own, regardless of its status, as long as CP has not selected it for processing.
2. You can purge files using a combination of CLASS, FORM, and *spoolid* specifications. For example:

```
PURge Printer CLass A 1932 CLass D 619 FORM INVOICE
```

specifies that CP should purge all class A and class D printer files, printer files with *spoolids* of 1932 and 619, and all printer files with form name INVOICE.

3. You cannot purge any file you spool to the reader of another user unless you first reclaim it using the TRANSFER command.

Response

{nnnn} FILES PURGED
{NO }

indicates the number of files purged. This response is not displayed if you issued the CP SET IMSG OFF command.

QUERY

Privilege Class: G and all classes except class Any

Use the class G QUERY command to determine your system status and machine configuration by requesting:

- The time you have used during a terminal session.
- The number of closed input and output spool files associated with your virtual machine. For information on how to alter spool file attributes, see the CHANGE command.
- The current settings of the color and/or extended highlight values in effect for your virtual machine console.
- The current settings of the SET command functions.
- The current settings of the TERMINAL command functions.
- The status of all the devices on your virtual machine.
- The channel operating mode of your virtual machine, whether block-multiplexer or selector.
- A listing of all users who are linked to a given virtual address, together with their device addresses and access modes.
- Display of the secondary user (*secuser*) that is specified in the CONSOLE directory statement.
- Identification and attributes associated with your virtual printer, punch, and reader spool files.
- The identification of your virtual processor.
- The mode of processor operation of your installation: uniprocessor mode (UP), attached processor mode (AP), or multiprocessor mode (MP).
- System software information.
- The log messages of the day.
- The names of the users that are logged on.
- The number of users that are logged on or dialed to the system.
- The *userid* and system identifier.
- A listing of the PER traceset elements.

There are other operands you can use with the QUERY command if you have the privilege class required to use them. These are described in the *VM/SP HPO Operator's Guide*. Also, if you are a CMS user, you can use the CMS QUERY command to query the status of your CMS virtual machine.

Note: Use the NETWORK QUERY command to query 3704/3705/3725 resources and remote 3270 devices attached to a 2701 or 2703 or to a 3704/3705/3725 in EP or shared PEP mode.

QUERY

The format of the Class G QUERY command is:

Query	<pre> Time Set TERMinal Files [Class c] [FORM form] [NOHold¹ HOld USERhold SYShold] [Virtual] [Channels GRaf CONsole DAsd TAPes LINES UR STORAge ALL vaddr[-vaddr]] Links vaddr CPUid SECuser [Reader] [RDR] [Printer] [PRT] [Punch] [PCH] } [Class c][FORM form] spoolid [NOHold HOld USERhold SYShold] } ALL[TBL]¹ PF[nn] PROCEssr SCREen S370E SPMODE VMSave CPLEVEL LOGmsg Names Users [userid] userid PER [Current Names traceset-name All] </pre>
-------	---

¹The options may be specified in any order.

where:

TIME

displays the current time, time zone, weekday, date, connect and processor time for the current terminal session.

SET

displays the status of the SET command functions.

TERMINAL

displays the current options in effect for your virtual console environment.

FILES

displays the number of spooled input and output files for your virtual machine. Files currently being processed are not included in the totals. The CLASS, FORM, and HOLD, NOHOLD, SYSHOLD, or USERHOLD operands restrict counting to certain spool files.

VIRTUAL

displays the status of all virtual devices.

CHANNELS displays the channel mode of operation for the virtual machine.

GRAF displays the status of all display devices and graphics devices that are locally and remotely attached to your virtual machine. It also displays the SNA *luname* and VTAM service machine (VSM) *userid* for each SNA device accessing your virtual machine using the CP DIAL command.

CONSOLE displays the status of your virtual consoles. It also displays the SNA *luname* and the VSM *userid* for SNA user's console devices.

DASD displays the status of all your virtual direct access storage devices.

TAPES displays the status of all your virtual magnetic tape devices.

LINES displays the status of all your virtual communications lines.

UR displays the status of all your unit record devices and the pseudo timer.

STORAGE displays the size of your virtual storage.

ALL displays the status of all your virtual devices.

vaddr displays the status of the virtual device at address *vaddr*.

vaddr-vaddr displays the status of the devices within the range of addresses specified.

LINKS vaddr

displays the *userid*, device address, and access mode at the terminal for all users linked to the specified virtual address (*vaddr*). A range of virtual addresses is not supported by the LINKS operand. A link is indicated for those users with virtual devices that have the same starting cylinder or block on the same real device.

CPUID

displays the 16-digit processor identification that is in use by the virtual machine. Values within the CPUID can be set by the directory OPTION statement or modified by the SET CPUID.

SECUSER

displays the *userid* of the secondary user that is specified in the CONSOLE directory statement.

[READER]
[RDR]
[PRINTER]
[PRT]
[PUNCH]
[PCH]

displays the following information, pertaining to your closed virtual reader, virtual printer, and virtual punch spool files:

- *UserId* (of user who created the file)
- Spool file identification (spoolid)
- Class and originating device type
- Number of logical records in the file
- Number of copies specified for the file (has no effect for reader files)
- File hold status
- Form name

One line of information is displayed for each spool file (spool file specified by spoolid).

ALL

displays additional information for spool files. The ALL operand must follow the READER, PRINTER, or PUNCH operand. In addition to the information normally displayed for the reader, printer, or punch files, the following is also displayed:

- Date and time the file was created
- Filename and filetype of file (if any); if your file was assigned a dsname and you later issue QUERY, only the first 18 characters of the 24-character field are displayed.
- Distribution code of the file

Note: The form name is not displayed when the ALL option is used.

One line of information is displayed for each spool file of the type specified.

TBL

displays additional information about spool files pertaining to the 3800 printer. This option must follow the **READER**, **PRINTER**, or **PUNCH** operand. In addition to the information normally displayed, the following is also displayed:

- Character arrangement table
- Vertical format (FCB)
- Copy modification module
- Forms overlay frame
- Number of copies to be printed with forms overlay frame in place
- The placement of 3800 LOAD channel command words (CCWs) within a 3800 spool file

Note: The form name, number of copies, and hold status are not displayed when you use the TBL option.

One line of information is displayed for each print file specified.

spoolid

selects the specified spool file for display
(**READER/PRINTER/PUNCH** option)

CLASS c

selects spool files of class *c* for counting (**FILES** option) or display
(**READER/PRINTER/PUNCH** option)

FORM form

selects spool files with a particular form name for counting (**FILES** option) or display (**READER/PRINTER/PUNCH** option)

NOHOLD

selects spool files that are not held (either by system hold or user hold) for counting (**FILES** option) or display
(**READER/PRINTER/PUNCH** option)

HOLD

selects spool files that are held (either by system hold or user hold) for counting (**FILES** option) or display (**READER/PRINTER/PUNCH** option)

USERHOLD

selects spool files that are in user hold for counting (**FILES** option) or display (**READER/PRINTER/PUNCH** option)

SYSHOLD

selects spool files that are in system hold for counting (**FILES** option) or display (**READER/PRINTER/PUNCH** option)

PF nn

displays the program function key number specified for a 3270 Information Display device, along with its associated command lines. If you do not specify *nn*, CP displays all 24 program function keys and their associated data lines. The value, *nn*, is a number from 1 (or 01) to 24. See the CP SET command for an explanation of how to define and use program function keys.

PROCESS

displays for the user which processors are currently online.

SCREEN

displays the color and extended highlight values currently in effect for your virtual console. These values are set either by the SCREEN statement in your directory entry or when you issue class G SCREEN command(s).

S370E

indicates that MVS/System Extensions or MVS/System Product support is active or inactive (not available) on the system.

SPMODE

displays whether single processor mode is ON or OFF for VM/SP HPO.

VMSAVE

displays information about all of the target VMSAVE areas specified in module DMKSNT of which this user is either the owner or receiver.

CPLEVEL

displays the system software product name, release number service level, nucleus generation time and date, and IPL time and date.

LOGMSG

displays the log messages of the day.

NAMES

displays a list of all the users logged on and the real address of the line to which each is connected. If a user is disconnected, the abbreviation DSC is printed instead of the line address.

USERS

displays the number of logged on users and the number of users logically connected to other virtual machines.

USERS userid

userid

displays the user identification and the terminal device address of the specified user if he is logged on. If the user is not logged on, a message to this effect is issued. Use the QUERY USERS userid format if the userid is the same as an operand of the QUERY command (for example, TAPES).

Note: It is possible for the number of users logged on as indicated by the "NAMES" operand to differ from the number of users indicated by the "USERS" operand if the QUERY command is executed while users are in the process of logging on or logging off the system.

PER [C U r r e n t
N a m e s
t r a c e s e t - n a m e
A l l]

displays PER command tracesets and trace elements.

CURRENT displays the current traceset elements and any active PER count.

NAMES displays the names of all saved tracesets.

traceset-name displays the traceset elements for the specified traceset.

ALL displays all the current traceset elements, the value of an active PER count, and the name and contents of all the saved tracesets.

Responses

This section describes the messages that CP issues in response to your specific QUERY operands.

QUERY TIME

TIME IS hh:mm:ss zone weekday mm/dd/yy

The current real clock time in hours:minutes:seconds, the time zone (for example, EST), the day of the week and the calendar date (month/day/year) are displayed.

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

The time spent in the current terminal session is displayed.

where:

CONNECT = hh:mm:ss

is the actual clock time spent in the current terminal session in hours:minutes:seconds.

QUERY

VIRTCPU = mmm:ss.hs

is the virtual processor time used in the current terminal session in minutes:seconds.hundredths of seconds.

TOTCPU = mmm:ss.hs

is the total processor time (virtual and overhead) used in the current terminal session in minutes:seconds.hundredths of seconds.

QUERY SET

MSG {ON
OFF } IUCV , WNG {ON
OFF } IUCV , EMSG {ON
OFF
CODE
TEXT } IUCV , ACNT {ON
OFF } , RUN {ON
OFF }

LINEDIT {ON
OFF } , TIMER {ON
OFF
REAL } , ISAM {ON
OFF } , ECMODE {ON
OFF }

ASSIST {ON
OFF
NOT ACTIVE } {SVC
NOSVC } {TMR
NOTMR } , PAGEX {ON
OFF } , AUTOPOLL {ON
OFF }

IMSG {ON
OFF } IUCV , SMSG {ON
OFF } IUCV , AFFINITY {NONE
nn , } NOTRAN {ON
OFF } , VMSAVE {ON
OFF } , 370E {ON
OFF [XMEM] }

STBYPASS {VR
NNNNNK } {OFF } , STMULTI {NN/QQ xx/YY
OFF }

MIH {ON
OFF } , VMCONIO {OFF
IUCV } , CPONIO {OFF
IUCV }

QQ in the STMULTI response is the actual number of shadow tables being used. With the exception of QQ, all of the response values are the result of and explained with the SET command.

XMEM in the QUERY SET response means that MVS/SP cross memory services assist is enabled for the virtual machine.

The settings of all functions controlled by the SET command and the directory ISAM and ECMODE options are displayed. Refer to the discussion of the SET command for explanation of these functions.

QUERY TERMINAL

LINEND {ⁿOFF}, LINEDEL {ⁿOFF}, CHARDEL {ⁿOFF}, ESCAPE {ⁿOFF}, TABCHAR {^{ON}OFF
 {^{CHAR}}
 LINESIZE {ⁿⁿⁿOFF}, ATTN {^{ON}OFF}, APL {^{ON}OFF}, TEXT {^{ON}OFF}, MODE {^{CP}VM}, HILIGHT {^{ON}OFF}
 CONMODE {³²¹⁵3270}, BREAKIN {^{IMMED}GUESTCTL}, BRKKEY {^{PA1}PF1
 {^{PF24}}}, SCRNSAVE {^{ON}OFF}
 TYPE {³¹⁰¹TTY}, PROMPT {^{VM}TTY}, SCROLL {ⁿⁿⁿCONT}

The settings of all functions that are controlled by the **TERMINAL** command are displayed. Refer to the discussion of the **TERMINAL** command for explanations of the functions. If **LINEDIT** is turned off, the logical editing symbols displayed are those that were in effect before line editing was turned off.

*Note: If the user has no physical console, only the first line is issued. Also, the responses after the **HILIGHT** message do not appear when there is no virtual console defined.*

QUERY CPUID

CPUID= aabbbbbbbccccddd

where:

aa

is the 16-digit processor identification associated with the virtual machine.

aa

is the version code; these two digits are forced to X'FF' to identify that the virtual machine is running under VM/SP HPO.

bbbbbb

is the processor identification number; this field contains six hexadecimal digits. This is the only part of the CPUID that can be modified by means of the **SET CPUID** command or set by the directory's **OPTION** control statement.

cccc

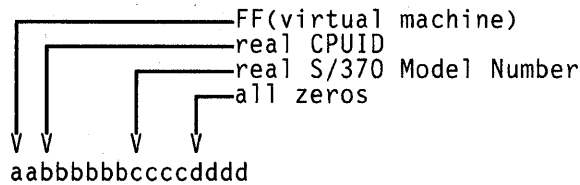
is the model number of the processor you are running on; this field is padded to the left with zeroes when you are not running on a 308x or 4341 processor. This field defaults to the model number of the real machine. In MP mode, this value is the model number stored in the PSA (prefix storage area) of the IPL processor. If the IPL processor is not online at the time of the **QUERY**, CP displays the model number of the processor that is online at the time the **QUERY** command is issued.

QUERY

dddd

is the machine check extended logout; this field is forced to X'0000' since CP does not reflect machine checks back to the virtual machine.

Note: If the CPUID is not supplied by the directory OPTION statement or the SET CPUID command, then the CPUID associated with virtual machine(s) is as follows:



QUERY FILES

FILES: { nnn } RDR, { nnn } PRT, { nnn } PUN
 { NO } { NO } { NO }

CP displays the total number of spool files in your system. If you specify the CLASS, FORM and/or NOHold, Hold, USERhold, or SYShold options with QUERY FILES, CP displays only the totals for the options you specify rather than for all files on your system.

QUERY VIRTUAL CHANNELS

CHANNELS= { SEL }
 { BMX }

The operating mode of the virtual machine channels is displayed. This response applies to all of the virtual machine channels except channel 0, which is always a byte multiplexer channel.

QUERY VIRTUAL GRAF

GRAF vaddr { ON luname } VSMvvvvvvvvv
 { ON DEV raddr }
 { NOT READY }

The status of all locally attached virtual display devices defined to your virtual machine is displayed. The SNA *luname* and VTAM service machine (VSM) *userid* for each SNA device accessing your virtual machine using the CP DIAL command are also displayed.

where:

vaddr

is the virtual address to which the device is attached.

raddr

is the real address (*cuu*) of the device.

NOT READY

shows the status of a virtual display device that has not been attached via the DIAL command.

luname

is the logical unit name of the SNA device accessing your virtual machine.

vvvvvvvvv

is the *userid* of the VSM controlling the SNA device.

QUERY VIRTUAL CONSOLE

```

CONS vaddr ON {
    VSM=vvvvvvvvv
    luname
    GRAF
    LINE raddr } { TERM } { STOP }
                { NOTERM } { START }
                DISCONNECTED

vaddr CL c { CONT } { HOLD } COPY nnn { READY }
           { NOCONT } { NOHOLD }        { NOTREADY } FORM form

vaddr { TO }
      { FOR } userid DIST distcode FLASHC ccc

vaddr FLASH ovly CHAR ctab MDY cmod FCB fcb
    
```

For virtual machine consoles and SNA terminals, a four-line response is displayed. The first line shows the console status and options and the next three lines show the virtual console spooling status.

where:

vaddr

is the virtual address of the virtual machine console.

raddr

is the real address (*cuu*) of the terminal associated with the virtual console.

c

is the spooling class of the console.

nnn

is the number of copies spooled.

userid

is the user identification.

distcode

is the distribution code.

ovly

is the name of the forms overlay frame superimposed on the output pages.

ctab

is the name of the character arrangement table used to generate output.

cmod

is the name of the copy modification module used to alter output text.

fcv

is the name of the FCB module used in the vertical formatting of a page.

ccc

is the flash count for the file.

luname

is the logical unit name for the SNA terminal simulating a virtual console. The response to the QUERY VIRTUAL CONSOLE command displays the *luname* rather than the real address (*raddr*) and the GRAF/LINE option. The variable *luname* is a one-to-eight character identifier assigned to a SNA terminal user and used by CP instead of *raddr*. It is valid for SNA terminals connected to the host VM/SP HPO system by either the Group Control System (GCS/VTAM/VSCS) or VTAM Communications Network Application (VM/VCNA) program product.

vvvvvvvv

is the *userid* of the VTAM service machine (VSM) when you are connected through Group Control System.

The other fields indicate the setting of the respective options in the SPOOL command.

The default settings for a virtual console are:

```
CONS vaddr ON DEV raddr TERM STOP
      vaddr CL T NOCONT NOHOLD COPY 001 READY
      vaddr FOR userid DIST distcode FLASHC 000
      vaddr FLASH CHAR MDFY FCB
```

The default settings for a SNA terminal simulating a virtual console are:

```
CONS vaddr luname TERM STOP
      vaddr CL T NOCONT NOHOLD COPY 001 READY
      vaddr FOR userid DIST distcode FLASHC 000
      vaddr FLASH CHAR MDFY FCB
```

QUERY VIRTUAL DASD

DASD vaddr type volser $\left\{ \begin{array}{l} \text{R/W} \\ \text{R/O} \end{array} \right\} \left\{ \begin{array}{ll} \text{nnnnnn} & \text{BLK} \\ \text{nnn} & \text{CYL} \end{array} \right\}$

The status of each virtual minidisk defined for your system is displayed.

where:

vaddr

is the virtual address to which the DASD device is attached.

type

is an IBM direct access device. If 3340 is displayed, this indicates that the device is either a 3340 or 3344.

For a dedicated DASD unit, the response is:

DASD vaddr ON DASD raddr

where:

vaddr

is the virtual address to which the FB-512 device is attached.

raddr

is the real address (*cuu*) of the FB-512 device.

volser

is the volume serial number of the system disk on which this virtual disk resides.

R/W

R/O

indicates the read/write status of the disk.

nnn

is the number of cylinders on the virtual disk for count-key-data devices.

nnnnnn

is the number of blocks on the virtual disk for FB-512 devices.

For 3880-11 and 3880-21 Storage Subsystem devices, the status of each 3350 comprising the 3880 Storage Subsystem is displayed as follows:

DASD raddr ATTACHED TO userid vaddr
pagaddr1 pagaddr2 pagaddr3 PAGING ADDRESSES

where:

raddr

is the real address (*cuu*) of the DASD.

QUERY

userid

is the identification of the user to whose virtual machine the real device is logically attached.

vaddr

is the virtual base address used for direct mode I/O to a DASD that is a part of the 3880-11 and 3880-21 Storage Subsystem.

pagaddr1**pagaddr2****pagaddr3**

are the virtual paging addresses used for paging I/O to the device at *vaddr*.

If the response to a QUERY DASD indicates a 3880-11 or 3880-21 Storage Subsystem device, the response also identifies all paging addresses in the form *cuu*.

If you are using a 3880 Model 11, the response to:

```
QUERY DASD
```

when the base address is 280 will be:

```
DASD 280 ONLINE
      288 290 298 PAGING ADDRESSES
```

If you are using a 3880 Model 21, the response to:

```
QUERY DASD
```

where the base address is 280 will be:

```
DASD 280 ONLINE
      284 288 28C PAGING ADDRESSES
```

QUERY VIRTUAL TAPES

```
TAPE vaddr ON DEV raddr
```

The status of each tape defined for your system is displayed.

where:

vaddr

is the virtual address to which the tape is attached.

raddr

is the real address (*cuu*) of the tape.

QUERY VIRTUAL LINES

LINE vaddr ON DEV raddr

The status of all communication lines defined in your virtual machine is displayed.

where:

vaddr

is the virtual address to which the line is attached.

raddr

is the real address (*cuu*) of the line.

LINE vaddr { ENABLED }
 { DISABLED }

The status of virtual communication lines at virtual address *vaddr* is displayed.

QUERY VIRTUAL UR

RDR vaddr CL c { CONT } { HOLD } { EOF } { READY }
 { NOCONT } { NOHOLD } { NOEOF } { NOTREADY }

The status of all the virtual readers attached to your virtual machine is displayed.

where:

vaddr

is the virtual device address of the virtual reader.

c

is the spool file class which the device services. A class of * indicates the device serves all classes of spool files for input.

The other fields indicate the setting of the respective options in the SPOOL command.

QUERY

The default settings for a reader are:

```
RDR vaddr CL * NOCONT NOHOLD READY EOF
```

```
PUN vaddr CL c { CONT } { HOLD } CPY nnn { READY }  
          { NOCONT } { NOHOLD } { NOTREADY } FORM form
```

```
vaddr { TO }  
      { FOR } userid DIST distcode
```

```
PRT vaddr CL c { CONT } { HOLD } CPY [*] nnn { READY }  
          { NOCONT } { NOHOLD } { NOTREADY } FORM form
```

```
          TO  
vaddr FOR userid DIST distcode FLASHC ccc
```

```
vaddr FLASH ovly CHAR ctab [char[char[char]]] MDYF cmod n FCB fcb
```

```
vaddr SIZE ww ll { 2WCGM } { BTS } { DATCK } { 3800-1 }  
                  { 4WCGM } { CFS } { NODATCK } { 3800-3 }
```

```
DEV { ON }  
   { OFF } PSEUDO TIMER
```

The status of all the virtual printers and punches attached to your virtual machine and the status of the pseudo timer are displayed.

Note: The next to the last line of the response (beginning vaddr SIZE) is displayed only for a virtual 3800 printer.

where:

vaddr

is the virtual device address of the virtual printer or punch.

c

is the output class assigned to spool files produced from the device.

[*]nnn

is the number of copies from 1 to 255, of each output file to be produced. When the optional asterisk (*) appears, the 3800 performs the duplication internally, provided the spool file contains sufficient information to be spooled to the 3800 printer. If there is insufficient information, the asterisk is ignored and printing occurs normally on the printer assigned in the spool file class.

Notes:

1. You can read more about spooling to a 3800 printer under "Spooling Output to a 3800 Printer" on page 223.
2. See the descriptions of the CMS SETPRT command in the VM/SP CMS Command and Macro Reference to determine which spool options may be changed by the SETPRT command.

TO userid

indicates that the output from the device, when closed, becomes a reader input spool file for the indicated *userid*.

FOR userid

indicates the *userid* identification (spool file owner) assigned to spool files produced from the device.

distcode

is the distribution code assigned to each spool file produced from the device.

ccc

is the number of copies that are printed while the forms overlay frame is in place.

ovly

is the name of the forms overlay frame superimposed on the output pages.

ctab

is the name of the character arrangement table used to generate output.

cmod

is the name of the copy modification module used to alter output text.

fcbl

is the name of the FCB module used in the vertical formatting of a page.

ww

indicates the form width code (in hexadecimal) for a virtual 3800 printer.

ll

indicates in decimal notation the length of the paper in half-inches loaded into a 3800 printer.

Note: The distcode in this case indicates the FOR userid; however, the distcode produced on the output files when the file is closed is the distcode assigned to the FOR userid as specified in the virtual machine directory entry.

The other fields indicate the setting of the respective options in the SPOOL command.

The default settings are:

```
PUN vaddr CL A NOCONT NOHOLD COPY 001 READY FORM form
vaddr FOR userid DIST distcode
```

```
PRT vaddr CL A NOCONT NOHOLD COPY 001 READY FORM form
vaddr FOR userid DIST distcode FLASHC 000
vaddr FLASH CHAR MDFY FCB
vaddr SIZE OF 11 4WCGM CFS NODATCK
```

QUERY

where:

userid and *distcode* are assigned for the virtual machine.
form is the installation defined default value.

QUERY VIRTUAL STORAGE

STORAGE = nnnnnK

The size of the virtual machine in multiples of 1024 bytes is displayed.

QUERY VIRTUAL ALL

Has the same effect as if all the following commands were issued:

```
QUERY VIRTUAL STORAGE
QUERY VIRTUAL LINES
QUERY VIRTUAL TAPE
QUERY VIRTUAL UR
QUERY VIRTUAL DASD
QUERY VIRTUAL GRAF
QUERY VIRTUAL CONSOLE
QUERY VIRTUAL CHANNELS
```

QUERY VIRTUAL vaddr -vaddr

The response is in the same form as QUERY VIRTUAL DASD, TAPES, LINES, or UR, depending on virtual device type. When a range of addresses is specified a response is used for each device within that range.

If *vaddr* corresponds to a 3330V (Mass Storage System's virtual 3330 volume) address, the following response is issued:

```
DASD vaddr ON DASD vaddr
```

If *vaddr* on the command line is a Mass Storage Control 3851 port address, the following response is issued:

```
MSC vaddr ON DEV vaddr
```

QUERY LINKS vaddr

```
userid vaddr  R/O , ...
.         .    R/W
:         :
.         .
```

A list of users who are currently linked to the device at virtual address *vaddr* is displayed.

where:

userid

is the identification of the user who originated the link.

vaddr

is the virtual address by which the user (*userid*) refers to the device.

{R/O }
{R/W }

is the type of access the user (*userid*) has to the device.

QUERY SCREEN

The response you receive is:

```
CPOUT      color  exthilight  VMOUT      color  exthilight
INREDISP   color  exthilight  INAREA     color  exthilight
STATAREA   color  exthilight
```

where:

color

is the color value currently in effect for your virtual console.

exthilight

is the extended highlight value currently in effect for your virtual console.

Note: If you do not have a SCREEN directory statement in your directory entry and you have not issued the SCREEN command during the current terminal session, the response to the QUERY SCREEN command is DEFAULT for the color value and NONE for the exthilight value.

QUERY SECUSER

The response is one of the following:

SECONDARY USER IS UNDEFINED

is the response if no secondary user appears in the user's directory entry. If the primary user disconnects, all output issued during that period is lost.

SECONDARY USER nnnnnnnn IS LOGGED ON

is the response if the secondary user is logged on. If the primary user disconnects, the secondary user receives all output issued until the primary user reconnects.

SECONDARY USER nnnnnnnn IS NOT LOGGED ON

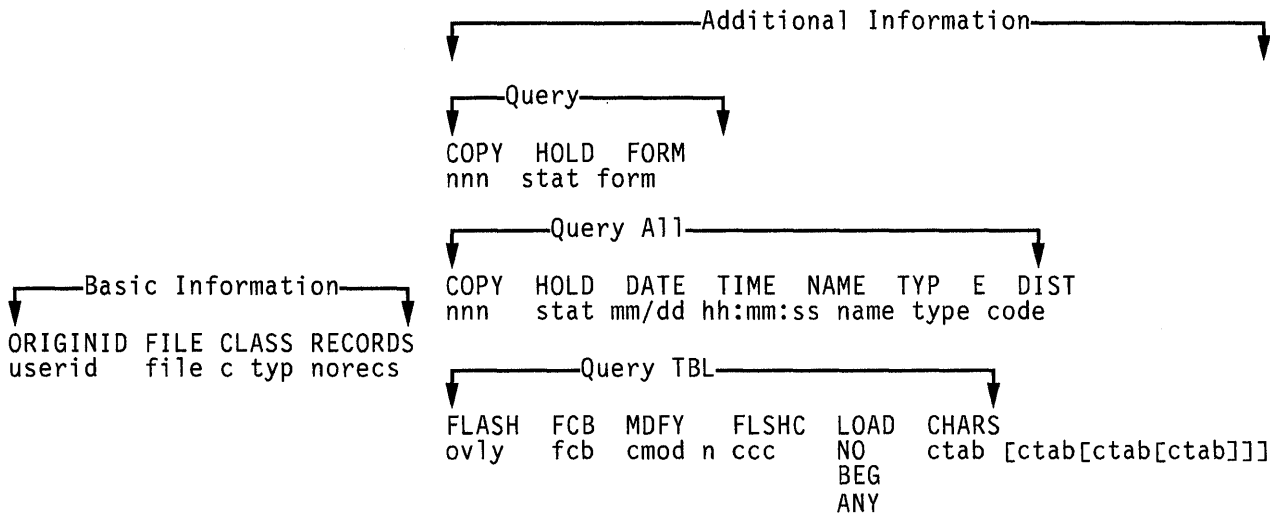
is the response if the secondary user is not logged on. If the primary user disconnects, all output issued is lost until the primary user reconnects or the secondary user logs on.

SECONDARY USER nnnnnnnn IS DISCONNECTED

is the response if the secondary user is disconnected. If the primary user disconnects, all output issued is lost until either the primary user reconnects or the secondary user reconnects.

QUERY

QUERY READER, QUERY PRINTER, QUERY PUNCH



where:

userid

is the user who originally created the file.

file

is a unique, system-assigned spool file identification number (spoolid) which is used by VM/SP HPO to identify the file.

c

is the spool file class.

typ

is the originating device type (PRT, PUN, CON, DMP, or RDR).

norecs

is the number of logical records contained in the file.

*** nnn**

is the number of copies assigned to the file (it has no effect for virtual reader files). Appearance of the asterisk indicates that spool file duplication is handled by the 3800 printer. The device copies a spool file internally one page at a time (collated output produced).

stat

is the file hold status: NONE (no hold), USER (user hold), SYS (system hold), or USYS (system and user hold).

form

is the form name of the spool file.

mm/dd

is the date the file was created in month/day.

hh:mm:ss

is the time the file was opened for creation in hours:minutes:seconds. For some files, such as spooled console files, this time may be a span of hours before the time at which the file is closed.

fn

is the filename assigned to the file (if any). If the file has a 24-character data set name (dsname), only 18 characters are displayed. These characters extend from the "name" field through the "type" field.

ft

is the filetype assigned to the file (if any).

distcode

is the distribution code assigned to the file.

ovly

is the name of the forms overlay frame superimposed on the output.

ctab

is the name of the character arrangement table used to generate output.

fcbl

is the name of the FCB module used in the vertical formatting of a page or is the number of print lines per inch (plpi) which can be 6, 8, or 12. This number may be 10 for the 3800 Model 3.

cmod

is the name of the copy modification module used to alter output text.

n

is the copy modification character arrangement table number.

ccc

is the number of copies that are printed while the forms overlay frame is in place.

NO

indicates that no 3800 LOAD channel command words (CCWs) are positioned within the 3800 spool file.

BEG

indicates that 3800 LOAD channel command words (CCWs) are positioned at the beginning of the 3800 spool file.

ANY

indicates that 3800 LOAD channel command words (CCWs) are interspersed throughout the 3800 spool file.

When you issue QUERY READER, QUERY PRINTER, or QUERY PUNCH commands, CP responds by listing (in the form described) all the closed spool files associated with your virtual reader, printer, or punch.

The information listing DATE, TIME, NAME, TYPE, and DIST (date of file creation, time of file creation, filename, filetype and file distribution code) is displayed only when you specify the ALL operand. However, if you specify the TBL operand, the information listing FLASH, CHARS, FCB, MDFY, FLSHC, and LOAD (3800 overlay form, 3800 character arrangement table, 3800 vertical formatting, 3800 copy modifications, the number of copies to be printed by 3800 with overlay form, and the placement of 3800 LOAD CCW's within a 3800 spool file) is displayed.

Once a printer file starts printing, it does not appear in the response to a QUERY PRINTER command.

If a reader file appears to be lost, due to the virtual machine abnormally terminating without closing the reader, you can issue the command CLOSE READER HOLD to try to recover the reader file.

To terminate the displaying of the response to a QUERY READER, QUERY PRINTER, or QUERY PUNCH command, press the Attention key (or its equivalent).

Note: For further information on altering spool file attributes, refer to the CHANGE command in this section.

QUERY PFnn

$$\text{PFnn} \left\{ \begin{array}{l} \{ \text{IMMED} \} \\ \{ \text{DELAY} \} \\ \text{RETRIEVE} \end{array} \left[\begin{array}{l} \text{tab...} \\ \text{copy...} \\ \text{pfdata1...} \end{array} \right] \right\}$$

The program function defined for a program function key is displayed. If there is no function defined for the program function key, the following message is issued:

PFnn UNDEFINED

Notes:

1. *A logical line end character imbedded in pfdata1 appears as a carriage return in the response to a QUERY PFnn command.*
2. *In the case of PFnn COPY or PFnn TAB, the existence of unnecessary or lack of trailing blanks may negate the function and force the PF key to be recognized as pfdata.*

QUERY VMSAVE

```

USER RECEIVER SYSNAME PRTY IPLD ENBLD IN USE DATE/TIME VOLID
USERF USERA AREA 6 0015 NO NO NONE N/A TND198
0001-0002 0006

```

where:

USER

is the *userid* of the user who can store into the area or IPL the area (the owner of the area).

RECEIVER

is the *userid* of the user who can IPL the area.

SYSNAME

is the name of the area specified on a SET VMSAVE or IPL command.

PRTY

is the priority number of this area.

IPLD = YES

indicates that either the owner or the receiver is currently loaded from the area.

ENBLD = YES

indicates that the owner has issued a SET VMSAVE for this area.

IN USE = YES

indicates that a VMSAVE area that overlaps this one is in use by some user (the area has been loaded, VMSAVE is enabled, or a system is already saved in the area).

DATE/TIME = NONE

indicates that no system is currently stored in this area.

DATE/TIME = N/A

indicates one of the following:

- The DASD volume is not mounted or is not CP owned.
- An overlapping area is enabled for VMSAVE or contains a valid system.

VOLID

is the volume serial number of the target volume.

The second and subsequent lines indicate the page range associated with this area. A page range is indicated by the first and last page numbers of the range separated by a hyphen. A single page is simply indicated by the page number. For a saved system, these pages are the ones that were stored, and the ones that are loaded by the IPL command; for a virtual machine with VMSAVE enabled, this is the

list of pages that are saved when and if the virtual machine is terminated by VM/SP HPO or when VM/SP HPO itself terminates. See the discussion on the VMSAVE option in the *VM/SP HPO System Programmer's Guide* for more information regarding the specification and use of the VMSAVE option.

QUERY S370E

S370E { ON [XMEM] } [PROC xx, { ON [XMEM] } PROC yy]
 { OFF } { OFF }

This response tells you if the System/370 extended (hardware) special feature has been installed and, if installed, whether or not the system operator has enabled the feature. When installed and enabled, an MVS/SP virtual machine may use the feature after the virtual machine operator has issued the SET 370E command. If you are running MVS/System Product Release 3 or later in your virtual machine and your system is equipped with cross memory assist, XMEM appears in the QUERY SET response.

With attached processors, the system operator may enable the feature for either or both processors. When enabled for only one processor in an AP/MP system, you must issue the SET AFFINITY ON command for that processor before the system accepts the SET 370E command. If your MVS/SP virtual machine is configured as an AP or MP, and you issue a SET 370E ON XMEM command, the XMEM response indicates the processor that is enabled for this facility.

QUERY PROCESSR

PROCESSOR mm ONLINE[, PROCESSOR nn ONLINE, SYSTEM IN { AP } MODE]
 { MP }

where:

mm

is the processor ID number of the main processor if the system is running in UP or AP modes; in MP mode, it is the processor ID number of the IPL processor.

nn

is the processor ID number of the attached processor if the system is running in AP mode; in MP mode, it is the processor ID number of the non-IPL processor.

If VM/SP HPO is running in uniprocessor mode, the bracketed portion of the response is omitted. If the system is running in AP or MP modes, the bracketed portion of the response is displayed.

QUERY SPMODE

SPMODE { ON }
 { OFF }

In a VM/SP HPO system, this response indicates whether or not VM/SP HPO is operating in single processor mode. If it is, VM/SP HPO uses only the main processor in the configuration; the V=R virtual machine has exclusive use of the other processor.

Note: When VM/SP HPO is AP or MP generated and SPMODE is off, the user can reset the affinity setting of the virtual machine after the attached processor or non-IPL processor has been varied online — VM/SP HPO has resumed AP or MP mode. You cannot set affinity for the preferred machine assist-mode guest while preferred machine assist is active. (You can set affinity for other machines.)

QUERY CPLEVEL

VM/SP HPO RELEASE *r*, SERVICE LEVEL *ssss*
GENERATED AT *mm/dd/yy hh:mm:ss* *timezone*
IPL AT *mm/dd/yy hh:mm:ss* *timezone*

where:

r
is the software release number

ssss
is the software service level number. The number indicates which service tape has been applied to the CP software; it cannot identify which individual updates have been incorporated into CP.

GENERATED AT *mm/dd/yy hh:mm:ss* *timezone*
indicates the date and time the CP system software was generated.

IPL AT *mm/dd/yy hh:mm:ss* *timezone*
indicates the date and time the CP was last started.

QUERY LOGMSG

```
* logmsg text line 1
.
.
.
* logmsg text line n
```

All lines (both those with an asterisk and without) in the log message file are displayed.

QUERY

QUERY NAMES

```
userid - { DSC } , ...  
          { raddr }  
          { Lnnn }  
          { nnnn }  
userid - { DSC } , ...  
          { raddr }  
          { Lnnn }  
          { nnnn }  
VSM      - userid , ...  
userid - luname , ...
```

These responses list all logged-on users. If a user is not connected to the system, DSC is displayed. If he is connected, the real address to which he is connected is displayed as *raddr*. If he is connected through a VM/Pass-through program product, Lnnn displays the logical device address of the user accessing the host VM/SP HPO system through the Pass-through virtual machine. If he is connected through a 3704/3705/3725 line, nnnn, the resource identification, is displayed. If the user is logged on to the system via an SNA virtual console, CP is not aware of that LU (logical unit) device. In this case, it displays the *luname* for the *userid* rather than the real address. The *userid* of the VTAM Service Machine (VSM) that supports the SNA console precedes the console listing. The VSM and *luname* displays follow all *userid* and real address listings connected to the system.

Note: The logical unit is the port through which an end user accesses the SNA network to communicate with another end user and through which the user can access the facilities of the system services control point.

If a terminal user is logged on but has not yet been validated this is shown by:

```
LOGOLnnn, if he is using a logical terminal;  
LOGONnnn, if he is using a local terminal;  
LOGNnnnn, if he is using a remote terminal.
```

QUERY USERS

```
nnn USERS, mmm DIALED, xxx NET
```

The number of users logged on and dialed to the system is displayed.

where:

nnn

is the total number of logged-on users.

mmm

is the total number of users attached using DIAL to virtual machines.

xxx

is the total number of users logged on through a VSM in a SNA network.

Note: DIALED means the line is not available to CP because it is logically attached to a multiple-access virtual machine and is a part of that user's virtual machine operation.

QUERY userid or QUERY USERS userid

```
userid - { raddr }
          { nnnn }
          { luname }
          { Lnnn }
```

The real address (*raddr*) or the resource identification of a 3704/3705/3725 line (*nnnn*) to which the specified user is connected is displayed. The *luname* is displayed instead of the *raddr* or resource identification if that user is logged on through a SNA network using the VTAM Service Machine. The logical address of the specified user, *Lnnn*, is displayed if your installation has the VM/Passthrough program product.

QUERY PER CURRENT

```
1 trace-element1
2 trace-element2
.
.
.
n trace-elementn
PER COUNT c
```

where:

n is the trace element number.

trace-elementn

is the nth trace element in the current traceset.

c is the value of an active PER count.

QUERY PER NAMES

```
saved-traceset1 saved-traceset2...saved-tracesetn
```

where:

saved-tracesetn

is the name of the nth saved traceset.

QUERY

QUERY PER traceset-name

```
saved-trace-element1
saved-trace-element2
      .
      .
      .
saved-trace-elementn
```

where:

saved-trace-elementn

is the nth trace element in the saved traceset.

QUERY PER ALL

```
1 trace-element1
2 trace-element2
      .
      .
      .
n trace-elementn
PER COUNT c
saved-traceset1
  saved-trace-element1
      .
      .
      .
  saved-trace-elementn
      .
      .
      .
saved-tracesetn
  saved-trace-element1
      .
      .
      .
  saved-trace-elementn
```

where:

n is the traceset element number.

trace-element

is an element in the current traceset.

c is the value of an active PER count.

saved-traceset

is a name of a previously saved traceset.

saved-trace-element

is an element in a saved traceset.

Note: Output produced for the QUERY PER display will have the GUESTR or GUESTV keyword even if DATOFF or DAT was specified. If both second- and third-level activity is being traced, then neither GUESTR nor GUESTV will be included in the QUERY PER output.

READY

Privilege Class: G

Use the READY command to set a device-end interruption pending for a specified virtual device.

The format of the READY command is:

READY	vaddr
-------	-------

where:

vaddr

is a virtual device address (*cuu*).

Usage Notes

1. The status of the virtual machine is unchanged.
2. Other than having a device-end interruption pending, the virtual device is unchanged.

Responses

None.

REQUEST

REQUEST

Privilege Class: G

Use the REQUEST command to make an attention interruption pending at your virtual console.

The format of the REQUEST command is:

REQuest	
---------	--

Usage Notes

1. The REQUEST command performs the same functions as ATTN and the two commands can be used interchangeably.
2. CP ignores the REQUEST command if any I/O operation is in progress at the console or if other interruptions are pending. This condition may result if the user issues the REQUEST command by means of DIAGNOSE X'8'.

Responses

None.

RESET

Privilege Class: G

Use the RESET command to clear all pending interruptions from the specified virtual device.

The format of the RESET command is:

RESET	vaddr
-------	-------

where:

vaddr

is a virtual device address (*cuu*) of the device to be reset.

Usage Notes

1. In addition to clearing all pending interruptions, all error conditions occurring as a result of unit checks and virtual sense bytes are reset.
2. When the DIAL command is issued from a local or remote Binary Synchronous 3270 terminal, the virtual system user must use the CP command RESET to drop the dialed connection.

Responses

DEVICE RESET

is the response you receive upon the the successful completion of the command.

REWIND

REWIND

Privilege Class: G

Use the REWIND command to rewind (but not unload) a real tape unit attached to your virtual machine at a specific virtual device address.

The format of the REWIND command is:

REWInd	vaddr
--------	-------

where:

vaddr

is the virtual device address (*cuu*) of the tape unit to be rewound.

Usage Note

The REWIND command accomplishes the manual operation of rewinding and making the tape ready at the tape unit.

Responses

REWIND COMPLETE

is the response you receive upon the successful initiation of the command. Receiving this message does not imply that the tape is physically rewound yet.

REWIND NOT PERFORMED

is the response you receive if the real tape unit is not ready.

SCREEN

Privilege Class: G

Use the SCREEN command to alter or change any color and/or extended highlight definitions for your virtual machine console. You may issue the command from any IBM supported terminal or from a PROFILE EXEC because the SCREEN command is not device dependent. However, the SCREEN command only has an effect on your display when the Extended Color Feature has been installed on the terminal controller.

You can assign color and extended highlighting values to six distinct display screen areas: the input area, the system status area, and the output area that encompasses three other areas: CP output, virtual machine output, and an input redisplay area. The physical attributes of 3270 Information Display station screens vary according to model. For a general discussion of screen layout and the way CP handles color and extended highlight definitions, refer to the discussion of display terminals in *VM/SP Terminal Reference*.

The format of the SCREEN command is:

SCREEn	$\left\{ \text{area} \left\{ \left[\left\{ \text{extcolor} \right\} \left\{ \text{exthilight} \right\} \right] \right\} \right\}^1$
¹ Each time you enter the command, you must specify at least one screen 'area' operand with at least one 'extcolor' and/or 'exthilight' value. You may specify more than one 'area' operand on the same command line.	

where:

area

specifies the area of the screen. Area can be:

ALL

the entire screen. If you specify this operand, you cannot specify any other area on the same command.

INAREA

the input area.

STATAREA

the system status area.

OUTAREA

the output area. The OUTAREA operand encompasses the next three areas. If you specify OUTAREA, you cannot specify CPOUT, VMOUT, or INREDISP on the same command line as OUTAREA.

CPOUT

is the output from the control program. You may not specify CPOUT on the same command as OUTAREA.

VMOUT

is the output from CMS or the virtual operating system running in your virtual machine. You may not specify VMOUT on the same command line as OUTAREA.

INREDISP

is your input redisplay. Messages are displayed with the same characteristics specified for the input redisplay area. You may not specify INREDISP on the same command line as OUTAREA.

extcolor

specifies the extended color attribute you want to assign to an 'area' of your screen. The extended color value can be:

BLUe – blue
RED – red
GREen – green
YELlow – yellow
TURquois – turquoise
PINK – pink
WHItE – white
DEFault – monochrome is the default.

exthiligh

specifies the extended highlight value you want to assign to an area of your screen. The exthiligh value can be:

BLInk – blinking
REVvideo – reverse video
UNDerlin – underlining
NONE – no extended highlighting

Usage Notes

1. If you issue the SCREEN command from a terminal that does not support extended color and/or extended highlighting, CP still sets the values you specify for your virtual machine console. Then, if you disconnect from that terminal and logon at a terminal that does support extended color and/or extended highlighting, the values you set are implemented for your virtual machine after the reconnection. The extended color and extended highlighting values remain in effect for the duration of the current terminal session unless you alter them with other SCREEN commands.
2. You cannot specify a particular 'area' operand more than once on the same SCREEN command line.

3. If you enter the SCREEN command and you fail to enter any operands (or enter invalid operands), command processing terminates, you receive an error message, and the extended color and extended highlighting values in effect for your terminal remain unchanged by the invalid command.
4. If you logon at a VCNA terminal, disconnect, and then reconnect, the extended color and extended highlighting attributes previously set will not take effect again until the system issues the first "CP READ" following the reconnect process.

Example:

If you enter:

```
SCREEN OUTAREA RED NONE INAREA GREEN BLINK STATAREA
PINK UNDERLIN
```

CP sets the following values for your terminal:

AREA		COLOR	EXTHIGHLIGHT
cpout	} outarea	red	none
vmout		red	none
inredisp		red	none
inarea		green	blinking
statarea		pink	underlining

Response

COMMAND COMPLETE.

SEND

Privilege Class: G

Use the SEND command to pass commands and message replies from the secondary user's console to disconnected virtual machines for execution. This is accomplished using the Single Console Image Facility, which is documented in the *VM/SP HPO System Programmer's Guide*.

The format of the SEND command is:

```
SEND      [#CP]  userid [ text]
```

where:

CP

indicates that CP executes the text on behalf of the receiving virtual machine. If the first three bytes of text are '#CP' or if the receiving virtual machine is already in CP READ state, the user need not specify the CP operand when sending a CP command. If the SEND command does not include text, CP puts the receiving virtual machine into console function mode. This operand is equivalent to the #CP command.

userid

represents the disconnected virtual machine that has a specified secondary user (*secuser*) in its directory entry.

text

can be any command allowed in the virtual machine command environment or a reply to a message issued by the disconnected virtual machine. CP passes the text to the disconnected virtual machine for processing. If the user wants to send multiple lines in the text, he must use the logical line end (#) immediately preceded by the logical escape character (""); in addition, the user must have line editing enabled in his virtual machine. Note that this restriction applies when the CP command '#CP' is part of the SEND command text. The logical line edit character (#) is, in fact, a carriage return, implying multiple lines.

Usage Notes

1. To use the SEND command, you must qualify as the secondary user. This happens when your *userid* is including in the CONSOLE control statement in the directory entry of the receiving virtual machine. For more about this control statement, see the *VM/SP HPO Planning Guide and Reference*.
2. The receiver of the SEND command input must be running disconnected.

3. When a logical line edit character is sent in the **SEND** command text, the character must be the one defined for the sender, not the disconnected machine.

Responses

None.

SET

SET

Privilege Class: G

Use the SET command to control various functions within your virtual system.

The format of the SET command is:

SET		
	ACNT	
	AUTOPoll	
	RUN	
	LINEDit	{ ON }
	ECmode	{ OFF }
	ISAM	
	NOTrans	
	PAGEX	
	MIH	
	EMSG	{ ON }
		{ OFF }
		{ CODE }
		{ TEXT }
		{ IUCV }
	{ MSG }	{ ON }
	{ SMsg }	{ OFF }
	{ WNG }	{ IUCV }
	{ IMSG }	
	CPConio	{ OFF }
	VMConio	{ IUCV }
	TIMER	{ ON }
		{ OFF }
		{ REAL }
	AFFinity	{ ON }
		{ OFF }
	CPUid	bbbbbb

(Part 1 of 2)

ASsist	{ ON [SVC] [TMR] [NOSVC] [NOTMR] }
	{ OFF }
PFnn	{ IMMed [pfdata1#pfdata2#...#.pfdata _n] DELayered RETRieve [TAB n1 n2 ...] [COPY [nnnn [COPY [raddr [COPY [luname
VMSave	{ ON OFF name }
STBypass	{ [nnnnnK NOVERIFY] [nnM] VR OFF }
STMulti	{ [nn [USEG xx] CSEG yyy] ON OFF }
370E	{ ON[XMEM] OFF }

(Part 2 of 2)

where:

ACNT {ON }
{OFF }

controls whether accounting information is displayed at the terminal or not (ON and OFF, respectively) when the operator issues the CP ACNT command. When you logon, ACNT is set on.

AUTOPOLL {ON }
{OFF }

controls whether or not CP tests each BTAM autopoll CCW to see if it has been dynamically modified. ON indicates that (1) CP bypasses testing, and (2) the VS1 operating system is to notify CP, using the DIAGNOSE instruction code X'0028', after an autopoll CCW has been modified. ON substantially reduces the overhead required by CP to service BTAM autopoll channel programs. OFF causes CP to check each autopoll CCW after each execution of the channel program and is the default status when the user logs on.

Note: If you SET AUTOPOLL ON and the virtual operating system does not have the appropriate Diagnose interface, CP does not detect changes to BTAM channel programs and unpredictable results can occur. OS/VS 1 Release 6.0, however, does have this interface.

MSG { ON
 OFF
 IUCV }

controls whether messages sent by the MSG command from other users are to be received at the terminal. If ON is specified, the messages are displayed. If OFF is specified, no messages are received. In addition to controlling messages generated by the MESSAGE command, spooling messages generated by users sending punch, printer or reader files to another virtual machine are also suppressed if OFF is specified. When you logon, MSG is set on. If IUCV is specified, the message is passed to the virtual machine via IUCV if a connection to the Message System Service exists. If no IUCV connection exists, the message will be handled as if SET MSG ON had been issued.

SMSG { ON
 OFF
 IUCV }

controls whether a virtual machine is receiving special messages or not. If OFF is specified, the virtual machine cannot receive special messages. If IUCV is specified, the virtual machine will receive special messages via IUCV if a connection to the Message System Service exists.

WNG { ON
 OFF
 IUCV }

controls whether warning messages are displayed at the terminal. If ON is specified, all warning messages sent using the CP WARNING command from the system operator are received at the terminal. If OFF is specified, no warning messages are received. When you logon, WNG is set on. If IUCV is specified, the message is passed to the virtual machine via IUCV if a connection to the Message System Service exists. If no IUCV connection exists, the message will be handled as if SET WNG ON had been issued.

IMSG { ON
 OFF
 IUCV }

controls whether certain informational responses issued by the CP CHANGE, DEFINE, DETACH, IPL, ORDER, PURGE, and TRANSFER commands are displayed at the terminal or not. The descriptions of these CP commands tell which responses are affected. If ON is specified the informational responses are displayed. If OFF is specified, they are not. The SET IMSG ON or OFF command line has no effect on the handling of error messages set by the SET EMSG command. When you logon, IMSG is set on. If IUCV is specified, the

message is passed to the virtual machine via IUCV if a connection to the Message System Service exists. If no IUCV connection exists, the message will be handled as if SET IMSG ON had been issued.

RUN {ON }
{OFF }

controls whether the virtual machine stops when the attention key is pressed. ON allows you to activate the attention or PA1 key (causing a read of a CP command) without stopping your virtual machine. If you use the ON operand, the command output may be interrupted by a guest virtual machine read or write. When you enter the CP command, it is immediately executed and the virtual machine resumes execution. OFF places the virtual machine in the normal CP environment, so that when you press the attention key, the virtual machine stops. When you logon, RUN is set off.

If you plan to issue an ADSTOP command, you must first issue a SET RUN OFF command. This command allows the virtual machine to halt execution at the address specified on the ADSTOP command line.

LINEDIT {ON }
{OFF }

controls the line editing functions. ON specifies that the line editing functions and the symbols of the system are to be used to edit virtual processor console input requests. This establishes line editing features in systems that do not normally provide them. OFF specifies that no character or line editing is to be used for the virtual machine operating system. When you logon, LINEDIT is set on. The LINEDIT operand is unaffected by the setting of the APL/TEXT operand of the TERMINAL command.

ECMODE{ON }
{OFF }

controls whether the virtual machine operating system may use extended control mode and control registers 1 through 15. Control register 0 may be used with ECMODE either ON or OFF. When you logon, ECMODE is set according to the user's directory option; ON if ECMODE was specified and OFF if not.

Notes:

1. *Whenever you enter the SET ECMODE ON/OFF command, the virtual machine is reset.*
2. *The SET ECMODE command automatically resets (turns off) the preferred machine assist, whether or not you have enabled preferred machine assist's control switch assist.*

Setting ECMODE does not change the PSW EC mode bit.

**ISAM {ON }
{OFF }**

controls whether additional checking is performed on virtual I/O requests to DASD in order to support the OS Indexed Sequential Access Method (ISAM). When you logon, ISAM is set according to the OPTION control statement in the user's directory entry; ON if ISAM was specified and OFF if not.

**NOTRANS {ON }
{OFF }**

controls CCW translation for CP. NOTRANS can be specified only by a virtual machine that occupies the virtual=real space. It causes all virtual I/O from the issuing virtual machine to bypass the CP CCW translation except under the following conditions:

- SIO tracing active
- First CCW not in the V=R region
- I/O operation is a sense command
- I/O device is a dial-up terminal
- I/O is for a non-dedicated device
- Pending device status

Any of the above conditions forces CCW translation.

To be in effect in the virtual=real environment, SET NOTRANS ON must be issued after the virtual=real machine is loaded using the IPL command. (IPL sets the NOTRANS option to an OFF condition. The SET NOTRANS ON command cannot be issued until after the IPL of the virtual=real machine is completed.)

**PAGEX {ON }
{OFF }**

controls the pseudo page fault portion of the VM/VS handshaking feature. PAGEX ON or OFF should only be issued for a virtual machine that has the VM/VS handshaking feature active. It can only be specified for a virtual machine that has the extended control mode (ECMODE) option. PAGEX ON sets on the pseudo page fault portion of handshaking; PAGEX OFF sets it off. When you logon, PAGEX is set OFF. Also, each time you IPL your virtual machine, PAGEX is set off. If you want to use the pseudo page fault handling portion of handshaking, you must issue SET PAGEX ON after you load your operating system.

**MIH {ON }
{OFF }**

controls whether the Missing Interrupt Handler is set on or off, provided that DMKDID is installed on your system. ON specifies that a missing interrupt will be simulated by CP. OFF specifies that a missing interrupt will not be simulated by CP, but that a message will be written to the operator when a missing interrupt is detected.

Note: When MIH is set on, you will receive an interface control check (IFCC) when an interrupt cannot be simulated.

EMSG { ON
OFF
CODE
TEXT
IUCV }

controls error message handling. ON specifies that both the error code and text are displayed at the terminal. TEXT specifies that only text is displayed. CODE specifies that only the error code is to be displayed. OFF specifies that no error message is to be displayed. When you logon, EMSG is set to TEXT. IUCV specifies that both the error code and message text are to be passed to the virtual machine via IUCV if a connection to the Message System Service exists. If no IUCV connection exists, the data will be handled as if SET EMSG ON had been issued.

If the console is being spooled, then for CP error messages only, the OFF setting is ignored for the spooled output. The spooled output contains the full error message. If you use any of the other four settings, you will receive spooled output that matches the console printout.

It is the responsibility of the operating system running under CP to determine whether its own error messages will honor the EMSG setting in the spooled output.

Note: CMS recognizes EMSG settings for all error (E), information (I), and warning (W) messages. However, CMS ignores the EMSG setting and displays the complete message, including error code and text, for all response (R), severe error (S), and terminal (T) messages. Also, data written by means of a type in, such as a list of names following a message, are not controlled by the EMSG operand.

CPCONIO { IUCV }
{ OFF }

controls the destination of messages and responses generated by CP. If IUCV is specified, CP generated messages and responses are sent to the virtual machine via IUCV if a connection to Message System Service exists. If OFF is specified, messages are displayed at the terminal. If no IUCV connection exists, data will be handled as if SET CPCONIO OFF had been issued.

VMCONIO { IUCV }
{ OFF }

controls the destination of virtual machine generated responses. This includes any CMS error messages and any linemode I/O done by the virtual machine to the virtual console (e.g. WRTERM, LINEDIT, DMSERR). If IUCV is specified, responses are sent to the virtual machine via IUCV if a connection to the Message System Service exists. If OFF is specified, messages are displayed at the terminal. If no IUCV connection exists, data will be handled as if SET VMCONIO OFF had been issued.

TIMER { **ON**
 OFF
 REAL }

controls the virtual timer. ON specifies that the virtual timer is to be updated only when the virtual processor is running. OFF specifies that the virtual timer is not to be updated. REAL specifies that the virtual timer is to be updated during virtual processor run time and also during virtual wait time. If the REALTIMER option is specified in your directory entry, TIMER is set to REAL when you logon; otherwise it is set to ON when you logon.

Note: You cannot issue the SET TIMER command when preferred machine assist support is active. This restriction holds whether or not you enable preferred machine assist's control switch assist.

SET TIMER REAL must be specified if an interrupt-driven system such as Customer Information Control System (CICS) is to be run in a virtual machine.

AFFINITY { **ON** }
 OFF }

controls whether or not CP dispatches a virtual machine and executes its programs on either the main processor, the attached processor of an attached processor system, or the nonIPL processor of a multiprocessor system. Normal mode is AFFINITY OFF, allowing virtual machine execution on either processor. If you enter AFFINITY ON, CP goes to the directory to find an affinity setting; if there is none in the directory, the user receives an informational message. In uniprocessor mode, affinity is meaningless because all virtual machines execute on the main processor.

When affinity is in effect for a virtual machine, only the specified processor runs the virtual machine. This is particularly useful for virtual machines that require features or performance enhancements that are only available on one processor in an attached processor or multiprocessor system. CP instructions (such as privileged operation instructions) executed on behalf of a virtual machine are not restricted to a processor by an affinity setting. If you are the V=R user and you logon while the system is in AP or MP mode, you must set affinity to one processor before you can issue the IPL command, regardless of whether you enter the PMA or PMAV options. This affinity cannot be changed after you have IPLed.

You cannot issue the SET AFFINITY command when preferred machine assist is active (either with or without the control switch assist) since you must set your affinity before IPL.

CPUID bbbbbb

CPUID is the keyword that provides a means of setting or changing the processor identification for a virtual machine. The variable value bbbbbb represents a string of six hexadecimal digits that permit unique identification of the processor. This command allows the user to provide a unique processor identification to be stored in response to the 'STIDP' instruction. It is necessary to associate a unique CPUID with each virtual machine that is attached to an MSC port since solicited/unsolicited messages are directed to the host system by means of the CPUID. There is no checking to ensure that all virtual machines using the SET CPUID command have specified unique processor serials. The hexadecimal field 'bbbbbb' is the processor identification number (processor serial).

Note: The processor identification number (serial) is only a portion of the complete CPUID. The CPUID identification stored in response to a STIDP instruction is a string of 16 hexadecimal digits as follows:

aabbbbbbcccd

where:

aa

- version code; these two digits are forced to X'FF' to identify that the virtual machine is running under VM/SP HPO.

bbbbbb

- processor identification number; this field contains six hexadecimal digits. This is the only part of the CPUID that can be set by the OPTION control statement and modified by means of the SET CPUID command.

cccc

- model number; this field contains a high order 0 digit followed by the three digits of the model number (0-9). This field defaults to the model number of the real machine.

dddd

- machine check extended logout; this field is forced to X'0000' since CP does not reflect machine checks back to the virtual machine.

ASSIST { [ON] [SVC] [TMR] }
 { OFF [NOSVC] [NOTMR] }

All functions of SET ASSIST operands pertain to VM/SP HPO hardware assist. VM/SP HPO hardware assist consists of processor hardware that assists the processor in the execution of certain instructions, series of instructions, privileged operations, and interruption processing that are normally handled by the control program portion of VM/SP HPO. In attached processor or multiprocessor systems, VM/SP HPO hardware assist is available for installation on either or both processors.

All, some, or none of the function performed by VM/SP HPO hardware assist may be available on the processor, attached processor, or nonIPL processor that the virtual machine is logged on. (For the extent of VM/SP HPO hardware assist available on supported VM/SP HPO processors, refer to the *VM/SP HPO Operator's Guide*.) The extent of VM/SP HPO hardware assist available to your virtual machine can be determined by invoking QUERY SET and by the responses to attempts to invoke the various operands of SET ASSIST. The hardware assist functions can only be activated for an individual virtual machine provided the system operator permits the associated hardware assist function to be active for the entire system (or in attached processor or multiprocessor systems, the appropriate processor). Overall system control of hardware assist is controlled by SET SASSIST and SET CPASSIST commands; see the *VM/SP HPO Operator's Guide* for details on these commands.

If VM/SP HPO hardware assist is active for the entire system, then hardware assist is active and used during virtual machine operations immediately after logon. Hardware assist as it pertains to a virtual machine can then be turned off and on again by SET ASSIST OFF and SET ASSIST ON respectively.

The SVC handling portion of the hardware assist is invoked when you logon unless your directory entry has the SVCOFF option. Issue the QUERY SET command line to see if the hardware assist is activated and whether the hardware assist or VM/SP HPO is handling SVC interruptions. All SVC 76 requests are passed to CP for handling, regardless of the SVC and NOSVC operands. If you issue the SET ASSIST command line and specify SVC or NOSVC while the hardware assist is turned off, the appropriate bits are set. Later, if hardware assist is turned on again, the operand you specified while it was off becomes effective. ON sets the hardware assist on for the virtual machine; OFF turns it off. SVC specifies that the hardware assist handles all SVC interruptions except SVC 76 for the virtual machine; NOSVC means VM/SP HPO handles all the SVC interruptions.

If you request VM/SP HPO hardware assist and it is available and active, you will get it. If VM/SP HPO hardware assist is not active but is available and you attempt to enable hardware assist, the virtual machine remembers the enabling attempt and you will receive an appropriate message. When the system operator enables hardware assist, the virtual machine utilizes it immediately. If that hardware assist is not available or active on the system, you will receive an error message. In an attached processor or multiprocessor system, hardware assist is available to any virtual machine provided it is installed on one of the processors; then, to use hardware assist, the virtual machine's affinity must be set to match the processor that contains hardware assist.

In the case of systems with attached processors or multiprocessors, the VM/SP HPO hardware assists may be installed on none, one, or both processors. Similarly, hardware assist, if equipped on both processors of an attached processor or multiprocessor system, can be activated for none, either, or both systems by the system operator. Thus, when

you have set `AFFINITY` of a virtual machine, you will not receive a response indicating that hardware assist is installed and active on his processor. In attached processor or multiprocessor setups where you have no processor preference (`AFFINITY = NONE`), you will receive no response to the command `SET ASSIST` if either processor has VM/SP HPO hardware assist active. Any error message you receive indicates that hardware assist is not available or active for user. If the user receives a message indicating that hardware assist is not available on a particular processor, it is also implying that it is available on the other processor. Therefore, the user, to take advantage of VM/SP HPO hardware, need only change his affinity setting, then initiate the `SET ASSIST` command.

The `TMR` and `NOTMR` operands control the method of updating the virtual interval timer (virtual address `X'50'`). If `TMR` is specified, the virtual interval timer assist hardware updates the virtual interval timer in the same way that the real interval timer is updated, that is, 300 times per second. If `NOTMR` is specified, the virtual interval timer is updated only when CP is given control. This can be as seldom as once per time slice interval.

See the *VM/SP HPO System Programmer's Guide* for additional information on how to use VM/SP HPO hardware assist.

Note: You cannot issue the SET ASSIST ON command when preferred machine assist is active. This restriction holds for preferred machine assist's control switch assist as well.

PFnn [**IMMED**] [**DELAYED**] [`pfdata1#pfdata2#...#pfdatan`]

defines a program function for a program function key on a 3270 Information Display station and indicates when that function is to be executed. If `PFnn` is specified with no operands, that program function key is considered *undefined*. See the *VM/SP Terminal Reference* for a description of how to use the 3270 program function keys.

In `PFnn`, *nn* is a number from 1 (or 01) to 24 that corresponds to a key on a 3270 display station. The program function is a programming capability you create by defining a series of VM/SP HPO commands or data you want executed. This series of commands is executed when you press the appropriate program function key.

`IMMED` specifies that the program function is executed immediately after you press the program function key.

`DELAYED` specifies that execution of the program function is delayed for a display terminal. When the program function is entered, it is displayed in the input area and not executed until you press the Enter key. `DELAYED` is the default value for display terminals.

Both SET PFnn TAB and SET PFnn COPY are executed immediately upon pressing the appropriate program function key. If you insert the keywords DELAYED or IMMED after the PFnn operand, the command is accepted, but the keywords are ignored. The program function still executes immediately.

pfdata1#pfdata2#...#.pfdatan

defines the VM/SP HPO command or data lines that constitute the program function. If more than one command line is to be entered, the pound sign (#) must separate the lines. If you use the pound sign (#) to separate commands that you want executed with the designated PF key, you must precede the command line with #CP, turn line editing off, or precede each pound sign with the logical escape character (").

If you use the SET PFnn command to set up a series of concatenated commands, you should be careful of line-editing characters. The following examples demonstrate two methods for entering a series of commands:

Example 1

Enter the following command while in CMS mode:

```
set linedit on
```

and press the enter key.

Now enter one of the following commands while in CMS mode:

```
#cp set pf02 immed q rdr#q prt#q pun
```

or

```
cp set pf02 immed q rdr"#q prt"#q pun
```

or

```
set pf02 immed q rdr"#q prt"#q pun
```

and press the enter key.

CP assigns the three QUERY commands as functions of the PF02 key. Pressing the PF02 key executes the three QUERY commands.

Example 2

Enter the following command while in CMS mode:

```
set linedit off
```

and press the enter key.

Then enter:

```
set pf02 immed q rdr#q prt#q pun
```

or

```
cp set pf02 immed q rdr#q prt#q pun
```

and press the enter key.

CP assigns the three QUERY commands as functions of the PF02 key.

Then enter:

```
set linedit on
```

and press the enter key.

Pressing the PF02 key executes the three QUERY commands.

If a terminal is in read-inhibited (non-display) mode and you use either a delayed PF key or an undefined PF key, the input area will be rewritten without the inhibited attribute byte, therefore displaying any data typed in at that point. The clear key can be used following the PF key to restore the inhibited read. If no command lines are entered following the PFnn designation, then the program function is marked undefined. Program functions cannot be embedded within one another.

PFnn [TAB n1 n2 ...]

specifies a program function key number to be associated with tab settings on a terminal. The number of the PF key, *nn*, can be a value from 1 (or 01) to 24. If the TAB keyword and tab settings are not entered, that program function key is considered "undefined." For examples of how this feature is used, see the *VM/SP CMS User's Guide*. TAB is a keyword identifying the tab function. The tab settings (n1 n2 ...) may be entered in any sequence.

Note: When defining the TAB function, the existence of blanks preceding the keyword TAB or the lack of trailing tab settings forces the PF key to be treated as pfdata.

SET PFnn TAB is executed immediately upon pressing the appropriate program function key. If you insert the keywords DELAYED or IMMED after the PFnn operand, the command is accepted, but the keywords are ignored. The program function still executes immediately.

PFnn [COPY [Rnnnn]]

specifies that the program function key numbered *nn* performs a COPY function for a remote 3270 Information Display terminal. The number, *nn*, must be a value from 1 (or 01) to 24. The COPY function produces a printed output of the entire screen display at the time the PF key is pressed. The output is printed on an IBM printer connected to the same control unit as your display terminal; if no printer is connected, the copy function is not available. The PF key COPY function is not available for dedicated 3284, 3286, 3287, 3288, and 3289 printers.

SET PFnn COPY is executed immediately upon pressing the appropriate program function key. If you insert the keywords DELAYED or IMMED after the PFnn operand, the command is accepted, but the keywords are ignored. The program function still executes immediately.

The resource identification, *nnnn*:

- is a four-digit hexadecimal resource identification number assigned to a specific printer;
- directs the printed copy to a specific printer; if not, the copy is printed on the printer with the lowest resource identification number;
- may be specified if more than one printer is connected to the same control unit as your display terminal (Get the numbers of these from your system programmer.);
- need not be specified if only one printer is available, as with the 3275 Display Station.

To specify PF key 12 to perform a COPY function for a remote 3270 Information Display terminal that had resource identification number A912, you would write this command:

```
SET PF12 COPY RA912
```

PFnn [COPY [raddr]]

specifies that the designated program function key, PFnn, performs a COPY function for a local 3270 Information Display terminal. The number of the PF key, *nn*, must be a value from 1 (or 01) to 24. When the PF key is pressed, the COPY function produces a printed output of the entire local screen display except for the status field, which is replaced with blanks. The *raddr* in the form *cuu* represents the real hardware address of the display printer, and may specify a printer that is on a different control unit than the one to which your 3270 is attached. If you do not specify *cuu*, the printer with the lowest *cuu* that is available on the same control unit as your 3270 is selected.

PFnn [COPY [luname]]

specifies that the program function key numbered *nn* performs a COPY function for a terminal controlled by VM/VTAM. The *nn* portion of this operand must be a value between 1 (or 01) to 24. The COPY function produces a printed output of the entire screen display at the time the PF key is pressed. The function can be used for any printer managed by VM/VTAM (that is, the function is not restricted to printers connected to the same controller). It also works on terminals controlled by VM/VTAM. CP does not know the real address (*raddr*) of the logical unit (LU) device, so you must specify a one- to eight-character *luname* (logical unit name).

Notes:

For both remote and local COPY functions:

1. *You receive a NOT ACCEPTED message, displayed in the screen status field of your 3270, if any of the following situations occur:*
 - *The printer is already busy, or all printers are busy*
 - *The printer is turned off*
 - *The printer is operational but has not been varied online by the system operator*
 - *The printer is out of paper or is in any other intervention-required condition*
 - *The designated printer does not support the copy functions. See VM/SP HPO Planning Guide and Reference for a list of 3270 Information Display printers that support the COPY function*
 - *The SET PFnn COPY command format is invalid.*
2. *You may include your own identification on the printed output by entering the data into the user input area of the screen before you press the PF key. The identification appears on the last two lines of the printed copy.*
3. *To insure the definition of a PF key for the COPY function, you can either start with one leading blank before the keyword COPY or follow the keyword with at least five blanks.*
4. *The existence of more than one data field (cuu or resid) following the keyword COPY identifies the string as pfdata rather than a hardcopy request.*

PFnn RETRIEVE

specifies that the program function key numbered *nn* performs the RETRIEVE function on a 3101 or 3270 display terminal. The number *nn* must be a value from 1 (or 01) to 24.

Once you have defined a RETRIEVE program function key, CP starts keeping your input lines. When you press the designated PF key, VM/SP HPO redisplay the most current input line from the input area, so that you can modify and re-enter the data.

VM/SP HPO can actually keep track of several input lines: The number of lines kept depends on the length of the lines; VM/SP HPO keeps more short lines than long lines, but it can always keep at least one full input line. Duplicate input lines (lines that are the same as the previous input) are not kept since it is not useful to retrieve the same line twice. For security reasons, input lines that are not displayed on the terminal, such as passwords, are not kept.

When a designated RETRIEVE program function key is first pressed, VM/SP HPO redisplay the most current input line. If the key is pressed again, VM/SP HPO displays the previous input line. Each time the key is pressed, VM/SP HPO steps through the input lines, displaying them one at a time. When VM/SP HPO reaches the oldest line that it has kept, it returns to the most current line once again.

Input lines are only kept when you have a program function key defined for the RETRIEVE function, and are logged on at a 3270 or at a 3101 using non-character mode. Input lines are not kept through a disconnect or when you use another type of terminal.

VMSAVE { **ON**
 { **OFF**
 { **name** }

controls the status of the VMSAVE option. ON enables the option for the user with exactly one DASD area. If SET VMSAVE is on, a user's pages in a 3880 Storage Subsystem device or cache will be saved when the system goes down. ON is the default when the VMSAVE option is specified in that user's directory entry. OFF disables the option for the user. The name operand allows a user with one or more DASD areas to designate which DASD area is to be enabled. When the name operand is used, the name must be identical to the name used in the SYSNAME parameter of the NAMESYS macro in the DMKSNT module. The DEFINE STORAGE command turns VMSAVE off. See the *VM/SP HPO System Programmer's Guide* for more information. Note, when you specify the VMSAVE option, the virtual machine size cannot exceed eight megabytes.

STBypass { [**nnnnnK** [**NOVERIFY**]]
 { **nnM**
 { **VR**
 { **OFF** }

allows V=R users to eliminate and reestablish shadow tables. Allows V=V users to reduce shadow-table processing.

nnnnnK

defines in thousands of bytes the highest contiguous storage address starting from zero where the virtual and real addresses are equal for a virtual machine running in a V=V area. The value is rounded down to the nearest full page boundary.

nnM

defines in millions of bytes the highest contiguous storage address starting from zero where the virtual and real addresses are equal for a virtual machine running in a V=V area.

NOVERIFY

prevents the scanning of a virtual machine's page and segment tables.

VR

eliminates shadow tables for a virtual machine running in the V=R area.

VR is an invalid option on the SET STBYPASS command if you are the MVS virtual machine working in single processor mode. However, the nnnk operand is valid under these conditions.

Note: You cannot issue the SET STBYPASS VR command when preferred machine assist is active. This restriction holds whether or not preferred machine assist's control switch assist is enabled.

See usage note 2 on page 211.

You cannot issue the SET STBYPASS VR command while working in SPMODE if you have loaded a prefix other than zero.

If you issue the SET STBYPASS VR command after issuing SET 370E ON XMEM and while using MVS cross memory services, you enable the MVS page fault assist feature.

OFF

terminates the shadow table bypass function.

$$\text{STMULTI} \left\{ \begin{array}{l} \text{nn} \left[\left[\text{USEG xx} \right] \text{CSEG yyy} \right] \\ \text{ON} \\ \text{OFF} \end{array} \right\}$$

allows users running in the system V=V or V=R who have set ECMODE on to have VM/SP HPO maintain multiple shadow tables when a virtual operating system, such as MVS, SVS and VM/SP HPO itself, uses multiple segment tables.

nn

defines the number of shadow tables between 1 and the maximum of 16. If you are working with cross memory enabled, you need a minimum of two shadow tables. CP will meet this requirement for you by building two shadow tables even if you specify SET STMULTI 1 or SET STMULTI OFF. If you specify a value greater than one, CP uses that value. If you issue a SET STMULTI ON command instead of specifying a value, the system uses the normal default, three. To reduce the overhead involved in building STOBLOK chains, you should specify the maximum STMULTI value, 16.

ON

establishes the system default of 3 for the number of shadow tables.

USEG xx

defines the size of the contiguous preallocated pool of shadow page tables. *xx* is the number of shadow page tables in each pool (one pool per set of shadow tables) and ranges from 8 through 99, or can be 0. The USEG operand is only valid when STMULTI *n* or ON was previously specified.

CSEG yyy

defines the number of full segments at the high end of the address space that are shared by all address spaces. The value of *yyy* ranges between 0 and 128 and represents the number of shadow page tables to be allocated for CSEG.

OFF

sets the number of shadow tables to 1.

Note: You cannot issue the SET STMULTI command when preferred machine assist is active. This restriction holds whether or not preferred machine assist's control switch assist is enabled. See usage notes 2 on page 211.

**370E {ON [XMEM]
 {OFF**

provided the System/370 Extended (hardware) special feature has been installed and the VM/SP HPO system operator has enabled it, this operand allows you to execute the MVS/SP, MVS/SE or any other operating systems capable of using the 370E feature.

To determine whether the product has been installed and enabled, issue the QUERY S370E command. To determine the status of the feature for your virtual machine, issue the QUERY SET command.

With attached processors, the system operator may enable the feature for either or both processors. When enabled for both processors, the SET 370E command operates as previously explained. When only one processor is enabled, you must SET AFFINITY ON for that processor before VM/SP HPO accepts the SET 370E command.

If you specify XMEM, you improve the performance of MVS/SP cross memory services (Release 3 or a later release). Cross memory services require the same installation and/or affinity setting precautions as System/370 Extended (above). See usage note 3 on page 211. If you issue a SET 370E OFF command while cross memory assist is enabled, you turn off the XMEM feature along with the System/370 Extended feature.

Note: You do not need to issue the SET 370E command running with preferred machine assist. Because the preferred machine assist handles privileged operations for the preferred guest, the MVS/SP cross memory features operate as if they were running a real machine. This is true whether or not you are running with the preferred machine assist's control switch assist enabled.

Usage Notes

1. System default values for SET operands or the values set by previously invoked SET command operands can be determined by the use of the QUERY command.
2. You cannot issue the SET STMULTI and SET STBYPASS VR commands with preferred machine assist, enabled, since these preferred machine assist do not use shadow tables. This is true whether or not the preferred machine assist are enhanced with control switch assist.
3. If MVS/SP Cross Memory Services are available on only one processor in an AP or MP configuration, you must first issue the SET AFFINITY ON command for the processor equipped with the 3033E feature. Then you may issue the SET 370E ON XMEM command to work with cross memory. Issuing a SET 370E ON XMEM command without setting affinity does not enable the cross memory assist for the virtual machine.

Responses

None.

SLEEP

Privilege Class: Any

Use the SLEEP command to place the virtual machine in a dormant state but allow messages to be displayed. You can specify a sleep interval in the command line and the virtual machine is awakened automatically when the specified interval has elapsed.

The format of the SLEEP command is:

```
Sleep [nn [SEC  
MIN  
HRs]]
```

where:

$nn \begin{bmatrix} \text{SEC} \\ \text{MIN} \\ \text{HRs} \end{bmatrix}$

indicates the number of seconds, minutes, or hours, measured by the time-of-day clock, that the virtual machine is to remain dormant. The value *nn* can be any decimal number from 00 through 99. If you do not specify a time unit, the value of *nn* defaults to minutes.

Usage Notes

1. During the dormant period, the virtual machine does not run but connection time is still being counted.
2. For systems using an interval timer, the terminal can be awakened at any time by signaling attention.
3. If no interval is specified, the virtual machine remains dormant until awakened by signaling attention.
4. If you issue the SLEEP command from a CP read or from a VM read using the CP “escape” function (#CP SLEEP), the end of the time interval or signaling attention returns you to the CP environment.
5. If you issue the SLEEP command while in virtual machine mode (for example, CMS execution of the command line CP SLEEP), the end of the time interval or signaling attention returns your terminal to virtual machine mode without entering the CP environment.
6. The SLEEP command, with the time interval, is a convenient way to delay or schedule the execution of certain jobs that could be run more efficiently at a later time; for example, second shift.
7. The SLEEP command will not work if the LOGON or DISCONNECT command is chained to it with #CP.

- 8. The terminal can be awakened at any time by signaling attention.
- 9. If you issue the SLEEP command while in a FULL SCREEN application, it will result in a hung terminal condition. Any write to the terminal will unlock the keyboard.

Responses

None.

SMSG

Privilege Class: G

Use the SMSG command to send a special message to a virtual machine programmed to accept and process the message.

The format of the SMSG command is:

Smsg	{userid} msgtext
------	------------------

where:

userid

is the receiving virtual machine's logon ID; if you specify *userid* as *, it indicates the message is to be sent to yourself.

msgtext

is the text of the message that is to be transmitted. Because the input buffers on various console devices differ, the length of the message is limited by the number of characters remaining on the input line after entering the command and the *userid*. For example: A 3270 display device contains a 139 character input buffer. If you enter:

$\underbrace{\text{msg pd58gh1c}}_{14 \text{ characters used}}$	$\underbrace{\text{special message}}_{125 \text{ characters left}}$
$\underbrace{\text{sm u}}_{5 \text{ characters used}}$	$\underbrace{\text{special message}}_{134 \text{ characters left}}$

Note: Although the 3270 display station contains 139 characters in its input area, CP only reads the first 134 characters. When the SMSG command is issued by DIAGNOSE X'8', CP reads only the first 132 characters.

Usage Notes

1. Special messages do not appear on the specially programmed virtual machine's screen because the messages are designed to be instructions that are passed as data parameters.
2. If SET SMSG ON is in effect, the special messages are stored in the prespecified VMCF data receiving area as they are received, and the instructions are processed by the programmed virtual machine. If SET SMSG IUCV is in effect, the message is sent to the receiving virtual machine via IUCV instead of VMCF.

3. Virtual machines can only receive and process special messages under these conditions:

- The virtual machine has issued the SET SMSG ON command, and the user wishing to receive special messages has issued an AUTHORIZE with the Virtual Machine Communication Facility (VMCF) of VM/SP HPO, or,
- The virtual machine has issued SET SMSG IUCV command, and the user wishing to receive special messages has issued an IUCV CONNECT to the CP Message System Service of VM/SP HPO.

Note: For more information on VMCF and IUCV, refer to the VM/SP HPO System Programmer's Guide. For more information on the directory options of IUCV, refer to the VM/SP HPO Planning Guide and Reference.

Responses

None.

SPOOL

Privilege Class: G

Use the SPOOL command to modify the spooling control options in effect for a given virtual spooling device or for a group of devices. The SPOOL command can also start or stop the spooling of virtual console input and output. You can direct a file to a remote location by using the SPOOL command in conjunction with the TAG command. This latter function is discussed in “Transmitting Spool Files Locally” on page 224. You will find the default values for the SPOOL command under “Default Values for Spool Files” on page 222.

The format of the SPOOL command is:

SPOOL	<pre> {Reader} {vaddr} </pre>	<pre> [Class c CONT HOLD EOF] ¹ [* NOCont NOHold NOEof] </pre>
	<pre> (Printer PRT) (Punch PCH) (CONsole) {vaddr} </pre>	<pre> { [To] {userid} [HOLD] [CONT] [For] * [NOHold] [NOCont] [Class c] [Copy[*] nnn] ¹ [SYSTEM] [OFF] } </pre>
		<pre> [CLOSE] [Flash name nnn] ³ [FORM {form} [Dist {distcode}] [PURGE] [MODify name [n] {OFF} {OFF} Chars name1 [name2 [name3 [name4]] CHars name1 [Chars name2 [Chars name3 [Chars name 4]] FCB name </pre>
		<pre> [STArt] ² [Term] ² [STOp] [NOTerm] </pre>

¹At least one of the options within braces must be selected; however, more than one may be specified, and they may be entered in any order on the command line.

²These options apply only to a virtual spooled console.

³These options can only be used to modify a virtual spooling device. They only apply to a 3800 printer as a virtual spooling device.

where:

```
[READER]
[RDR]
```

modifies the options for all reader spool devices.

```
[PRINTER]
[PRT]
```

modifies the options for all printer spool devices.

**[PUNCH
PCH]**

modifies the options for all punch spool devices.

CONSOLE

modifies the options for the virtual console spool file and/or initiates or stops the spooling of virtual console input and output, including CP input/output.

vaddr

is the virtual device address (*cuu*) of the virtual unit record device or console whose options are to be modified.

CLASS c

*

specifies the spool class of the device. *c* is a one-character alphameric field whose values can be A through Z, or 0 through 9.

Unless your virtual reader class is asterisk (*), you must ensure that any files to be read by your virtual reader are of the same spool class as your virtual reader. The * is the universal class; if your virtual reader is class *, it can read any file, regardless of class.

CONT

ignores intermediate end-of-file indications or CLOSE requests. For virtual readers, reading is continuous, with all end-of-file indicators ignored until all files spooled to the virtual machine are read in. If this option is not in effect, a unit exception is reflected to the virtual machine at the end of each spooled file. CONT specified for the punch or printer causes all CLOSE requests to be ignored until reset by NOCONT. If CONT is specified, NOCONT cannot be specified. NOCONT is the default if you do not specify CONT.

CONT specifies that reading is to continue, without intervening end-of-file indications until all files in the system that belong to the user are read. If CONT is not in effect or is reset by specification of NOCONT, an end-of-file indication is reflected to the virtual machine at the end of each SPOOL file in the system. The nature of the end-of-file indication to be reflected is set by the EOF and NOEOF options. If the EOF option is in effect, end-of-file is signaled by a unit exception: this corresponds to pressing the end-of-file button on a real card reader. If NOEOF is in effect for a virtual reader, end-of-file is signaled by the reflection of a unit check/intervention required status.

NOCONT

resets the continuous spooling option. If NOCONT is specified, CONT cannot be specified. NOCONT is the default if CONT is not specified.

HOLD

places all files created by the specified device in a user hold status. For **READER** files, this option specifies that input files for the specified reader are not deleted from the system after they are read. The status of all files must be changed by the **CHANGE** command. The status of output devices is changed by the **SPOOL** command. If **HOLD** is specified, **NOHOLD** cannot be specified.

If the **HOLD** option is specified for a virtual printer or punch that is transferred to a user for input (**TO *userid***), that virtual device places a user hold status on the reader file. The user receiving the file cannot read its status until it is changed by issuing the **CHANGE** command with the **NOHOLD** operand. The spool file class of the virtual output device must match the class of the receiver's virtual reader (or the virtual reader must have a class of *****) in order for the spool file to be processed. If these conditions are not satisfied, the reader appears empty to the virtual machine attempting to read a file, even though reader files do exist.

If a virtual reader is operating with **CONT** and **HOLD**, then virtual reader files are saved and placed in a user **HOLD** status. The file cannot be read until it is changed (using the **CHANGE** command) to a **NOHOLD** status.

NOHOLD

resets the **HOLD** operand. Future files are not held. **NOHOLD** resets the **HOLD** operand in effect for the specified reader. This operand can be overridden for an active file being closed by the **CLOSE** command using the **HOLD** or **NOHOLD** operand. If **NOHOLD** is specified, **HOLD** may not be specified.

EOF

sets a virtual end-of-file condition on the specified reader, thereby ensuring that a unit exception condition is reflected on the read that follows the reading of the last card in a file. If **EOF** is specified, **NOEOF** may not be specified.

NOEOF

specifies that the reading continues to physical end-of-file. The virtual reader stops when no cards are left in the reader and when a unit check/intervention required status is pending. If **NOEOF** is specified, **EOF** may not be specified.

[TO] { *userid* }
 *
 SYSTEM }

transfers the output of the virtual device to the virtual card reader of the specified *userid*. If **TO** is omitted, the *userid* may not be **TO** or **T**. **TO *** may be coded if the output is to be transferred to your own virtual card reader. If **TO *userid*** is specified, neither **OFF** nor **FOR** may be specified on the same command line.

If you specify COPY with *TO userid*, the number of copies you specify has no effect on the receiver of the spool file; he receives only one copy. However, if OFF or FOR is specified on a subsequent command, the receiver of your spool file receives the number of copies you specify using COPY. For example, if the following command is entered:

```
spool pun to usera copy 3 class b
```

the COPY operand has no effect on the file going to USERA. However, if the command:

```
spool pun off
```

or

```
spool pun for usera
```

is entered, the COPY 3 specified in the first statement affects subsequent punch files.

TO SYSTEM is equivalent to specifying OFF and resets the transferred spool option.

FOR{userid
*
SYSTEM}

indicates the *userid* under which printed or punched output is produced. The *userid* becomes the owner of the output spool file and the *distcode* on the file is the *distcode* for the user that is specified in the directory. The file is not transferred to the user's reader input. The default setting is for your own virtual machine identification. FOR *, or FOR SYSTEM can be coded to specify your own identification and is equivalent to the OFF option.

OFF

resets the transferred spool option.

COPY[*]nnn

is the number of copies that are to be printed or punched when the file is spooled to the real unit record device. This operand is valid only for output files; the number of copies, *nnn* must be between 1 and 255 (leading zeros need not be specified). When the optional asterisk (*) is specified, output files spooled to the 3800 printer (see usage note "Spooling Output to a 3800 Printer" on page 223) are duplicated inside the device one page at a time. If a file is not spooled to the 3800 printer, the function performed by specifying the asterisk (*) is ignored and duplication occurs normally.

CLOSE

closes the specified device regardless of the CONT setting for the device. If CLOSE is specified, PURGE may not be specified. CLOSE does not affect the setting of any other operand and is provided as a convenience to close a virtual output device. As an example, this sequence of commands:

```
SPOOL PRT CONT
  (print file)
  (print file)
  (print file)
SPOOL PRT NOCONT
CLOSE PRT
SPOOL PRT CONT
  (print file)
  (print file)
  .
  .
  .
```

can be replaced with the following sequence to achieve the desired result:

```
SPOOL PRT CONT
  (print file)
  (print file)
  (print file)
SPOOL PRT CLOSE
  (print file)
  (print file)
  .
  .
  .
```

PURGE

closes and purges the spool file from the specified virtual output device regardless of the CONT setting for the device. If PURGE is specified, CLOSE cannot be specified. PURGE does not affect the setting of any other operand and is equivalent to issuing the CLOSE command for a device (or type of device) with the PURGE operand.

FLASH name nnn

signifies that a forms overlay contained in the 3800 printer is to be superimposed onto certain pages of the output. The name, one to four characters, identifies the forms overlay to be used in the projection. The number, *nnn*, can be anything from 0 to 255. The number specifies that the first *nnn* copies of output are to be superimposed. For example, if you specify COPY 4 and FLASH name 2, the first two copies of your file are flashed, while the last two copies remain unflashed. The default, 0, indicates no superimposing is desired.

MODIFY name [n]

assigns a copy modification module to be used when printing a file on the 3800. This function makes it possible to alter text by either preventing printing of certain information or adding labels to output. The name must be one to four characters. If the name is omitted, the file prints normally without modification. *n* selects one of the character arrangement tables specified with the CHARS keyword to be

used for the copy modification text. The values 0, 1, 2, or 3 correspond to the order in which the table names have been specified. If no *n* is specified, the first character arrangement table is assumed.

CHARS name1 [name2 [name3 [name4]]]

CHARS name1 [CHars name2 [CHars name3 [CHars name4]]]

specifies the name of the character arrangement table used when printing a file. The *name* must be from one-to-four characters with a maximum of four names given per command.

Note: The multiple use of CHars need only be used if NameN. would conflict with an option name.

FCB name

controls the vertical spacing of output on a page. The *name* must be from one-to-four characters. This parameter applies to output sent to 3800 printers only.

START

places all console input and output in a spool file. Until a CLOSE is issued for the console, characteristics of the console spool file may be changed by use of the SPOOL CONSOLE command. After the console is closed, the file becomes a printer spool file whose characteristics can be changed by issuing the CHANGE PRINTER command.

STOP

terminates the spooling of console input and output. The command SPOOL CONSOLE STOP does not close the console spool file.

TERM

displays the virtual console input and output at the terminal in addition to placing it in a spool file. The TERM operand has no effect until the START operand is specified.

NOTERM

suppresses the display of console input and output of a system running in a virtual machine. The display of console input and output is not suppressed if:

- It is the result of CP commands entered from CP mode.
- It is a message or warning.
- It is the result of a CP command entered by means of the #CP command.
- It is the result of commands entered on a 3270 in CMS EDIT mode.
- The output line immediately precedes a virtual machine read to the console. (This output line prints twice in the spooled console file.)

The NOTERM operand has no effect until the START operand is specified.

FORM form

specifies the form name for spool files created on this virtual device. 'form' is a one to eight characters name defined during system generation. This form can be overridden by the FORM option of the CLOSE command.

FORM OFF

resets the form for the virtual device to the system default as specified by the installation. When a virtual device is defined (at logon or with the DEFINE command), its form is set to this default.

DIST distcode

specifies the distribution code for spool files created on this virtual device. *distcode* is one to eight characters. This *distcode* can be overridden by the DIST option of the CLOSE command.

DIST OFF

resets the distribution code for the virtual device to the default specified in the directory for this virtual machine. When a virtual device is defined (at logon or with the DEFINE command), its *distcode* is set to this default.

Default Values for Spool Files

Unless otherwise set, the following options are default values for spool files:

Spool File	Default Options
Reader	NOHOLD NOCONT CLASS * ¹ EOF
Punch	NOHOLD NOCONT CLASS A ¹ OFF COPY 001
Printer	NOHOLD NOCONT CLASS A ¹ OFF COPY 001 FLASHC 000
Console	NOHOLD NOCONT CLASS T ¹ OFF COPY 001 TERM
¹ These are default classes only if a specific class option is not part of the virtual machine directory entry.	

Spool Class Considerations

Your system programmer can define certain spool classes to automatically print certain classification titles (for, example, TOP PRIORITY or CONFIDENTIAL). The title prints on the separator page, and, optionally, at the bottom of each printed page. Ask your system programmer if he has defined certain classes for this function, or request that classes be defined. For example, you may wish class P to print TOP PRIORITY on the separator page, and class X to print CONFIDENTIAL on the separator page and also on the bottom of each page of output.

Spooling Output to a 3800 Printer

In order to print on a 3800, the user can:

- Specify the correct FLASH name that was designated by the operator in the START command.
- Specify the correct spool file class, previously assigned to the 3800 printer by the installation.

A user can also nullify any previous or existing settings of the CHARS, MODIFY, or FCB options by specifying NULL in the name field of these special 3800 operands. However, if you spool a spool file containing 3800 options to a printer other than the 3800, the printer ignores the options and printing occurs normally. You cannot use the options available only on the 3800 Model 3 (for example, 10 lines per inch FCB) on the 3800 Model 1.

Also, the user can avoid FCB forms mismatch by specifying 6, 8, 12, or (for the 3800 Model 3 only) 10 in the name field of the FCB option to the SPOOL command. These special names force the entire spool file to print at 6, 8, 12, or (for the 3800 Model 3 only) 10 lines per inch respectively. VM/SP HPO calculates the proper FCB to load for these special names.

Notes:

1. *When the 3800 printer is activated, the operator specifies the character arrangement table and the FCB value that prints the separator page for every output file. If a user omits the CHARS option on a file spooled to the 3800, the printing defaults to the character arrangement table previously defined by the operator. This is also the case for the FCB option.*
2. *The system issues a message to the user whenever a file spooled to the 3800 printer contains invalid or undefined option names. If the operator has not automatically purged the file, the user can correct the situation using the CHANGE command or spool the file to another printer.*
3. *Due to the change in PEL density, customized 3800 Model 1 character sets are not interchangeable with the 3800 Model 3 character sets. Users may racode customized 3800 Model 1 character sets and build new modules through the use of the GENIMAGE command (described in VM/SP HPO Planning and Reference Guide and VM/SP HPO Operator's Guide). The MVS Character Conversion Aid may also be used to convert existing customized character sets to the 3800 Model 3 PEL density.*

The Spoolid Number: An Identifier for Your Spool Files

Once you close a spool file by issuing the CMS PRINT or PUNCH command or the CP CLOSE command, CP assigns the spool file a number between 1 and 9900. This number is called the spoolid (spool file identification) for the file. It can be used as a convenient way to identify the file. It can also be used when you are manipulating the file with spooling commands such as ORDER, CHANGE or CLOSE.

Spoolids are assigned to all your spool files sequentially. When the maximum number (9900) is assigned, CP begins again with the number 1.

When you print or punch a file, CP displays at your terminal the spoolid it assigned to your file if (1) multiple copies of the file are being processed, (2) the file is being transferred to another user, or (3) the file is placed in a user hold status. You can find out various kinds of information about a file using the spoolid with the many forms of the QUERY command.

Transmitting Spool Files Locally

Spool files can be transferred between users using the SPOOL and TRANSFER commands. To transfer printer or punch files of your virtual machine to the virtual reader of some other user, issue the SPOOL command with the TO option:

```
spool device to userid
```

where device is the virtual device type or virtual address (for example, PRINTER, PUNCH, 00E) and *userid* is the name of the designated user's reader.

If you only want to print or punch a file for someone else, you can issue the SPOOL command with the FOR option:

```
spool device for userid
```

where device is the same as the previous example. The *userid* provides the distribution code for the designated user.

To transfer printer or punch files from your virtual card reader to another user's reader, issue the TRANSFER command with the TO option:

```
transfer option to userid
```

where option is SPOOLID, CLASS, or ALL and *userid* is the designated user's virtual reader.

If you merely wish to reclaim a transferred file, you can issue the same command with the FROM option:

```
transfer option from userid
```

where *option* and *userid* are the same as the previous example.

Notes:

1. Files that have been spooled for another user are not in that user's virtual reader and, therefore, you cannot reclaim them.
2. You can transfer any file that is queued on your virtual reader; however, you can reclaim only those files which you originated.

Transmitting Files to Remote Locations

To direct files to remote stations, use the CP TAG and SPOOL commands in conjunction with a command that causes the file to be closed and sent to a virtual device (for example, a virtual printer or punch). Use the TAG command to specify the device to be spooled and to associate with that device the location identifier (*locid*) for the destination of the file:

```
tag dev device locid
```

where *device* is the virtual device type (for example, PRINTER or PUNCH) or virtual device address (*vaddr*) and *locid* is the name of the destination to which the file is to be transmitted.

Use the SPOOL command to specify that output to the device specified in the command is to be sent to the RSCS virtual machine, which performs the actual transmission of the file:

```
spool device to userid
```

where *device* is the same virtual device type or virtual device address specified in the TAG command and *userid* is the *userid* of the RSCS virtual machine at your installation. You can find out the *userid* of your installation's RSCS virtual machine and the *locid* for the various remote stations from your installation's system programmer.

After you issue the TAG and SPOOL commands, use a command (such as the CMS PRINT or PUNCH command or the CP CLOSE command) to cause the spool file to be generated, closed, and spooled to the specified virtual device.

Once you have closed a spool file that is spooled to another user, you cannot change or query the spool file tag. If the file has not been selected for processing by the RSCS virtual machine, you can use the TRANSFER command to get the file back to your virtual reader. When changes are necessary, you can issue the TAG QUERY DEV or FILE command to determine the TAG information associated with a given virtual device or spool file. You can then use the FILE operand of the TAG command to change a tag that was previously set.

The following example shows how to use these three commands to transmit a CMS file to a remote location:

```
tag dev punch cambridg
spool punch to net
punch myprog assemble
```

The TAG command defines the type of file to be transmitted, a punch file, and the remote station to which you want it transmitted, CAMBRIDG. NET is the *userid* of the virtual machine controlling the RSCS network: you direct your file to that virtual machine with the SPOOL command. The PUNCH command causes the file MYPROG ASSEMBLE to be punched on your virtual machine card punch, closed, and then spooled to the virtual reader of the RSCS virtual machine, which you specified in the SPOOL command. The RSCS virtual machine then processes your file (now a VM/SP HPO spool file) and transmits it across the RSCS network.

Receiving Files from a Remote Spooling Network

If your virtual machine is logged on VM/SP HPO, the remote spooling network notifies you of the arrival of a file for your machine from the RSCS network by displaying a message at your terminal. The file is sent to your virtual card reader.

VM/SP HPO can accumulate files from the network destined for your virtual card reader, regardless of whether you are logged on your virtual machine or not. If you are logged on your virtual machine, issue the QUERY command to see if you have any files in your virtual reader. When you logon your virtual machine, the logon process transmits a message informing you of accumulated spool files residing in your virtual reader (punch or printer).

Spooling Considerations During Shutdown and Abend Situations

CP does not automatically close user console spool files when VM/SP HPO abends or the operator issues the SHUTDOWN command. The abend or SHUTDOWN condition causes console records to be lost and, consequently, an incorrect record count for spool files.

Responses

If the SPOOL command is used to close an output device, the message

```
{PRT}
{PUN} FILE spoolid {TO } userid COPY nnn {HOLD }
{CON}              {FOR}                  {NOHOLD}
```

is received if (1) multiple copies of the file are being processed, (2) the file is being transferred to another user, or (3) the file is placed in a user hold status. However, if SET MSG OFF is in effect, this message is not issued, even though the above conditions are met.

STORE

Privilege Class: G

Use the STORE command to alter the contents of specified registers and locations of the virtual machine. The contents of the following can be altered:

- Virtual storage locations (second-level virtual storage only; see usage note 1 on page 229)
- General registers
- Floating-point registers
- Control registers (if available)
- Program status word

The STORE command can also save virtual machine data in low storage.

The format of the STORE command is:

Store	<pre> hexloc Lhexloc hexword1 [hexword2...] Shexloc hexdata... {Greg} {Xreg} hexword1 [hexword2...] {Yreg} hexword1 [hexword2...] Psw [hexword1] hexword2 STATUS </pre>
-------	--

where:

hexloc

Lhexloc hexword1 [hexword2...]

stores the specified data (*hexword1* [*hexword2...*]) in successive fullword locations starting at the address specified by *hexloc*. The smallest group of hexadecimal values that can be stored using this form is one fullword. Either form (*hexloc* or *Lhexloc*) can be used.

If *hexloc* is not on a fullword boundary, it is rounded down to the next lower fullword. The operands *hexword1* [*hexword2...*] each represent up to 16 hexadecimal digits. If the value being stored is less than a fullword (eight hexadecimal digits), it is right-adjusted in the word and the high order bytes of the word are filled with zeros. If two or more *hexwords* are specified, they must be separated by one or more blanks.

Shexloc hexdata...

stores the data specified (*hexdata...*) in the address specified by *hexloc*, without word alignment. The shortest string that can be stored is one byte (two hexadecimal digits). If the string contains an odd number of characters, the last character is not stored, an error message is sent, and the function is terminated. The *hexdata...* operand is a string of two or more hexadecimal digits with no embedded blanks.

Greg hexword1 [hexword2...]

stores the hexadecimal data (*hexword1* [*hexword2...*]) in successive general registers starting at the register specified by *reg*. The *reg* operand must be either a decimal number from 0 through 15 or a hexadecimal digit from 0 through F. The *hexword1* [*hexword2...*] operands each represent up to eight hexadecimal digits. If the value being stored is less than a fullword (eight hexadecimal digits), it is right-adjusted in the word and the high order bytes of the word are filled with zeros. If two or more *hexwords* are specified, they must be separated by one or more blanks.

Xreg hexword1 [hexword2...]

stores the hexadecimal data (*hexword1* [*hexword2...*]) in successive control registers starting at the register specified by *reg*. The *reg* operand must either be a decimal number from 0 through 15 or a hexadecimal digit from 0 through F. If the virtual machine is in basic control mode, you can store data in register 0 only. The *hexword1* [*hexword2...*] operands each represent up to eight hexadecimal digits. If the value being stored is less than a fullword (eight hexadecimal digits), it is right-adjusted in the word and the high order bytes of the word are filled with zeros. If two or more *hexwords* are specified, they must be separated by one or more blanks.

Yreg hexdword1 [hexdword2...]

stores the hexadecimal data (*hexdword1* [*hexdword2...*]) in successive floating-point registers starting at the register specified by *reg*. The *reg* operand must be a digit from 0 through 7. If *reg* is an odd number, it is adjusted to the preceding even number. The *hexdword1* [*hexdword2...*] operand each represents up to 16 hexadecimal digits. If the value being stored is less than a doubleword (16 hexadecimal digits), it is left-justified in the doubleword and low order positions are filled with zeros. If two or more hexadecimal words are specified, they must be separated by one or more blanks.

PSW [hexword1] hexword2

stores the hexadecimal data in the first and second words of the virtual machine's program status word (PSW). If you specify *hexword1*, the low order two bytes of this word are replaced with zeros before *hexword1* is stored in the first word of the PSW. *Hexword2* is stored in the second word of the PSW whether or not you specify *hexword1*. The [*hexword1*] *hexword2* operands each represent up to eight hexadecimal digits. These operands must be separated by one or more blanks. If the value being stored is less than a fullword (eight hexadecimal digits), it is right-adjusted in the word and the high order bytes of the word are filled with zeros.

STATUS

stores selected virtual machine data in certain low storage locations of the virtual machine, simulating the hardware store status facility. These locations are permanently assigned locations in real storage. To use the STATUS operand, your virtual machine must be in the extended control mode. The STATUS operand should not be issued for CMS virtual machines or for DOS virtual machines generated for a processor smaller than a System/360 Model 40. The STATUS operand stores the following data in low storage:

Decimal Address	Hexadecimal Address	Length in Bytes	Data
216	D8	8	CPU Timer
224	E0	8	Clock Comparator
256	100	8	Current PSW
352	160	32	Floating-point registers 0-6
384	180	64	General registers 0-15
448	1C0	64	Control registers 0-15

Usage Notes

1. First-level storage is real main storage. Only second-level storage (storage that is real to the virtual machine) can be stored into. Operating systems running in a virtual machine such as DOS/VS and OS/VS have virtual storage of their own. This third-level virtual storage cannot be stored into directly. The user or the virtual operating system is responsible for converting any third-level storage locations to second-level storage locations.
2. The operands may be combined in any order desired, separated by one or more blanks, for up to one full line of input. If an invalid operand is encountered, an error message is issued and the store function is terminated. However, all valid operands entered before the invalid one are processed properly.
3. If you combine the operands for storing into storage, registers, the PSW, or the status area on a single command line, all operands must be specified; default values do not apply in this case.
4. If the STORE command is used by your virtual machine to alter the contents of a shared segment, you receive a warning message and your virtual machine is unshared from (given a private copy of) the named system. Other users of the segment continue to operate with an unchanged copy of that segment.
5. With the STORE command, data is stored either in units of one word with fullword boundary alignment or in units of one byte without alignment.
6. The STORE STATUS command stores data in the extended logout area. The STORE STATUS command stores CPU Timer and Clock Comparator values that may then be displayed at the terminal using the DISPLAY command. This procedure is the only way to get timer information at the terminal using a CP command.

STORE

7. You should issue the **STORE STATUS** command while in **CP READ** if a subsequent stand-alone dump is to be taken, to ensure that the information in the dump will correspond to the stored data. This prevents the virtual machine from continuing operation before the dump is taken.
8. If you are the **V=R** user running with preferred machine assist enabled, you cannot use the **STORE** command to store data above the system area (the first 16 Mb of storage). **CP** cannot service storage above this area.

Response

STORE COMPLETE

is the response at the successful completion of the command.

SYSTEM

Privilege Class: G

Use the SYSTEM command to simulate the action of the RESET and RESTART buttons on the real computer console, and to clear storage.

The format of the SYSTEM command is:

SYStem	{ CLEAR RESET RESTART }
--------	-------------------------------------

where:

CLEAR

clears virtual storage and virtual storage keys to binary zeros.

RESET

clears all pending interruptions and conditions in the virtual machine.

RESTART

simulates the hardware system RESTART function by storing the current PSW at virtual location eight and loading, as the new PSW, the doubleword from virtual location zero. Interruption conditions and storage remain unaffected.

Usage Notes

1. The RESET function and the CLEAR function leave the virtual machine in a disabled wait state. Both CLEAR and RESET clear pending interruptions, both functions load a PSW of all zeros with only the WAIT bit set on (X'0002000000000000').
2. After issuing the SYSTEM command with RESET or CLEAR specified, either STORE a PSW and issue BEGIN or issue BEGIN with a hexadecimal storage location specified, to resume operation. The virtual machine automatically restarts at the location specified in the new PSW (which is loaded from the doubleword at location zero) after the SYSTEM RESTART command is processed.
3. If you issue the SYSTEM CLEAR or SYSTEM RESET command to reset preferred machine assist, you must issue an IPL command with the PMA operand specified to resume working with preferred machine assist enabled.

Responses

STORAGE CLEARED - SYSTEM RESET

is the response given if the command SYSTEM CLEAR is entered.

SYSTEM RESET

is the response given if the command SYSTEM RESET is entered.

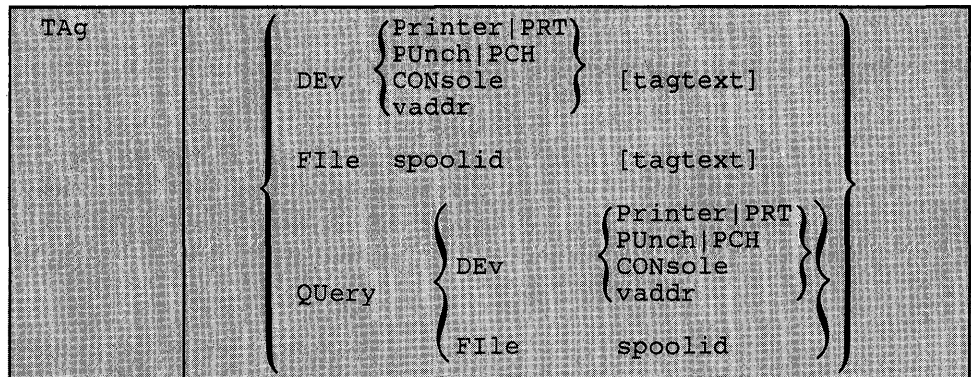
If the command SYSTEM RESTART is entered, no response is given; the virtual machine resumes execution at the address in the virtual PSW loaded from virtual storage location X'0'.

TAG

Privilege Class: G

Use the TAG command to associate file descriptive information with a spool file.

The format of the TAG command is:



where:

DEV { [PRINTER]
[PRT]
[PUNCH]
[PCH]
[CONSOLE]
vaddr }

specifies a previously closed spooling device whose output is to be associated with the tag information or whose tag information is to be queried.

Note: When a generic device name, such as PRINTER or CONSOLE is used in the TAG command, all current virtual devices of that type are affected.

FILE spoolid

specifies a previously closed spool file whose tag information is to be replaced or queried. This form of the TAG command can be used for spool files that are on your reader, printer, or punch queues.

The spoolid operand is the spool file identification, a number between 1 and 9900, assigned by CP when the spool file was closed.

tagtext

is the information (up to 134 characters in length, including imbedded blanks) that is to be associated with the specified spool device or spool file. The contents and format of this data is completely flexible and is the responsibility of the file originator and the end user.

Certain control and addressing information, meaningful to the RSCS component of VM/370, or the RSCS Networking program product (5748-XP1), can be specified in this field. For details on how to use the tagtext operand when transmitting files across the RSCS network, refer to usage note 1.

QUERY	{	DEV	{	[PRINTER]	}	}
					PRT			
					PUNCH			
					PCH			
					CONSOLE			
					vaddr			
		FILE			spoolid			

displays, at your terminal, the current tag information associated with a given virtual device or spool file. The operands used with the TAG QUERY command correspond to the operands used with TAG itself.

Usage Notes

1. The RSCS control program interprets the tag information as being addressing and control parameters. When you spool a file to the RSCS virtual machine to be transmitted to a remote station, code the tagtext operand as follows:

```
locid [userid] [priority]
```

where:

locid

is the location identifier (one-to-eight alphanumeric characters) of the location to which the file is being transmitted. Your system programmer can give you the *locids* of remote stations accessible to your virtual machine.

userid

is the *userid* of the virtual machine (a one-to-eight character user identification) to which a file is being transmitted. This operand is used by remote stations when they transmit files to a virtual machine in a remote spooling network and want the files sent to a particular virtual machine. You can ignore this operand if you are not specifying a priority. However, if you are specifying a priority, you must code some *userid* operand; it is ignored in a remote spooling network.

priority

is the requested transmission priority, a decimal number between 0 and 99. The highest transmission priority is 0, next highest is 1, and so on. If you wish to specify this operand, you must also specify a *userid* operand.

2. If you enter the TAG command with no tagtext information, the tag area associated with the device or file is set to all blanks.

3. A spool file that you have created and spooled to another virtual machine is owned by the other user as soon as the file is closed. If you want to query or change the tag data, you must first reclaim the file using the TRANSFER command.
4. Although the buffer that contains TAG information is 134 characters in length including imbedded blanks, only 126 characters of it are usable because of input command line limitations.
5. The TAG command works only on files that are not currently in use.
6. There are four types of spool files that can't be tagged. They are:
 - a. Monitor files
 - b. Accounting files
 - c. Dump files
 - d. Real card reader (Input files)

Responses by Command Operands

TAG QUERY DEV { PRINTER }
 { PUNCH }
 { CONSOLE }

Displays the contents of the tags associated with each device of the specified class, as follows:

{ PRT }
 { PUN } vaddr TAG:
 { CONS }

tagtext.....

repeated for each device in the specified class

TAG QUERY DEV vaddr

Displays the contents of the tag associated with the specified device, as follows:

{ PRT }
 { PUN } vaddr TAG:
 { CONS }

tagtext.....

if tag data exists

TAG NOT SET

if the TAG command was never issued to that device.

TAG QUERY FILE spoolid

Displays the contents of the tag associated with the specified file, as follows:

tagtext.....
if tag data exists

(TAG BLANK)
if the tag is all blanks

(TAG MISSING)
if the file did not contain a tag because it was either an input file from the real card reader or was an output file generated prior to VM/370 Release 2 PLC 11.

TERMINAL

Privilege Class: G

Use the **TERMINAL** command to control the following functions associated with your virtual console:

- Logical line-editing symbols
- Masking of password
- The APL character set
- The Text character set
- Signaling of an attention interrupt
- Attention handling mode for your virtual console
- Line length for output on your virtual console
- Specifying terminal device type as 3101 or TTY
- Location of cursor preceding terminal read
- Scrolling rate for 3101 terminal

TERMINAL

The format of the TERMINAL command is:

TERMinal	}	CHardel	{ ON }	}	1
		LINEDel	{ OFF }		
		LINeNd	{ char }		
		EScape			
		TABchar			
		APL	{ ON }		
		TEXT	{ OFF }		
		ATtn			
		HIlight			
		SCRNsave			
		MODE	{ CP }		
			{ VM }		
		LINESize	{ nnn }		
			{ OFF }		
		CONmode	{ 3215 }		
			{ 3270 }		
		BREakin	{ IMmed }		
	{ GUESTct1 }				
BRKkey	{ PA1 }				
	{ PF1 }				
	{ . }				
	{ . }				
	{ PF24 }				
TYpe	{ 3101 }				
	{ TTY }				
PROMpt	{ VM }				
	{ TTY }				
SCROll	{ nnn }				
	{ CONT }				

¹More than one function can be specified in a single entry of the TERMINAL command. For example:

```
TERMINAL CHARDEL OFF LINESIZE 20 MODE VM
```

where:

CHARDEL { ON }
 { OFF }
 { char }

defines the logical character delete symbol. If ON is specified, the default symbol becomes the logical character delete symbol. The default symbol is normally @, but depends on what is specified in your virtual machine directory entry. If OFF is specified, no logical character delete symbol is allowed. If char is specified, then that character becomes a logical character delete symbol. The character chosen should be unique and not common to the data stream being entered. When you logon, CHARDEL ON is in effect; if your virtual machine is logged on by the system operator using the AUTOLOG command, CHARDEL OFF is the default.

LINEDEL { **ON** }
 { **OFF** }
 { **char** }

defines the logical line delete symbol. If ON is specified, the default symbol becomes the logical line delete symbol. The default symbol is normally ϕ , but depends on what is specified in your virtual machine directory entry. If OFF is specified, no logical line delete symbol is allowed. If char is specified, then that character becomes the logical line delete symbol. The character chosen should be unique and not common to the data stream being entered. When you logon, LINEDEL ON is in effect; if your virtual machine is logged on by the system operator using the AUTOLOG command, LINEDEL OFF is the default.

LINEND { **ON** }
 { **OFF** }
 { **char** }

defines the logical line end symbol. If ON is specified, the default symbol becomes the logical line end symbol. The default symbol is normally #, but depends on what is specified in your virtual machine directory entry. If OFF is specified, no logical line end symbol is allowed. If char is specified, then that character becomes the logical line end character. The character chosen should be unique and not common to the data stream being entered. When you logon, LINEND ON is in effect; if your virtual machine is logged on by the system operator using the AUTOLOG command, LINEND OFF is the default.

ESCAPE { **ON** }
 { **OFF** }
 { **char** }

defines the logical escape character. If ON is specified, the default symbol becomes the logical escape character. The default symbol is normally ", but depends on what is specified in your virtual machine directory entry. If OFF is specified, no logical escape character is allowed. If char is specified, then that character becomes the logical escape character. The character chosen should be unique and not common to the data stream being entered. When you logon, ESCAPE ON is in effect; if your virtual machine is logged on by the system operator using the AUTOLOG command, ESCAPE OFF is the default.

TABCHAR { **ON** }
 { **OFF** }
 { **char** }

allows a user to define a variety of logical tab characters on graphics devices. QUERY TERMINAL command displays the current logical tab character.

ON sets the logical tab character to the one used by the system.

OFF sets the logical tab character to X'00'.

char defines the logical tab character to be used by the system internally. It may be any valid special graphic character. When an invalid character is specified, an error message is issued to the terminal and the logical tab character remains unchanged.

**APL {ON }
{OFF }**

controls the use of APL character translation tables. If APL ON is specified, CP uses the translation tables applicable to display stations equipped with the APL hardware feature and typewriter terminals equipped with the standard APL typing element. APL ON also applies to the 3767 terminal equipped with the APL alternate character selection. If APL OFF is specified, CP uses the normal translation tables (that is, BCD or correspondence code). Unless otherwise specified, APL OFF is in effect.

If APL is on, TEXT is forced off.

Notes:

1. *This operand is valid for certain 3270 Information Display Stations. See VM/SP HPO Planning Guide and Reference for applicable APL terminals.*
2. *If the TERMINAL APL ON command is issued from a terminal that is not equipped with APL keys, or if the user's virtual machine is not running APL when the TERMINAL command is issued, the results may be unpredictable.*

If APL ON is specified, the LINESIZE value is overridden (see the explanation for the LINESIZE operand, below). However, the setting of the SET LINEDIT command is independent of the TERMINAL APL setting.

**TEXT {ON }
{OFF }**

controls the use of Text translation tables. If TEXT ON is specified, CP uses the translation tables applicable to display terminals equipped with the appropriate Text hardware features. If TEXT OFF is specified, CP uses the normal translation tables. Unless otherwise specified, TEXT OFF is in effect. When TEXT ON is specified, APL OFF is forced.

Notes:

1. *This operand is valid for certain 3270 Information Display Stations. See VM/SP HPO Planning Guide and Reference for applicable TEXT terminals.*
2. *The setting of the SET LINEDIT command is independent of the TERMINAL TEXT setting.*

ATTN { ON }
 { OFF }

controls signaling of an attention interruption. If ATTN ON is specified, the exclamation point is displayed when an attention interruption occurs. The OFF option suppresses the displaying of the exclamation point (!) and carrier return for those systems that perform special line editing using the attention key. Unless otherwise specified, ATTN ON is in effect.

Note: The ATTN operand is not valid for display type terminals.

HILIGHT { ON }
 { OFF }

highlights the command line that a user enters on a terminal. When the feature is deactivated, user input is displayed at normal intensity on the screen. When you logon, HILIGHT is set off.

SCRNSAVE { OFF }
 { ON }

allows you, as the console operator, to choose between having the guest screen saved when going into CP mode, or not having the guest screen saved. This option is only valid if you have a local 3270 Information Display device and you have issued the CONMODE 3270 operand of the TERMINAL command. If you specify ON with this option, the screen is saved by CP and redisplayed when the guest gets control again without an ATTN interruption. OFF specifies that CP does not save the guest screen (that is, a CLEAR interruption is posted by CP to the guest when CP gives control back to the guest). OFF is the default. SCRNSAVE ON is reset to OFF following a disconnect.

Note: SCRNSAVE ON must be specified if running a guest system control program such as MVS/SP with CONMODE 3270. Otherwise, if guest screen is not saved, unpredictable results can occur due to message traffic from CP. With SCRNSAVE set off, no problems should occur unless message traffic through CP exists.

MODE { CP }
 { VM }

controls the terminal attention environment. CP specifies that one or more attentions force the virtual machine into the CP environment. VM specifies that one attention is reflected to your virtual machine and that more than one attention forces your virtual machine into the CP environment. VM is the default for all VM/SP HPO users except the primary system operator. For more information on this, see 'Controlling the Virtual Machine' in Section 3.

LINESIZE {**nnn**} {**OFF**}

nnn specifies the maximum allowable line length for terminal output. nnn can be a number from 1 through 255. OFF specifies that the terminal output line length is not to be adjusted by CP. In other words, CP does not split a long output line into two or more shorter ones. If neither nnn nor OFF is specified, the default value is device dependent.

Note: Except for TTY terminals, if APL ON is specified, CP does not separate output lines into LINESIZE segments. Instead, an output length of 1760 is allowed and CP assumes that the APL system has inserted the appropriate carriage control characters.

CONMODE {**3215**} {**3270**}

allows you to choose between 3215 and 3270 support at an SIO level. The initial definition is determined by having the console mode in the CONSOLE statement in your directory. You must have a local display device that has a 3270 compatible command set. If you specify 3215, then 3215 CCW command code processing occurs at an SIO level from the guest virtual machine. If you specify 3270, then 3270 CCW command code processing occurs at an SIO level from the guest virtual machine. The default setting is 3215. You must set CONMODE to 3215 to run CMS. CMS sets CONMODE to 3215 at IPL time, but if you dynamically reset CONMODE to 3270 while in CMS mode, results can be unpredictable. CONMODE 3270 is not supported for disconnected users. Also, if you disconnect a guest virtual machine and then reconnect, the CONMODE setting is unpredictable.

BREAKIN {**IMMED**} {**GUESTCTL**}

allows you to control break in by CP. This option is available only to users with local 3270 Information Display devices. The GUESTCTL option specifies that, while in full screen mode, messages from CP are only displayed when you hit the BRKKEY key. If you specify GUESTCTL, CP break in still occurs if you request a CP function or if a priority message is pending for your virtual machine. If you specify IMMED, the default, break in by CP takes place on any full screen READ and full screen WRITES that have an inactive keyboard; they also take place when you hit the BRKKEY.

BRKKEY { PA1
PF1
.
.
PF24 }

allows the PA1 key or a PF key you specify to control break in by CP for full screen guests. PA1 is the default. You must be sure that the PF key associated with this function is available on the guest virtual machine console. Choosing a PF key controller that does not exist on the guest virtual machine console hardware as the break in controller can lock out CP mode. Pressing PA1 will cause you to enter the CP environment even when in line mode. Some applications (for example, the System Product Editor) also interpret PA1 as a request for CP mode.

TYpe { 3101
TTY }

allows you to define a terminal as a combination keyboard and display screen (3101) device, or as a keyboard-printer device (TTY). In addition to the functions provided by a keyboard-printer terminal, specifying 3101:

- Allows use of the PROMPT operand to control the position of the cursor following a prompting sequence.
- Allows use of the SCROLL operand to control output pacing
- Allows use of program function keys one through 24.
- Causes output to begin at column 1 of line 24 on the display
- Allows editing of previously entered information without the necessity of rekeying the entire text.

PROMpt { VM
TTY }

causes the cursor to be positioned in column 1 of the input line for a read operation if the VM option is specified. If the TTY option is specified, the cursor is located in column 2 (with a period in column 1).

SCROLL { nnn
CONT }

nnn allows you to specify the number of lines that are to be scrolled up a TTY terminal device before scrolling stops with *****MORE***** displayed in the status area. Pressing the NEW LINE key (character mode) or the SEND key (block mode) causes another cycle of nnn lines to be scrolled. For normal frame by frame scrolling, make nnn equal to the number of data lines on the screen. Specifying CONT causes continuous scrolling to the end of a file.

Usage Notes

1. The terminal settings you specify with the **TERMINAL** command are in effect only for the duration of that terminal session. Whenever you initially logon, the system defaults are in effect. However, the settings you specify for line-editing (except **LINEDEL**) and **MODE** are still in effect when you logon after disconnecting. All the other operands (**TEXT**, **APL**, **ATTN**, **HILIGHT**, and **LINESIZE**) are reset if you logon after disconnecting.
2. The **CONMODE 3270** and **SCRNSAVE** options of the **TERMINAL** command are valid for the duration of the terminal session only, and are not supported across disconnect of the virtual machine. These options must be reset, if desired, upon reconnect.
3. Although you can define line-editing symbols and status with the **TERMINAL** command, the **LINEDIT** operand of the **SET** command determines whether the VM/SP HPO line-editing functions are on or off.
4. If an error occurs during processing of the command, all functions preceding the one with the error are in effect.
5. You cannot use any of the letters A through Z, or the numbers 0 through 9, as a symbol (char) in association with any of the **CHARDEL**, **LINEDEL**, **LINEND**, and **ESCAPE** operands.
6. Figure 10 describes the other features available for highlighting because the **TERMINAL** command cannot activate or deactivate these features.

Highlighted Feature	Designated By:
Certain data fields for application programs running under CMS	Application programmer
Logon logo	System programmer
Non-edit CMS and CP display sessions: Messages or warnings from the system operator or other users	System

Figure 10. 3270 High Intensity Features

7. The **SCROLL**, **TYPE**, and **PROMPT** operands must be issued from a TTY device or an error message results.

8. CP automatically masks your password on typewriter terminals. For installations migrating from a VM/370 operating system to VM/SP HPO, and for compatibility purposes, CP will accept the MASK operand on the TERMINAL command as valid. However, there is no need to specify this operand since CP automatically masks your password.
9. If your virtual machine is not accepting interruptions and SCRNSAVE is set OFF, you must hit the BRKKEY twice in order to enter CP mode.
10. Results are unpredictable if you issue both MODE VM and SCRNSAVE ON.

Responses

None.

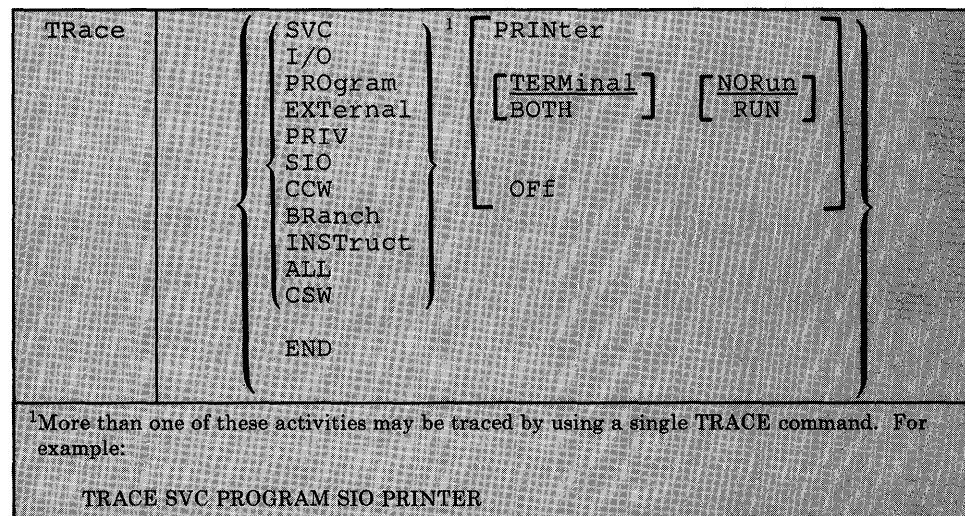
TRACE

Privilege Class: G

Use the TRACE command to trace specified virtual machine activity and to record the results at the terminal, on a virtual spooled printer, or on both terminal and printer. If you issue more than one TRACE command, the operands are cumulative; that is, operands specified for the first time are activated, whereas those specified with new modifiers are updated. The RUN and NORUN operands, however, can be specified in different TRACE commands for different tracing functions and do not cause a conflict.

You cannot issue the TRACE command while preferred machine assist is operating whether or not you have enabled the preferred machine assist's control switch assist.

The format of the TRACE command is:



where:

SVC
traces virtual machine SVC interruptions.

I/O
traces virtual machine I/O interruptions.

PROGRAM
traces virtual machine program interruptions.

EXTERNAL
traces virtual machine external interruptions.

PRIV
traces all virtual machine non-I/O privileged instructions.

SIO

traces TIO, CLRIO, HIO, HDV, and TCH instructions to devices, both spooled and non-spooled. SIO and SIOF instructions not be traced for spooled devices or consoles.

It also traces SIO instructions to attached remote printer and dedicated remote terminals.

CCW

traces virtual and real CCWs for nonspool nonconsole device operations. Virtual and real CCWs are not traced for remote printers and dedicated remote graphics terminals. When CCW tracing is requested, SIO and TIO instructions to all devices are also traced.

BRANCH

traces virtual machine interruptions, PSW instructions, and successful branches.

If you are a V=R user running in single processor mode, you cannot issue this command.

INSTRUCT

traces all instructions, virtual machine interruptions, and successful branches.

If you are a V=R user running in single processor mode, you cannot issue this command.

ALL

traces all instructions, interruptions, successful branches, privilege instructions, and virtual machine I/O operations.

If you enter this command from MVS cross memory privileged instructions, you will not see the PC, PT, LASP, or SSAR privileged operations.

If you are a V=R user running in single processor mode, you cannot issue this command.

CSW

provides contents of virtual and real channel status words at I/O interrupt.

END

terminates all tracing activity and prints a termination message.

PRINTER

directs tracing output to a virtual spooled printer. If PRINTER is specified, RUN is assumed; however neither RUN nor NORUN can be specified. Tracing output goes to the virtual printer having the lowest virtual address.

TERMINAL

directs tracing output to the terminal (virtual machine console).

BOTH

directs tracing output to both a virtual spooled printer and the terminal.

OFF

halts tracing of the specified activities on both the printer and terminal.

NORUN

stops program execution after the trace output to the terminal and enters the CP command environment.

Note: If a Diagnose code X'008' is being traced, NORUN has no effect and program execution does not stop.

RUN

continues the program execution after the trace output to the terminal has completed and does not enter the CP command environment.

Usage Notes

1. If your virtual machine has the virtual=real option and NOTRANS set on, CP forces CCW translation while tracing either SIO or CCW. When tracing is terminated with the TRACE END command, CCW translation is bypassed again.
2. If VM/SP HPO hardware assist is enabled on your virtual machine, CP turns it off while tracing SVC, PRIV, BRANCH, INSTRUCT, or ALL activities. If virtual interval timer support is available on your processor and is active (that is, SET ASSIST TMR is in effect), CP turns the timer support off while tracing external interruptions. After the tracing is terminated with the TRACE END command, CP turns hardware assist on again.
3. If trace output is being recorded at the terminal, the virtual machine stops execution and CP command mode is entered after each output message. This simulates the instruction step function.

However, all processing associated with the event being traced is completed and, therefore, execution may have stopped after an instruction has executed and the PSW has been updated.

For example, a privileged instruction traced with the PRIV operand stops after the privileged instruction executes, whereas the same instruction traced with the ALL operand stops before the instruction executes.

To determine whether the traced instruction has executed, display the virtual machine PSW.

To resume operation of the virtual machine, the BEGIN command must be entered. If the RUN operand is specified, the virtual machine is not stopped after each output message.

4. If trace output is being recorded on a virtual spooled printer, a CLOSE command must be issued to that printer in order for the trace output to be printed on the real printer.
5. Successful branches to the next sequential instruction and branch-to-self instructions are not detected by TRACE.
6. RUN and NORUN can be specified in different TRACE commands with no conflict as long as the functions being traced are different.
7. Instructions that modify or examine the first two bytes of the next sequential instruction cause erroneous processing for BRANCH and INSTRUCT tracing.
8. When tracing on a virtual machine with only one printer, the trace data is intermixed with other data sent to the virtual printer. To separate trace information from other data, define another printer with a lower virtual address than the previously defined printer. For example, on a system with 00E defined as the only printer, define a second printer as 00B. The regular output goes to 00E and the trace output goes to 00B.
9. If the BRANCH, INSTRUCT, or ALL activities are being traced by a virtual machine using a shared system, the user receives a warning message and the virtual machine is unshared from (given a private copy of) the named system. Other users of the segment continue to operate with an unchanged copy of that segment.
10. I/O operations for virtual channel-to-channel devices, with both ends connected to the same virtual machine, cannot be traced.
11. Use the TRACE command to trace specified virtual machine activity and to record the results at the terminal, at a virtual printer, or at both. This command is useful in debugging programs because it allows you to trace only the information that pertains to a particular problem.
12. If your virtual machine is performing I/O that results in program-controlled interruptions (PCIs), and you are tracing I/O or CSW activity, some of the PCIs may not be traced. This situation arises when the system is extending its free storage area and the additional demand on available free storage can cause a system abend.
13. When you are using the Group Control System (GCS), a TRACE at an address within the common GCS area can change the instruction at that address and affect the entire group. Therefore, use the PER command (described on "PER" on page 133) to debug rather than ADSTOP or TRACE.
14. The PA1 key on a 3270 display terminal should not be pressed with TRACE ALL RUN active while the display screen indicates RUNNING status as a subsequent CP command entered from the console may result in interleaved output.
15. There are some modules in CMS that do not issue SIOs. The I/O operation is initiated via DIAGNOSE instructions.

TRACE

16. If you are the V=R user running in single processor mode, you cannot issue the TRACE BRANCH, TRACE ALL, or TRACE INSTRUCT commands.
17. TRACE does not list operations resulting from MVS cross memory privileged instructions. If you enter the TRACE ALL command, you will not see the PC, PT, LASP, or SSAR privileged operations.

Responses

The following symbols are used in the responses received from TRACE:

Symbol	Meaning
vvvvvv	virtual storage address
ttttt	virtual transfer address or new PSW address
rrrrr	real storage address
xxxxxxx	virtual instruction, channel command word, CSW status
yyyyyyyy	real instruction, CCW
ss	argument byte (SSM-byte) for SSM instruction
ns	new system mask after execution of STOSM/STNSM
zz	low order byte of R1 register in an execute instruction (not shown if R1 register is register 0)
zzzzzzzz	referenced data
type	virtual device name (DASD, TAPE, LINE, CONS, RDR, PRT, PUN, GRAF, DEV)
V vaddr	virtual device address (<i>cuu</i>)
R raddr	real device address (<i>cuu</i>)
mnem	mnemonic for instruction
int	interruption type (SVC, PROG, EXT, I/O)
code	interruption code number (in hexadecimal)
CC n	condition-code number (0, 1, 2, or 3)
IDAL	Indirect data address list
***	virtual machine interruption
:::	privileged operations
= = >	transfer of control
SEEK	read/write position on count-key-data device
REAL EXTENT	real block location on a fixed-block device
VIRT EXTENT	virtual block location on a fixed-block device
LOCATE	read/write block position within extent

TRACE STARTED

This response is issued when tracing is initiated.

TRACE ENDED

This response is issued when tracing is suspended.

TCH, TIO, CLRIO, HIO, HDV, SIO, or SIOF**TCH**

I/O vvvvvv TCH xxxxxxxx type vaddr CC n

TIO, CLRIO, HIO, or HDV

I/O vvvvvv mnem xxxxxxxx type vaddr CC n type raddr CSW xxxx

SIO or SIOF

I/O vvvvvv mnem xxxxxxxx type vaddr CC n type raddr CSW xxxx CAW
vvvvvvvv

CCW:

```
CCW vvvvvv xxxxxxxx xxxxxxxx rrrrrr yyyyyyyy yyyyyyyy
CCW IDAL vvvvvvvv vvvvvvvv IDAL 00rrrrrr 00rrrrrr
CCW SEEK xxxxxxxx xxxxxx SEEK yyyyyyyy yyyy
CCW SEEK HEAD xxxxxx SEEK HEAD yyyy
CCW REAL EXTENT yyyyyyyy yyyyyyyy yyyyyyyy yyyyyyyy
CCW VIRT EXTENT xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx
CCW LOCATE xxxxxxxx xxxxxxxx
```

The IDAL or SEEK lines, as well as the REAL EXTENT, VIRT EXTENT, or LOCATE lines, are included only if applicable. The virtual IDAL is not printed if the real CCW operation code does not match the real CCW. With a SEEK HEAD CCW, virtual data is being used to execute at the hardware. There is no arm movement to the cylinder since it is already at the correct cylinder. there is no data relocation. Only the two bytes containing the head (track) are used. Therefore, with the CCW trace for SEEK HEAD, only the head (track) is output.

INSTRUCTION TRACING:**Privileged Instruction:**

```
::: vvvvvv SSM xxxxxxxx ss (normal SSM)
::: vvvvvv SSM xxxxxxxx ss tttttt (switch to/from translate mode)
::: vvvvvv STOSM xxxxxxxx ns (normal STOSM)
::: vvvvvv STOSM xxxxxxxx ns tttttt (switch to translate mode)
::: vvvvvv STNSM xxxxxxxx ns (normal STNSM)
::: vvvvvv STNSM xxxxxxxx ns tttttt (switch from translate mode)
::: vvvvvv LPSW xxxxxxxx tttttttt tttttttt (WAIT bit on)
::: vvvvvv LPSW xxxxxxxx ==> tttttttt tttttttt (WAIT bit not on)
::: vvvvvv mnem xxxxxxxx (all others)
```

Executed Instructions:

vvvvvv EX xxxxxxxx zz vvvvvv mnem xxxx xxxxxxxx

TRACE

For an executed instruction, where zz (see preceding explanation of symbols) is nonzero, the mnemonic for the executed instruction is given as if the zz byte had been put into the instruction with an OR operation.

All Other Instructions:

vvvvvv mnem xxxxxxxx xxxx

SUCCESSFUL BRANCH:

vvvvvv mnem xxxxxxxx == > ttttt

INTERRUPTION (SVC, PROGRAM, or EXTERNAL)

*** vvvvvv int code == > ttttt

I/O INTERRUPTION (First line given only if "CSW" was specified):

CSW V vaddr xxxxxxxx xxxxxxxx R raddr yyyyyyyy yyyyyyy

*** vvvvvv I/O vadd == > ttttt CSW xxxx

BRANCH TRACE: (ALL option selected)

Entry for 'branch from' instruction

vvvvvv mnem xxxxxxxx ttttt

Entry for 'branch to' instruction

== > vvvvvv mnem xxxxxxxxxxxx

TRANSFER

Privilege Class: G

Use the TRANSFER command to transfer your closed spool files to a specified user or queue, or to reclaim closed spool files that you created.

The format of the class G TRANSFER command is:

TRANSfer	$\left[\begin{array}{l} \text{Printer PRT} \\ \text{PUnch PCH} \\ \text{Reader RDR} \end{array} \right]$	$\left\{ \begin{array}{l} \text{spoolid} \\ \text{Class } c \\ \text{FORM form} \\ \text{ALL} \end{array} \right\}$	$\left(\begin{array}{l} \text{[To]} \left\{ \begin{array}{l} * \\ \text{userid} \end{array} \right\} \\ \text{From} \left\{ \begin{array}{l} \text{ALL} \\ \text{userid} \end{array} \right\} \end{array} \right)$	$\left[\begin{array}{l} \text{Printer PRT} \\ \text{PUnch PCH} \\ \text{Reader RDR} \end{array} \right]$
----------	---	---	---	---

where:

PRINTER/PRT

PUNCH/PCH

READER/RDR

specifies the queues to which and from CP transfers files.

spoolid

is the spool file you want to transfer to the queue of the *userid* you specify or retrieve from the queue of the *userid* specified.

CLASS c

is the class of spool files you want to transfer. The *c* is a one-character alphanumeric field with values from A to Z and from 0 to 9.

FORM form

specifies that you want to transfer all files that have the form name you specify. *form* is a one-to-eight character form name defined at system generation.

ALL

specifies that you want to transfer all spool files in a reader/printer/punch queue.

[TO]{userid}

is the user to whom you want to transfer the files. If you omit the optional keyword TO, the *userid* may not be TO or T. '*' denotes your own userid.

FROM {userid}
ALL

is the *userid* from whom you want to reclaim spool files of the specified reader/printer or punch queue. You can specify ALL to reclaim spool files you originally created and transferred to the reader/printer/punch queues of other users.

TRANSFER

Usage Notes

1. The TRANSFER command does not transfer any active spool files.
2. If you transfer a file to a device that is in system hold status, that file will be placed in system hold. In addition, when you transfer a file that is in system hold status, it will remain in system hold status even if the device to which it is transferred is not in system hold status.
3. The command only transfers spool files to valid queues. For example, files created as real card reader input decks, CP system dumps, user dumps (VMDUMP command), network dumps, or Monitor spool files can exist only on the reader queue. Files created on a virtual punch, or by accounting to disk support, can exist on either the reader queue or the punch queue. Files created on a virtual printer can exist on either the reader queue or the print queue. The following chart summarizes the valid queues to which you can transfer spool files.

	Spool Files	Queue		
		RDR	PRT	PCH
ORIGIN	Real Reader	*		
	Printer	*	*	
	Console	*	*	*
	Punch	*		*
	Dump/ Monitor	*		

“*” = files may reside on queue

CP flags, as an error, any attempt to transfer spool files from a print queue to punch queue, as well as any attempt to transfer spool files from a punch queue to a print queue. You may, however, transfer files on the reader queue to printer or punch queues, if they are script files and punch files respectively. Therefore, you cannot reclaim files (using the FROM option of the TRANSFER command) from the reader queue to another reader queue if they are print and punch files.

Examples:

TRANSFER 0123 TO USERA

transfers the spool file with spoolid 0123 from your reader queue to the reader queue of USERA.

TRANSFER PRT 0123 TO USERA

transfers the spool file with spoolid 0123 from your print queue to the reader queue of USERA.

TRANSFER PRT ALL TO * RDR

transfers all spool files on your print queue to your reader queue. Note that you didn't need to specify the default, RDR.

TRANSFER RDR CLASS B TO * PCH

transfers all the class B punch files on your reader queue to your own punch queue. CP does not transfer any class B files on your reader queue that are not punch files.

TRANSFER RDR ALL FROM ALL PRT

reclaims all print files that you created and sent to the print queues of other users. CP places these reclaimed files on your reader queue.

Responses

$\left. \begin{array}{l} \text{RDR} \\ \text{PRT} \\ \text{PUN} \end{array} \right\}$ FILE spoolid TRANSFERRED TO userid $\left. \begin{array}{l} \text{RDR} \\ \text{PRT} \\ \text{PUN} \end{array} \right\}$

is the response when you issue a TRANSFER command with the TO option. CP also sends this response to the previous owner of a file reclaimed with the FROM option.

where:

RDR/PRT/PUN

(first occurrence) is the queue on which the file previously resided.

RDR/PRT/PUN

(second occurrence) is the queue on which the file now resides. CP does not display the second occurrence of RDR/PRT/PUN unless the old and new queues are different.

spoolid

is the spool identification number of the file. This does not change.

userid

is the user identification of the new file owner.

$\left. \begin{array}{l} \text{RDR} \\ \text{PRT} \\ \text{PUN} \end{array} \right\}$ FILE spoolid TRANSFERRED FROM $\left. \begin{array}{l} \text{userid} \\ * \end{array} \right\}$ $\left. \begin{array}{l} \text{RDR} \\ \text{PRT} \\ \text{PUN} \end{array} \right\}$

is the response to the user issuing the TRANSFER command with the FROM option. This response is also sent to the new owner of a file transferred with the TO option.

where:

RDR/PRT/PUN

(first occurrence) is the queue on which the file now resides.

RDR/PRT/PUN

(second occurrence) is the queue on which the file previously resided. CP does not display the second occurrence of RDR/PRT/PUN unless the old and new queues differ.

TRANSFER

spoolid

is the spool identification number of the file. This does not change.

userid

is the identification of the previous file owner.

is displayed when the "TO *" option is used.

{ nnnn } FILES TRANSFERRED
{ NO }

is also a response you receive when you issue the TRANSFER command. It is not displayed if you issued the CP SET IMSG OFF command line.

CP does not display any of these responses if you issued the CP SET IMSG OFF command.

where:

hexloc1

0

is the first or only hexadecimal virtual storage address dumped. If you omit the *hexloc1* operand, the default is zero, the beginning of virtual machine storage. You may also specify the *hexloc2* or *bytecount* operands with the *hexloc1* operand when you know the dump size you want. If you specify *hexloc2* with *hexloc1*, CP takes a dump of the storage contents within the range of addresses between *hexloc1* and *hexloc2*. For example, when you enter:

```
vmdump 1000-2000
```

or

```
vmdump 1000:2000
```

CP dumps the contents of virtual machine storage at the hexadecimal addresses between 1000 and 2000. If you specify the *bytecount* operand with *hexloc1*, *hexloc1* specifies the beginning address of the dump and the hexadecimal number of bytes dumped. If you enter:

```
vmdump 1000.8A
```

CP dumps the contents of virtual machine storage beginning at hexadecimal address 1000 for a total of 8A bytes. Each time you issue the VMDUMP command you may specify as many address ranges as can be contained on one command line. See usage notes 2 on page 260, 3 on page 260, and 5 on page 260.

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

is the last hexadecimal virtual storage address dumped. If you do not specify the *hexloc2* operand, the default is END, and CP dumps the contents of virtual machine storage starting from *hexloc1* to the end of virtual storage. You must precede the *hexloc2* operand with one of the allowable delimiters (either a dash or a colon) to dump the contents of more than one location by storage address. For example, if you enter:

```
vmdump -:2000
```

or

```
vmdump -2000
```

CP takes a dump of the contents of virtual storage from hexadecimal location zero (the default for *hexloc1*) to hexadecimal address 2000. See usage notes 1 on page 260 and 4 on page 260 for an expanded explanation of the use of delimiters. You may specify up to a maximum of three different virtual machine storage areas you wish to

dump each time you issue the VMDUMP command. You must specify a value for *hexloc2* that is greater than *hexloc1* value. The *hexloc2* value must not exceed the limit of virtual storage.

{ . } [**bytecount**]
[**END**]

is the hexadecimal number of bytes dumped, starting with the first byte of virtual storage at *hexloc1*. If you do not specify *bytecount*, the default is END, and CP takes a dump of the contents of virtual machine storage from the first byte at *hexloc1* to the end of virtual storage. You must specify a period (.) delimiter if you want to dump the contents of more than one storage location by *bytecount*. When you enter:

```
vmdump 1000.20
```

CP takes a dump of the virtual storage contents from hexadecimal location 1000 for a length of X'20' bytes. See usage note 3 on page 260 for an expanded explanation of delimiter use.

{ **TO userid** }
{ **TO *** }

transfers the dump to the virtual card reader of the *userid* specified. If you enter an asterisk after TO, CP transfers the dump to your virtual card reader.

SYSTEM

transfers the dump to the virtual card reader of the *userid* specified on the SYSDUMP operand of the SYSOPR system generation macro instruction. You must not specify TO preceding the keyword SYSTEM.

FORMAT *vmtype*

provides VM/IPCS Extension with the virtual machine type (*vmtype*) which VM/IPCS Extension uses to format the dump. The *vmtype* keyword is a one-to-eight byte name of the operating system running in a virtual machine (for example, CMS). CP also uses the specified *vmtype* as the virtual card reader filetype. CP does not check the *vmtype* for a valid type. Any *vmtype* longer than eight bytes generates an error message and halts further VMDUMP processing. The dump header record includes the *vmtype* you specify and the VM/IPCS Extension uses the *vmtype* information to format the dump. If you enter FORMAT, you must also specify a *vmtype*. If you do not specify FORMAT, the default *vmtype* is FILE. See usage note 4 on page 260.

DSS

specifies that CP take a dump of all discontinuous saved segments in use by your virtual machine. When 'DSS' is specified, only the discontinuous saved segments will be dumped unless you explicitly specify other locations. An error message will be issued if DSS is the only address range specified when DSS does not exist.

***dumpid**

is a line of user input up to 100 characters long including imbedded blanks and asterisks which you can enter for your own benefit (that is, for descriptive purposes, such as the time and date of the dump, or what was being processed at the time of the dump). If you specify this operand, it becomes the DMPDMPID field in the dump file information record (DMPINREC) data area. If specified, you must enter **dumpid* as the last operand on the VMDUMP command line.

Usage Notes

1. First level storage is real main storage. CP, however, dumps only second level storage (that is, storage that VM/SP creates for the guest virtual machine.) Operating systems running in a guest virtual machine such as OS/VS, DOS/VS, and VSE/AF, have virtual (third level) storage of their own. CP cannot dump this third level storage directly. The user or the virtual operating system is responsible for converting any third level storage addresses to second level storage addresses before issuing the VMDUMP command.
2. The dump profile produced contains a whole number of 4K pages. CP rounds down the *hexloc1* operand to a 4K boundary if its value is not a multiple of 4K. CP rounds up the *hexloc2* operand to a 4K boundary minus one.
3. Due to 4K boundary rounding for the *hexloc1* operand, additional dumps of up to 4095 more bytes than you specified can occur.
4. If you enter the FORMAT operand followed by another valid keyword of the VMDUMP command (for example, SYSTEM), CP uses the keyword following FORMAT as the vdtype and CP ignores the normal meaning of the keyword following FORMAT.
5. If you omit a dash or a colon between the *hexloc1* and *hexloc2* operands, and you separate the operands with one or more blanks, CP dumps only the storage contents at those two specific hexadecimal addresses. You must insert blanks to separate operands or sets of operands if you specify more than one operand on the command line. You must not insert blanks to the left or right of the range or length delimiters (-, :, or .) unless you intend to use the default value of the missing operand that the blank represents.
6. If you are running MVS/SP Version 1 Release 3 or a later release in your virtual machine with preferred machine assist active, you may be using storage greater than 16 Mb. However, you cannot issue the VMDUMP command to dump this additional storage. Using the control switch assist feature of preferred machine assist does not alter this.

7. Only non-protected storage is dumped.
8. If you are running VM under VM (second-level), you must issue the VMDUMP command to first-level CP to obtain a dump of the second-level VM system.

Responses**COMMAND COMPLETE**

is the response indicating normal completion of the VMDUMP function.

Appendix A. CP Command Summary

This appendix contains an alphabetical list of the CP commands, the corresponding Function Type for each, the IBM-defined privilege classes which may execute the command, a blank column for recording User-defined classes, and a brief statement about the use 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.
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.
CACHE	<R>	B		Control which devices are eligible for caching and display caching status.
CHANGE	S	D		Alter one or more external attributes of a closed spool file or files.
CHANGE	G	G		Alter one or more attributes of a closed spool file.

Command	Function Type	IBM-Defined Privilege Class	User-Defined Class	Usage
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.
DEFINE	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.
DETACH	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.
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.

Command	Function Type	IBM-Defined Privilege Class	User-Defined Class	Usage
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.
INDICATE	A	E		Display, at the console, the use of and contention for system processor and storage.
INDICATE	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		Bring virtual pages into real storage and lock them; thus, excluding them from future paging.
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.
MESSAGE	<O>	A,B		Send message text to a specified user, to primary system operator, or to one or all logged-on users.

Command	Function Type	IBM-Defined Privilege Class	User-Defined Class	Usage
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.
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.
NETWORK	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). Also control remote 3270 devices via binary synchronous lines.
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.
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.
PURGE	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.
QUERY	R	B		Provide status information on the real or virtual machine and miscellaneous CP functions.
QUERY	P	C		Provide system log messages and information about system users and processors.
QUERY	S	D		Provide system spooling information.

Command	Function Type	IBM-Defined Privilege Class	User-Defined Class	Usage
QUERY	A	E		Provide status information on the real or virtual machine and miscellaneous CP functions.
QUERY	C	F		Provide system log messages and information about system users.
QUERY	G	G		Request information about machine configuration and system status.
QVM	<O>	A		Request the transition from VM 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 created 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.
SET	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.
SET	A	E		Establish system parameters and perform various functions to control the CP system and virtual machine options.

Command	Function Type	IBM-Defined Privilege Class	User-Defined Class	Usage
SET	C	F		Set the recording mode for a device and for soft errors.
SET	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.
SLEEP	N/A	Any		Places the virtual machine in a dormant state with the keyboard locked.
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.
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.
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.
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.

Command	Function Type	IBM-Defined Privilege Class	User-Defined Class	Usage
TRANSFER	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.

Appendix B. Summary of CP Commands

This section is provided as a quick-reference summary of the VM/SP HPO CP commands. Additional information concerning the commands with privilege classes A through F can be obtained from *VM/SP HPO Operator's Guide*. Additional information concerning the commands with privilege class G or ANY can be found earlier in this book.

Be aware that all commands are not available to every virtual machine user. Rather, the commands belong to different classes that represent different areas of computer activity. The class of commands that a user can issue is contained with the user's identification code in the system directory. The system directory is controlled by the system operations personnel.

The CP commands can be arranged in five categories:

- Program analysis and monitor functions
- Device control functions
- Virtual console functions
- Spooling and file control functions
- Miscellaneous functions (those commands that do not belong in any of the above categories)

Many command names and operands can be truncated. Where truncation is permitted, the shortest acceptable version of the command is represented by uppercase letters, with the optional part represented by lowercase letters. (Remember, however, that any command can be entered with any mixture of uppercase and lowercase letters.)

Program Analysis and Monitor Functions

Command	Explanation
ACNT	Render accounting information for and to the virtual user (Class A).
ADSTOP	Specify an instruction address location where the user's program is to be halted (Class G).
CPTrap	Create a READER file of selected trace table, CP interface, and virtual machine interface entries for problem determination (Class C).
DCP	Display real processor storage locations (Class E).
Display	Display virtual console registers, plus the current CAW, CSW, and PSW (Class G).

DMCP	Dump the real storage locations to a user's virtual printer (Class E).
DUmp	Dump virtual machine registers and storage to the virtual printer (Class G).
LOCate	Provide the starting location of the CP control blocks associated with a user or his virtual (or real) devices (Class E).
MONitor	Initiate or terminate the recording of events that occur in the real machine (Classes A and E).
PER	Monitor certain events as they occur during program execution in the user's virtual machine (Class A, B, C, D, E, F, and G).
SAVESYS	Provide a storage copy of virtual machine storage, registers, and PSW contents as they currently exist (Class E).
STCP	Change the contents of real processor storage (Class C).
TRace	Trace virtual machine activity at the terminal, spooled printer, or both (Class G).
VMDUMP	Dumps the individual virtual machine's entire storage area. This command must be processed by the VM/IPCS Extension program product or a user written program (Class G).

Device Control Functions

Command	Explanation
ATTach	Attach a real device for system control (Class B).
CACHE	Control which devices are eligible for caching and display caching status (Class B).
COUPLE	Connect virtual channel to channel adapters (Class G).
DEFine	Alter the device feature of a 3330V (Class B).
DETach	Remove a real or virtual device or channel from a virtual machine or the real system (Class B and G).
DISAble	Disable direct or switched communication lines from the system (Class A or B).
DISConn	Disconnect the user's terminal from his virtual machine (Class Any).

ENable	Connect specified communication lines to the system (Class B).
Link	Provide access to a specified DASD device by a user's virtual machine (Class G).
LOADBUF	Load the 1403 universal character set (UCS) buffer with a specified train image. This command also has options that control the 3203, 3211, 3262, 3289, and 4245 printers (Class D).
LOADVFCB	Load the virtual forms control buffer for a virtual spooled 3203, 3211, 3262, 3289, and 4245 printers (Class G).
NETwork	Allow the enabling and disabling of all or specific 3704/3705 resources. In addition, the Network command is used to vary resources offline or online (Class A and B).
VARY	Allow or disallow the availability of a device or a processor to a virtual machine or the control program (Class B).

Virtual Console Functions

Command	Explanation
BEGIN	Continue or resume execution of the virtual machine at a specified storage location or at the address specified in the current PSW (Class G).
EXTERNAL	Simulate an external interruption condition for a virtual machine and return control to that machine (Class G).
IPL	Initiate a program load for a virtual machine (Class G).
RESET	Clear all pending interruptions on a specified device (Class G).
SCREEN	Change or alter the color and extended highlighting values for the virtual console (Class G).
SPOOL	Allow the starting or stopping of virtual console spooling as well as the alteration of other spooling control options (Class G).
STORE	Alter registers and storage of a virtual machine (Class G).
SYSTEM	Initiate the virtual console functions of clear, System Reset, and PSW Restart (Class G).
TERMINAL	Redefine the user's terminal input characteristics and attention interruption handling capabilities (Class G).

Spooling and File Control Functions

Command	Explanation
BAckspac	Restart or backspace the output of a unit record spooling device (Class D).
CHange	Alter one or more attributes of a closed spooled file (Class D and G).
CLose	Terminate a spooling operation on a virtual console, reader, printer, or punch (Class G).
DRain	Stop spooling activity on the specified device after the current file in operation reaches termination (Class D).
Flush	Eliminate and halt the current file output on the specified real unit record device (Class D).
FRee	Release spool HOLD status from previously held files for real RDR, PUN, and PTR devices (Class D).
HOLD	Defer file output to the real RDR, PTR, and PUN devices (Class D).
ORDer	Redefine the order of closed spool files (Class D and G).
PUrge	Remove spooled file(s) before reading, punching, or printing (Class D and G).
REPeat	Add to the number of copies of an output printer or punch file on a real unit-record device (Class D).
SPACE	Force single space on the printer, regardless of carriage control codes contained in the file (Class D).
SPool	Change spooling control options on a given device or group of devices and also starting or stopping of console spooling (Class G).
SPTape	Dump spool file on tape or load spool file from tape (Class D).
STart	Start a unit record device, or restart a drained device, or restart and change output classes that it may service (Class D).
TAg	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 (Class G).
TRANsfer	Direct an input spool file to or retrieve it from a specified user's virtual card reader (Class G).

Miscellaneous Functions

Command	Explanation
*	Provide a means of inserting commentary in the stream of virtual console input and output messages (Class Any).
#CP	Allow CP command execution while remaining in the virtual machine environment without having to signal attention (Class Any).
ATTN	Create an interruption pending from the virtual console (Class G).
AUTOLOG	Provides a means to automatically log on and load any virtual machine with IPL defined in its directory (Class A or B).
COMMANDS	Display the commands and diagnose codes you are authorized to use (Class Any).
CP	Allow the user to key in CP commands following "CP" while in the CMS or virtual machine environment (Class Any).
DIAL	Logically attach a user terminal to a multi-access virtual system (Class Any).
DEFine	Reconfigure one's own virtual machine (Class G).
ECho	A terminal test facility that allows keyboard entered data to be displayed back at the user's terminal (Class G).
FORCE	Force the logoff of any virtual machine (Class A).
HALT	Stop any active channel program on a specified device (Class A).
INDicate	Displays use of and contention for major system resources (Class E and G).
LOCK	Lock a user's pages in processor storage (Class A).
LOGoff LOGout	Terminate user activity on a virtual machine (Class Any).
Logon Login	Gain access to a virtual system from a user terminal (Class Any).
Message	Send a specified message from one virtual machine to another (Class A, B, or Any).
MIGrate	Activates the normal page/swap table migration routines (Class A).

MONitor	Records system counters needed for performance measurement (Class A and E).
MSGNOH	Send a specified message, without the standard MESSAGE header, from one virtual machine to another (Class A and B).
NETwork	Provide loading, dumping, tracing, and other functions for the 3704/3705 Communications Controller (Class A and B).
NOTReady	Simulate the not-ready status on a virtual spooled unit record or virtual console device (Class G).
Query	Provide status information on the real or virtual machine, and miscellaneous CP functions (Class A, B, C, D, E, F, and G).
QVM	Request the transition from the VM/SP HPO environment to a particular virtual machine, running in native mode (Class A).
Ready	Simulate a ready (device end interruption) condition for a specified virtual device (Class G).
REQuest	Makes an interruption pending from the virtual console (Class G).
REWind	Rewind a tape to the load point on a real device attached to a user's virtual machine (Class G).
SEND	Pass commands and message replies to a disconnected virtual machine for processing (Class G).
Set	Establish system parameters for virtual and real machines, as well as other VM/SP HPO values (Class A, B, E, F, and G).
SHUTDOWN	Terminate VM/SP HPO activity in an orderly manner and checkpoint the system (Class A).
SLeep	Place the virtual machine in a dormant state, with the keyboard locked (Class Any).
SMsg	Send special messages to a specified virtual machine (Class G).
SPMode	Used to establish or reset to the single processor mode environment (Class A).
UNLOCK	Release previously locked page frames of real storage (Class A).
Warning	Transmit high priority messages or warnings to a user or all users (Class A or B).

Index

Special Characters

- * command
 - described 52
- * operand
 - of MESSAGE command 128
 - of SPOOL command 218
 - of VMDUMP command 259
- *dumpid
 - explained 48
- *dumpid operand
 - of DUMP command 98
 - of VMDUMP command 260
- #CP command
 - described 53
 - example of usage 53
 - operand
 - commandline 53
 - responses 54
 - usage notes 53-54
- {
 - use of 50

A

- abbreviation
 - use of 47
- abend
 - spooling considerations 226
- access mode password 113-118
- access modes 113-118
- accounting information
 - displaying 194-211
- ACNT operand
 - of SET command 195
- adding dedicated channels 23
- adding dedicated devices 21-22
- ADSTOP
 - turning off 30
- ADSTOP command
 - described 55
 - operands
 - hexloc 55
 - OFF 55
 - preparing to use 197
 - responses 56
 - syntax diagram 55
 - usage notes 55-56
- AFFINITY operand
 - of SET command 200
- affinity setting
 - for AP/MP installations 194-211
 - relation to the logon process 125-127

- ALL operand
 - of CHANGE command 60
 - of PURGE command 155
 - of QUERY command 160
 - of QUERY VIRTUAL command 159
 - of SCREEN command 189
 - of TRACE command 247
 - of TRANSFER command 253
- altering
 - logical editing symbols 237-245
 - spool file attributes 17
 - spooled device attributes 17
 - virtual machine environment 72
 - virtual printer file attributes 59-62
 - virtual punch file attributes 59-62
 - virtual reader file attributes 59-62
 - virtual storage 33-35
- annotating your console sheet 52
- Any user
 - described 9
- APL
 - control of translation tables 237-245
- APL operand
 - of TERMINAL command 240
- area operand
 - of SCREEN command 189
- ASSIST operand
 - of SET command 201-203
- attached processor
 - displaying load 103-106
- attached processor application
 - use of SET ASSIST 201-202
- attaching dedicated channels 23
- attention handling
 - terminal control of 237-245
- attention interruption
 - pending at virtual console 186
 - signalling 237-245
 - simulating 57, 107-112
- ATTN command
 - described 57
 - syntax diagram 57
 - usage notes 57
- ATTN operand
 - of IPL command 108
 - of TERMINAL command 241
- attributes
 - changing spool file 59-62
 - modifying for 3800 printer 216-226
 - modifying virtual device 216-226
- AUTOOCR option
 - of PARM operand
 - for CMS users 112
- automatic carriage return
 - option on IPL command 107-112
- automatic loading
 - suppressing 24

AUTOPOLL operand
of SET command 195

B

BATCH option
of PARM operand
for CMS users 111

BEGIN command
described 58
operand
hexloc 58
syntax diagram 58
usage notes 58
used after disconnecting 89

blanks
as delimiters 6

block address
used to load a system 107-112

BMX operand
of DEFINE command 76

BOTH operand
of TRACE command 248

BOTH option
of PER command 140

braces
use of 50

brackets
use of 50

BRANCH event type
of PER command 135

BRANCH operand
of TRACE command 247

branches
tracing 246-252

BREAKIN operand
of TERMINAL command 242

BRKKEY operand
of TERMINAL command 243

byte alignment
on terminal output 31

bytecount operand
of VMDUMP command 259

C

CAW (Channel Address Word)
displaying 91-96

CAW operand
of DISPLAY command 94

CCW operand
of TRACE command 247

CCW testing
controlling 194-211

CHANGE command

described 59

operands
ALL 60
CHARS 61
CLASS 59, 60
COPY[*]nnn 60
DIST distcode 60
FCB 61
FLASH name 60
FORM form 60, 61
HOLD 60
MODIFY named 60
NAME 61
NOHOLD 60
PRINTER 59
PUNCH 59
READER 59
spoolid 59

responses 62
syntax diagram 59
usage notes 61

changing
control registers 227-230
filename of spool file 59-62
floating-point registers 227-230
general registers 227-230
Program Status Word (PSW) 227-230
spool file attributes 59-62
spool file characteristics 17
spool file class 59-62
spoolid 59-62
virtual storage locations 227-230

Channel Address Word (CAW)
displaying 91-96

CHANNEL c operand
of DETACH command 83

channel operating mode
determining current setting 159, 166

Channel Status Word (CSW)
displaying 91-96

channel-to-channel adapters
connecting virtual 68

CHANNELS operand
of DEFINE command 76
of QUERY VIRTUAL command 159

character sets
controlling
APL 237-245
TEXT 237-245
CP command language 7

character translation tables
controlling 237-245

CHARDEL operand
of TERMINAL command 238

CHARS name operand
of SPOOL command 221

CHARS operand
of CHANGE command 61

class
changing for spool file 59-62

- modifying virtual device spool 216-226
 - privilege, for CP commands 9
- Class Any
 - described 12
- class G
 - described 11
- CLASS operand
 - of CHANGE command 59
 - of ORDER command 131
 - of PURGE command 155
 - of QUERY command 161
 - of SPOOL command 217
 - of TRANSFER command 253
- CLEAR operand
 - of IPL command 107
 - of SYSTEM command 231
- clearing
 - interruptions from a virtual machine 187
 - storage to zeros 107-112, 231
 - virtual storage 24
- CLOSE command
 - described 63
 - operands
 - CONSOLE 63
 - DIST distcode 64
 - FORM form 64
 - HOLD 63
 - NAME fn [ft]/dsname 64
 - NOHOLD 64
 - PRINTER 63
 - PUNCH 63
 - PURGE 64
 - READER 63
 - vaddr 63
 - responses 66
 - syntax diagram 63
 - usage notes 65
- CLOSE operand
 - of SPOOL command 220
- closing
 - files 63
 - spool files 18
 - TRACE output 36
 - virtual devices 220
- CMD option
 - of PER Command 140
- CMS users
 - AUTOOCR option 107-112
 - BATCH option 107-112
- code operand
 - of EXTERNAL command 102
- color
 - determining values in effect 162, 175
- command environment
 - CP 5
 - defined 5
 - virtual machine 5
- command name 6
- command operands 6
- commandline operand
 - of #CP command 53
 - of CP command 70
- commands
 - CP
 - miscellaneous 275
 - summarized 263
 - summary 271-275
 - used for analysis and monitor functions 271
 - used for device control 272
 - used for spooling and file control 274
 - used for virtual console 273
 - CP summarized 269
 - finding those authorized for user 67
 - notational conventions 47, 50
 - passing 192
 - summary of use 263-269
 - used to control a terminal session
 - summary of 263-269
- COMMANDS command
 - described 67
 - responses 67
 - syntax diagram 67
 - usage notes 67
- comments
 - entering 52
 - how to write 6
- CONMODE operand
 - of TERMINAL command 242
- connecting
 - remote terminals 86-87
 - virtual channel-to-channel adapters 68
- console
 - controlling spooling 217
 - disconnecting from virtual machine 89-90
 - modifying spool file 217
 - terminating input/output 217
- CONSOLE operand
 - of CLOSE command 63
 - of DEFINE command 73
 - of QUERY VIRTUAL command 159
 - of SPOOL command 217
 - of TAG command 233
- console sheet
 - annotating 52
- CONT operand
 - of SPOOL command 217
- continuous reading of file 216-226
- control functions
 - setting 194-211
- Control Program console function mode 28
- control registers
 - changing 227-230
- control switch assist
 - of preferred machine assist 56, 80, 109, 117, 209, 210
- controlling
 - I/O functions 14
 - system in a virtual machine 13
 - virtual machine 23-28
 - 3270 display terminal 237-245

- controlling a terminal session
 - summary of commands used 263-269
- COPY[*]nnn operand
 - of CHANGE command 60
 - of SPOOL command 219
- COUNT sub-command
 - of PER command 142
- COUPLE command
 - described 68
 - operands
 - userid 68
 - vaddr 68
 - responses 68-69
 - syntax diagram 68
 - usage notes 68
- CP command
 - described 70-71
 - environment 5
 - format 6
 - operand
 - commandline 70
 - responses 71
 - structure 6
 - syntax diagram 70
 - usage notes 70
- CP commands
 - entering from a virtual machine
 - environment 53
 - entering from CP console function mode 28
 - entering from the virtual machine
 - environment 70
 - entering while running 26
 - PER 272
 - privilege class for 9
 - summarized 263, 269
 - summary 271-275
 - text format of 51
- CP operand
 - of SEND command 192
- CPCONIO operand
 - of SET command 199
- CPOUT operand
 - of SCREEN command 190
- CPUID operand
 - of QUERY command 160
 - of SET command 201
- creating disk space
 - precautions 72-82
- Cross Memory Assist feature
 - enabling 164, 211
- CSW (Channel Status Word)
 - displaying 91-96
- CSW operand
 - of DISPLAY command 94
 - of TRACE command 247
- CTCA operand
 - of DEFINE command 73
- cuu
 - explained 48
- cylno operand

of IPL command 107

D

- DASD
 - listing users currently linked to the same 159, 170
- DASD operand
 - of QUERY VIRTUAL command 159
- DAT option
 - of PER command 142, 144
- DATOFF option
 - of PER command 142, 144
- debugging programs 29
- dedicated channels
 - attaching 23
 - detaching 23, 83
- dedicated devices
 - attaching 21-22
 - detaching 21-22
- dedicated 2305 restriction 72-82
- DEFINE command
 - described 72-82
 - operands
 - BMX 76
 - CHANNELS 76
 - CONSOLE 73
 - CTCA 73
 - GRAF 76
 - LINE 76
 - PRINTER 73
 - PUNCH 73
 - READER 73
 - SEL 76
 - STORAGE 78
 - TFB-512 78
 - TIMER 74
 - T2305 78
 - T2314 78
 - T2319 78
 - T3310 78
 - T3330 78
 - T3340 78
 - T3350 78
 - T3370 78
 - T3375 78
 - T3380 78
 - vaddr 77
 - 1403 74
 - 1443 74
 - 2501 74
 - 2540P 74
 - 2540R 74
 - 3036 76
 - 3066 76
 - 3088 74
 - 3138 76

- 3148 76
- 3158 76
- 3203 74
- 3211 74
- 3262 74
- 3270 76
- 3289E 74
- 3505 75
- 3525 75
- 3800 75
- 4245 76
- 4248 76
- responses 82
- syntax diagram 72
- usage notes 79-81
- defining
 - fixed-block mode devices 72-82
 - temporary disks 16
 - virtual devices 28, 72-82
- deleting spool files 21
- delimiters
 - in command lines 6
- DETACH command
 - described 83
 - operand
 - vaddr 83
 - responses 84
 - syntax diagram 83
 - usage notes 83
- detaching
 - dedicated channels 83
 - dedicated devices 21-22
 - devices 84
 - temporary disks 16
 - virtual devices 83-85
- determining
 - current command environment 5
- DEV operand
 - of TAG command 233, 234
- device
 - control
 - CP command used 272
- devices
 - simulating not ready status for virtual 130
 - virtual
 - linking to 113-118
- diagnose codes
 - finding those authorized for user 67
- DIAL command
 - described 86
 - device restriction 86-87
 - operands
 - userid 86
 - vaddr 86
 - responses 87
 - syntax diagram 86
 - usage notes 86-87
- dialing in 86-87
- directory entry
 - virtual machine attributes 14
- DISCONN command
 - described 89
 - operand
 - HOLD 89
 - responses 90
 - syntax diagram 89
 - usage notes 89
- disconnecting
 - virtual console 89-90
 - virtual machine 122-124
 - logging off your virtual machine 122-124
- disks
 - virtual permanent 15
 - virtual temporary 16
- DISPLAY command
 - described 91
 - operands
 - CAW 94
 - CSW 94
 - END 92
 - Greg 93
 - hexloc 92
 - PSW 93
 - regcount 93
 - Xreg 93
 - Yreg 93
 - responses 95-96
 - syntax diagram 91
 - usage notes 94-95
- displaying
 - accounting information 194-211
 - Channel Address Word (CAW) 91-96
 - Channel Status Word (CSW) 91-96
 - extended control registers 91-96
 - floating-point registers 91-96
 - general registers 91-96
 - Program Status Word (PSW) 91-96
 - storage keys 91-96
 - system load 103-106
 - virtual storage 30
 - virtual storage locations 91-96
- DIST distcode operand
 - of CHANGE command 60
 - of CLOSE command 64
 - of SPOOL command 222
- distribution code
 - for output files 59-62
- dsname
 - explained 48
- DSS operand
 - of VMDUMP command 259
- DUMP command
 - described 97
 - operands
 - *dumpid 98
 - END 97
 - hexloc 97
 - responses 100
 - syntax diagram 97
 - usage notes 98

- dumping
 - extended control registers 97-100
 - floating-point registers 97-100
 - general registers 97-100
 - guest virtual storage 257-261
 - storage locations 97-100
 - virtual storage 32-33

E

- ECHO command
 - described 101
 - operands
 - nn 101
 - 1 101
 - responses 101
 - syntax diagram 101
 - usage notes 101
- ECMODE operand
 - of SET command 197
- editing symbols
 - controlling use of 194-211
- ellipsis
 - use of 50
- EMSG operand
 - of SET command 199
- END operand
 - of DISPLAY command 92
 - of DUMP command 97
 - of TRACE command 247
 - of VMDUMP command 258
- END sub-command
 - of PER command 144
- ending a TRACE 36
- entering
 - comments 52
 - CP commands 5, 26
 - from virtual machine environment 70
- EOF operand
 - of SPOOL command 218
- error message handling
 - controlling 194-211
- ESCAPE operand
 - of TERMINAL command 239
- execution
 - halting at instruction address 55-56
 - resuming virtual machine 58
 - stopping virtual machine 29
- extcolor operand
 - of SCREEN command 190
- extended control mode
 - setting 194-211
- extended control registers
 - displaying 91-96
 - printing 97-100
- EXTERNAL command
 - described 102

- operands
 - code 102
 - 40 102
 - response 102
 - syntax diagram 102
 - usage notes 102
- external interruption
 - simulated 102
- EXTERNAL operand
 - of TRACE command 246
- exthilght operand
 - of SCREEN command 190

F

- FB-512 devices
 - determining information about 157-184
- FCB operand
 - of CHANGE command 61
 - of LOADVFCB command 119
 - of SPOOL command 221
- FILE operand
 - of TAG command 234
- FILE spoolid operand
 - of TAG command 233
- files
 - changing spool class of 59-62
 - changing spool filename for 59-62
 - changing spoolid 59-62
 - closing 63
 - distribution codes for 59-62
 - holding 216-226
 - holding before output processing 59-62
 - holding from further processing 63-226
 - purging 63-226
 - reclaiming 253-256
 - reordering closed spool files 131-132
 - terminating processing 63
 - transferring to another user 216-226
 - transmitting 253-256
- FILES operand
 - of QUERY command 159
- first-level storage
 - requesting 91-96
- FLASH name operand
 - of CHANGE command 60
 - of SPOOL command 220
- floating-point registers
 - changing 227
 - displaying 91-96
 - printing 97-100
- fn
 - explained 48
- following virtual machine activity 35-36
- FOR operand
 - of SPOOL command 219
- FORM form operand

- of CHANGE command 60, 61
- of CLOSE command 64
- of ORDER command 131
- of PURGE command 155
- of QUERY command 161
- of SPOOL command 222
- of TRANSFER command 253
- FORMAT vmtyp operand
 - of VMDUMP command 259
- forms controls for a virtual printer
 - specifying 119-121
- FROM operand
 - of TRANSFER command 253
- FROM option
 - of PER command 140
- ft
 - explained 48

G

- G[REG1] event-type
 - of PER command 137
- general registers
 - changing 227-230
 - displaying 91-96
 - printing 97-100
- general user
 - described 11
- GET sub-command
 - of PER command 143
- GRAF operand
 - of DEFINE command 76
 - of QUERY VIRTUAL command 159
- Greg operand
 - of DISPLAY command 93
 - of STORE command 228
- Group Control System (GCS)
 - with ADSTOP command 56
 - with DISCONN command 89
 - with LOGOFF command 122
 - with QUERY command 168
 - with TRACE command 249
- GUESTR option
 - of PER command 43, 142
- GUESTV option
 - of PER command 43, 142

H

- halting execution
 - at instruction address 55-56
 - of virtual machine 29
- handshaking
 - control of VM/VS 194-211
- hardware assists 194-211

- hexloc
 - explained 48
- hexloc operand
 - of ADSTOP command 55
 - of BEGIN command 58
 - of DISPLAY command 92
 - of DUMP command 97
 - of STORE command 227
 - of VMDUMP command 258
- hiding your password 125-127
- highlighting
 - 3270 terminal display 237-245
- HILIGHT operand
 - of TERMINAL command 241
- HOLD operand
 - of CHANGE command 60
 - of CLOSE command 63
 - of DISCONN command 89
 - of LOGOFF command 122
 - of QUERY command 161
 - of SPOOL command 218
- holding
 - file before output processing 59-62
 - files by user hold status 216-226
 - files from further processing 63-226

I

- I/O devices
 - defining for virtual machine 72-82
- I/O operand
 - of TRACE command 246
- IMSG operand
 - of SET command 196
- INAREA operand
 - of SCREEN command 189
- INDEX nn operand
 - for LOADVFCB command 119
- Indexed Sequential Access Method (ISAM) 194-211
- INDICATE command
 - described 103
 - operands
 - LOAD 103
 - USER 103
 - responses 103-106
 - syntax diagram 103
 - usage notes 103
- INDICATE LOAD command
 - responses 103-106
- INDICATE USER command
 - responses 105-106
- informational messages
 - controlling receipt 194-211
- initial program load (IPL)
 - of an operating system 107-112
 - stopping 107-112
 - suppressing automatic 125-127

- input/output operations
 - tracing 246-252
- INREDISP operand
 - of SCREEN command 190
- INSTRUCT event type
 - of PER command 135
- INSTRUCT operand
 - of TRACE command 247
- internal print file duplication
 - 3800 printer 59-62
- interruptions 186
 - clearing from a virtual machine 187
 - range of external codes 102
 - simulating attention 57
 - simulating external 102
 - specifying virtual device-end 185
 - tracing 246-252
- IPL command
 - described 107-112
 - for named systems 107-112
 - operands
 - ATTN 108
 - CLEAR 107
 - cylno 107
 - nnnnnn 107
 - NOCLEAR 107
 - PARM 109
 - PMA 108
 - PMAY 109
 - STOP 108
 - systemname 109
 - vaddr 107
 - responses 112
 - syntax diagram 107
 - usage notes 110-112
- ISAM (Indexed Sequential Access Method) 194
- ISAM operand
 - of SET command 198

K

- keys
 - controlling program function 194-211
 - printing storage 97-100
 - storage
 - displaying 91-96

L

- line editing symbols
 - controlling 194-211
- line length
 - controlling at your terminal 237-245
- LINE operand
 - of DEFINE command 76

- LINEDEL operand
 - of TERMINAL command 239
- LINEDIT operand
 - of SET command 197
- LINEND operand
 - of TERMINAL command 239
- LINES operand
 - of QUERY VIRTUAL command 159
- LINESIZE operand
 - of TERMINAL command 242
- LINK command
 - access mode passwords 113-118
 - batch job restrictions 115
 - described 113
 - operands
 - mode 113
 - PASS= password 114
 - userid 113
 - vaddr 113
 - responses 117-118
 - syntax diagram 113
 - usage notes 115-117
 - 2305 restrictions 115
- linking
 - a device to your virtual machine 113-118
 - to virtual disks 16
- links
 - listing users with current 157-184
- LINKS operand
 - of QUERY command 160
- load key
 - simulation of 107-112
- LOAD operand
 - of INDICATE command 103
- loading
 - an operating system 23
 - by named system 24
 - suppressing automatic 24
 - system by block address 107-112
- LOADVFCB command
 - described 119
 - operands
 - FCB 119
 - INDEX nn 119
 - name 119
 - vaddr 119
 - syntax diagram 119
 - usage notes 119-121
- logging on
 - a virtual machine 24, 125-127
- logical editing symbols
 - altering 237-245
 - controlling use of 194-211
- logically connected terminal 86-87
- LOGOFF command
 - described 122
 - operand
 - HOLD 122
 - responses 123-124
 - syntax diagram 122

- usage notes 122
- logon
 - affinity setting effect 126
- LOGON command
 - described 125
 - operands
 - NOIPL 125
 - password 125
 - userid 125
 - responses 127
 - syntax diagram 125
 - usage notes 125-126
- logon procedure
 - hiding your password during 125-127
- lowercase letters
 - words and symbols 47

M

- MASK event-type
 - of PER command 137
- masking password
 - during logon 125-127
 - using TERMINAL command 245
- MAXDEV xxxx option
 - of OPTION control statement 79, 84, 115
- MESSAGE command
 - described 128-129
 - operands
 - msgtext 128
 - OPERATOR 129
 - userid 128
 - response 129
 - usage notes 128-129
- messages
 - controlling informational 194-211
 - controlling transmission of 194-211
 - handling error 194-211
 - receiving special 194-211
 - receiving warning 194-211
 - sending to other users 128-129
 - sending to the operator 128-129
- MODE operand
 - of LINK command 113
 - of TERMINAL command 241
- MODIFY name operand
 - of CHANGE command 60
 - of SPOOL command 220
- modifying
 - spooling control options 216-226
 - storage 55-56
 - virtual device attributes 216-226
- monitor functions
 - CP commands used 271
- MSG operand
 - of SET command 196
- msgtext
 - explained 48

- msgtext operand
 - for MESSAGE command 128
 - of SMSG command 214
- MSS
 - defining devices 72-82
 - defining virtual volumes 72-82
- multiple shadow tables
 - enabling 194-211
- multiprocessor
 - displaying load 103-106

N

- n
 - explained 48
- NAME operand
 - of CHANGE command 61
 - of CLOSE command 64
 - of LOADVFCB command 119
- named system
 - loading by 24, 107-112
- NAMES operand
 - of QUERY command 187
- nesting
 - described 50
- nn operand
 - of ECHO command 101
- nn SEC/MIN/HRs operand
 - SLEEP command 212
- nnnnnn operand
 - of IPL command 107
- NOCLEAR operand
 - of IPL command 107
- NOCONT operand
 - of SPOOL command 217
- NOEOF operand
 - of SPOOL command 218
- NOHOLD
 - of CHANGE command 60
- NOHOLD operand
 - of CLOSE command 64
 - of QUERY command 161
 - of SPOOL command 218
- NOIPL operand
 - of LOGON operand 125
- NORUN operand
 - of TRACE command 248
- NORUN option
 - of PER command 140
- not ready status
 - simulating 130
- notational conventions 47-50
- NOTERM operand
 - of SPOOL command 221
- NOTRANS operand
 - of SET command 198
- NOTREADY command

described 130
operand
 vaddr 130
syntax diagram 130
usage notes 130

O

OFF operand
 of ADSTOP command 55
 of SET command 195, 196, 197, 198, 199, 200
 of SPOOL command 219
 of TRACE command 248
operands 6
operating systems
 initial program load 107-112
 loading 23
 passing parameters at IPL 24
 passing parameters to 107-112
operator
 sending messages to 128-129
OPERATOR operand
 of MESSAGE command 129
ORDER command
 described 131
 operands
 CLASS 131
 FORM form 131
 PRINTER 131
 PUNCH 131
 READER 131
 spoolid 131
 responses 132
 syntax diagram 131
 usage notes 131-132
OS Indexed Sequential Access Method
(ISAM) 194-211
OUTAREA operand
 of SCREEN command 189

P

PAGEX operand
 of SET command 198
parameters
 passing to virtual operating system 107-112
PARM operand
 of IPL command 109
PASS option
 of PER Command 140
PASS = password operand
 of LINBK command 114
passing parameters to virtual operating
 system 107-112
password

 access mode 113-118
 hiding during logon procedure 125-127
 invalid format on commandline
 precaution 125-127
 masking 245
password operand
 of LOGON operand 125
PCH operand
 of TAG command 233
PER
 command (CP) 272
PER command
 class 133
 CMD 42
 COUNT 40
 debugging 39
 event types 137
 BRANCH 134, 135, 144
 G[REG1] 134, 137
 INSTRUCT 134, 135, 146
 MASK 134
 STORE 134, 136, 144, 146
 format 133
 monitoring events 133
 options
 APPEND 146
 BOTH 134, 138, 140
 CMD 134, 138, 140, 144
 DAT 140, 142, 144
 DAT option 138
 DATOFF 140, 142, 144
 DATOFF option 138
 FROM 134, 138, 140
 GUESTR 43, 138, 140, 142, 144
 GUESTV 43, 138, 140, 142, 144
 NORUN 134, 138, 140, 144
 PASS 134, 138, 140, 144
 PRINTER 134, 138, 140, 144
 RANGE 134, 138, 139, 140
 RUN 134, 138, 140, 144
 STEP 134, 138, 140, 144
 TERMINAL 134, 138, 140, 144
 responses 152-153
 selectivity 37
 sequence indicator 154
 storage alteration 43
 sub-commands
 APPEND 143
 COUNT 134, 142
 END 134
 GET 134, 146
 SAVE 134, 146
 TABLE 134
 suspending 39
 syntax table 134
 table 142
 terminating 38
 trace elements
 changing options 145
 changing specific trace elements 145

- merging 144
- traceback table 39, 142
- traceset 142, 146
- usage notes 147-152
- using 36
- permanent virtual disks 15
- PF keys
 - controlling function 194-211
- PFnn operand
 - of QUERY command 162
 - of SET command 203-208
- PMA operand
 - of IPL command 108
- PMAV
 - of IPL command 109
- power off vs. logoff
 - with 3270 terminal 122-124
- preferred machine assist
 - control switch assist 56, 80, 109, 117, 209, 210
 - enabling 107-112
 - resetting virtual machine 76
 - restrictions 107-112
 - turning off 107-112
- primary system operator
 - described 9
- print file requirements for 3800 printer 62
- printer
 - specifying forms controls for virtual 119-121
- printer files
 - altering attributes of virtual 59-62
- PRINTER operand
 - of CHANGE command 59
 - of CLOSE command 63
 - of DEFINE command 73
 - of ORDER command 131
 - of PURGE command 155
 - of QUERY command 160
 - of SPOOL command 216
 - of TRACE command 247
 - of TRANSFER command 253
- PRINTER option
 - of PER command 140
- printing
 - contents of virtual machine components 97-100
 - extended control registers 97-100
 - first level storage 98
 - floating-point registers 97-100
 - general registers 97-100
 - Program Status Word (PSW) 97-100
 - storage locations 97-100
 - virtual storage 32-33
 - virtual storage keys 97-100
- PRIV operand
 - of TRACE command 246
- privilege class
 - described 9
- privilege classes
 - for CP commands
 - defined 9
- privileged instructions
 - tracing 246-252
- problem analysis
 - synopsis of CP commands used 271
- PROCESS operand
 - of QUERY command 162
- processing special messages 214-215
- processor
 - determining attached or main 157-184
 - determining operating mode 157-184
 - identifying 157-184
 - setting identification 194-211
- program function keys
 - controlling 194-211
- program load
 - initial 107-112
- PROGRAM operand
 - of TRACE command 246
- Program Status Word (PSW)
 - changing 227-230
 - displaying 91-96
 - printing 97-100
- PRT operand
 - of TAG command 233
- pseudo page fault processing
 - VM/VS handshaking 194-211
- PSW (Program Status Word)
 - displaying 91-96
 - printing 97-100
- PSW operand
 - of DISPLAY command 93
 - of STORE command 228
- punch files
 - altering attributes of virtual 59-62
- PUNCH operand
 - of CHANGE command 59
 - of CLOSE command 63
 - of DEFINE command 73
 - of ORDER command 131
 - of PURGE command 155
 - of QUERY command 160
 - of SPOOL command 217
 - of TRANSFER command 253
- PURGE command
 - described 155
 - operands
 - ALL 155
 - CLASS 155
 - FORM form 155
 - PRINTER 155
 - PUNCH 155
 - READER 155
 - spoolid 155
 - responses 156
 - syntax diagram 155
 - usage notes 155-156
- PURGE operand
 - of CLOSE command 64
 - of SPOOL command 220
- purging
 - closed files 63

closed spool files 155-156
spool files 21, 216-226



QUERY command

Class G

described 157

operands

ALL 159, 160
CLASS 161
CPLEVEL 162
CPUID 160
FILES 159
FORM form 161
HOLD 161
LINKS 160
NAMES 187
NOHOLD 161
PFnn 162
PRINTER 160
PROCESS 162
PUNCH 160
READER 160
SCREEN 162
SECUSER 160
SET 159
SPMODE 162
spoolid 161
SYSHOLD 161
S370E 162
TBL 161
TERMINAL 159
TIME 159
USERHOLD 161
VIRTUAL 159
VMSAVE 162

responses 163-184
syntax diagram 158

QUERY CPLEVEL command

responses 181

QUERY CPUID command

response 165-166

QUERY FILES command

response 166

QUERY LINKS command

responses 174

QUERY LOGMSG command

responses 181

QUERY NAMES command

responses 182

QUERY operand

of TAG command 234

QUERY PFnn command

responses 178-179

QUERY PRINTER command

responses 176-178

QUERY PROCESSR command

responses 180

QUERY PUNCH command

responses 176-178

QUERY READER command

responses 176-178

QUERY SCREEN command

responses 175

QUERY SECUSER command

responses 175

QUERY SET command

response 164-165

QUERY SPMODE command

responses 181

QUERY S370E command

responses 180

QUERY TERMINAL command

response 165

QUERY TIME command

response 163-164

QUERY USERS command

responses 182-184

QUERY VIRTUAL CHANNELS command

response 166

QUERY VIRTUAL command

operands

CHANNELS 159

CONSOLE 159

DASD 159

GRAF 159

LINES 159

STORAGE 159

TAPES 159

UR 159

vaddr 159

QUERY VIRTUAL CONSOLE command

response 167-168

QUERY VIRTUAL DASD command

responses 169-170

QUERY VIRTUAL GRAF command

responses 166

QUERY VIRTUAL LINES command

responses 171

QUERY VIRTUAL STORAGE command

responses 174

QUERY VIRTUAL TAPES command

responses 170

QUERY VIRTUAL UR command

responses 171-174

QUERY VMSAVE command

responses 179-180

querying

spool file characteristics 18

R

- raddr
 - explained 48
- RANGE option
 - of PER command 138
- read/write access
 - with LINK command 113-118
- reader files
 - altering attributes of virtual 59-62
- READER operand
 - of CHANGE command 59
 - of CLOSE command 63
 - of DEFINE command 73
 - of ORDER command 131
 - of PURGE command 155
 - of QUERY command 160
 - of SPOOL command 216
 - of TRANSFER command 253
- reading an entire file continuously 216-226
- READY command
 - described 185
 - operand
 - vaddr 185
 - syntax diagram 185
 - usage notes 185
- ready status
 - removing a device from 130
- real reset button
 - simulating 231
- real restart button
 - simulating 231
- reclaiming
 - files before purging 155-156
 - transmitted files 253-256
- reconfiguring virtual machine 28
- redefining
 - device configuration 79, 115
- reg
 - explained 48
- regcount
 - of DISPLAY command 93
- registers
 - displaying extended control 91-96
 - displaying floating-point 91-96
 - displaying general 91-96
 - printing extended control 97-100
 - printing floating-point 97-100
 - printing general 97-100
- releasing
 - dedicated devices 21-22
 - temporary disks 16
- remote spooling
 - TAG command 233-236
- remote spooling networks
 - receiving files 226
 - transmitting files 225-226
- remote terminal
 - connecting 86-87
- removing
 - closed spool files 155-156
 - dedicated devices 21-22
 - virtual devices from virtual machine 83-85
- reordering spool files 21
- REQUEST command
 - described 186
 - syntax diagram 186
 - usage notes 186
- reserved IBM class
 - described 9
- reset button
 - simulating 231
- RESET command
 - described 187
 - operand
 - vaddr 187
 - response 187
 - syntax diagram 187
 - usage notes 187
- RESET operand
 - of SYSTEM command 231
- resetting preferred machine assist 107-112
- resetting virtual machine 76
- resid
 - explained 48
- responses
 - COMMANDS command 67
 - COUPLE command 68-69
 - EXTERNAL command 102
 - for #CP command 54
 - for ADSTOP command 56
 - for CHANGE command 62
 - for CLOSE command 66
 - for CP command 71
 - for DEFINE command 82
 - for DETACH command 84
 - for DIAL command 87
 - for DISCONN command 90
 - for DISPLAY command 95-96
 - for DUMP command 100
 - for ECHO command 101
 - for INDICATE command 103-106
 - for INDICATE LOAD command 103-104
 - for INDICATE USER command 105-106
 - for IPL command 112
 - for LINK command 117-118
 - for LOGOFF command 123-124
 - for LOGON command 127
 - for MESSAGE command 129
 - for ORDER command 132
 - for PURGE command 156
 - for QUERY command 163-184
 - for QUERY CPLEVEL command 181
 - for QUERY CPUID command 165-166
 - for QUERY FILES command 166
 - for QUERY LINKS command 174
 - for QUERY LOGMSG command 181
 - for QUERY NAMES command 182

S

- for QUERY PFnn command 178-179
- for QUERY PROCESSR command 180
- for QUERY READER
PRINTER, and PUNCH command 176-178
- for QUERY SCREEN command 175
- for QUERY SECUSER command 175
- for QUERY SET command 164-165
- for QUERY SPMODE command 181
- for QUERY S370E command 180
- for QUERY TERMINAL command 165
- for QUERY TIME command 163-164
- for QUERY USERS command 182-184
- for QUERY VIRTUAL CHANNELS
command 166
- for QUERY VIRTUAL CONSOLE
command 167, 168
- for QUERY VIRTUAL DASD command 169-170
- for QUERY VIRTUAL GRAF command 166
- for QUERY VIRTUAL LINES command 171
- for QUERY VIRTUAL STORAGE
command 174
- for QUERY VIRTUAL TAPES command 170
- for QUERY VIRTUAL UR command 171-174
- for QUERY VIRTUAL vaddr command 174
- for QUERY VMSAVE 179-180
- for RESET command 187
- for REWIND command 188
- for SPOOL command 226
- for STORE command 230
- for SYSTEM command 231
- for TAG command 235
- for TRACE command 250-252
- for TRANSFER command 255-256
- for VMDUMP command 261
- responses to CP commands
types of 51
- responses vs. system messages 51
- restart button
simulating 231
- RESTART operand
of SYSTEM command 231
- restarting virtual machine 24
- resuming
execution 24
use of virtual machine after
disconnecting 89-90
virtual machine execution 58
- REWIND command
described 188
operands
vaddr 188
responses 188
syntax diagram 188
usage notes 188
- rewinding a real tape 188
- RUN operand
of SET command 197
of TRACE command 248
- RUN option
of PER Command 140

- SAVE sub-command
of PER command 143
sub-commands
APPEND 143
traceset 143
- saved systems
loading 107-112
- saving virtual machine data 227-230
- SCREEN command
described 189
operands
ALL 189
area 189
CPOUT 190
extcolor 190
exthilight 190
INAREA 189
INREDISP 190
OUTAREA 189
STATAREA 189
VMOUT 190
syntax diagram 189
usage notes 190-191
- SCREEN operand
of QUERY command 162
- SCRNSAVE operand
of TERMINAL command 241
- secondary users
list of current 157-184
- SECUSER operand
of QUERY command 160
- SEL operand
of DEFINE command 76
- SEND command
described 192
operands
CP 192
text 192
userid 192
syntax diagram 192
usage notes 192
- sending messages
to other users 128-129
to the operator 128-129
- sending special messages 214-215
- service representative, described 11
- SET command
described 194
determining current values 157-184
operands 210
ACNT 195
AFFINITY 200
ASSIST 201-203
AUTOPOLL 195

- CODE 199
- CPCONIO 199
- CPUID 201
- ECMODE 197
- EMSG 199
- IMSG 196
- ISAM 198
- IUCV 199
- LINEDIT 197
- MSG 196
- NOTRANS 198
- OFF 195, 196, 197, 198, 199, 200
- ON 199
- PAGEX 198
- PFnn 203-208
- RUN 197
- SMSG 196
- STBypass 208-209
- STMULTI 209
- TEXT 199
- TIMER 200
- VMCONIO 199
- VMSAVE 208
- WNG 196
- XMEM 210
- 370E 210
- syntax diagram 194
- usage notes 211
- SET operand
 - of QUERY command 159
- setting
 - device-end interruption for virtual device 185
 - pool file characteristics 18
- setting up a terminal session 13
- shadow tables
 - enabling multiple 194-211
 - setting bypass 194-211
- shared segment
 - modifying 55-56, 227-230, 249
- sharing virtual disks 16
- shutdown
 - pooling considerations 226
- signalling attention interruptions 237-245
- simulating
 - attention interruption 57, 107-112
 - hardware store 34
 - interruptions 25
 - not ready status for a virtual device 130
 - real reset button 231
 - real restart button 231
- SIO operand
 - of TRACE command 247
- SLEEP command
 - described 212
 - operand
 - HRs 212
 - MIN 212
 - nn SEC 212
 - syntax diagram 212
 - usage notes 212
- SMSG command
 - described 214
 - operands
 - msgtext 214
 - userid 214
 - syntax diagram 214
 - usage notes 214-215
- SMSG operand
 - of SET command 196
- special messages
 - controlling receipt 194-211
- special symbols
 - use of 49
- SPMODE operand
 - of QUERY command 162
- spool class
 - changing 59-62
 - modifying 216-226
- SPOOL command
 - described 216
 - operands
 - * 218
 - CHARS name 221
 - CLASS 217
 - CLOSE 220
 - CONSOLE 217
 - CONT 217
 - COPY[[*] nnn 219
 - DIST distcode 222
 - EOF 218
 - FCB name 221
 - FLASH name 220
 - FOR 219
 - FORM form 222
 - HOLD 218
 - MODIFY name 220
 - NOCONT 217
 - NOEOF 218
 - NOHOLD 218
 - NOTERM 221
 - OFF 219
 - PRINTER 216
 - PUNCH 217
 - PURGE 220
 - READER 216
 - START 221
 - STOP 221
 - SYSTEM 218
 - TERM 221
 - TO 218
 - userid 218
 - vaddr 217
 - response 226
 - usage notes 222-226
 - use with TAG command 216
- spool command options
 - virtual device 63
- spool file
 - altering TAG information 233-236
 - changing attributes of 59-62

- changing characteristics 18-19
- changing filename 59-62
- control
 - CP commands used 274
- determining number of closed 157-184
- querying characteristics 18-19
- removing from system 155-156
- reordering 131-132
- setting characteristics 18-19
- TAG command 233-236
- transmitting to remote location 233-236
- spoolid
 - explained 49
- spoolid number 223
- spoolid operand
 - of CHANGE command 59
 - of ORDER command 131
 - of PURGE command 155
 - of QUERY command 161
 - of TRANSFER command 253
- spooling
 - changing class 59-62
 - changing file attributes 59-62
 - changing filename 59-62
 - changing spoolid 59-62
 - class considerations 222
 - considerations during abend 226
 - considerations during shutdown 226
 - described 17
 - determining number of closed files 157-184
 - modifying control options 216-226
 - operator, described 9
 - purging files 21
 - removing closed files from the system 155-156
 - reordering files 21, 131-132
 - spoolid 223
 - transmitting files locally 224
 - transmitting files to remote locations 225-226
 - virtual console 20
- START operand
 - of SPOOL command 221
- STAtAREA operand
 - of SCREEN command 189
- status
 - storing virtual machine 34
- STATUS operand
 - of STORE command 229
- status words
 - tracing 246-252
- STBypass operand
 - of SET command 208-209
- STEP option
 - of PER command 140
- STMULTI operand
 - of SET command 209-210
- STOP operand
 - of IPL command 108
 - of SPOOL command 221
- stopping
 - a virtual machine 194-211
 - initial program load (IPL) procedure 107-112
 - stopping execution
 - virtual machine 29
- storage
 - altering virtual 33-35
 - clearing to zeros 107-112
 - clearing virtual 24
 - modifying 55-56
- storage keys
 - displaying 91-96
 - printing 97-100
- storage locations
 - altering virtual machine 227-230
 - displaying virtual 91-96
 - displaying virtual machine 91-96
 - printing 97-100
- STORAGE operand
 - of DEFINE command 78
 - of QUERY VIRTUAL command 159
- STORE command
 - described 227
 - operands
 - Greg 228
 - hexloc 227
 - PSW 228
 - STATUS 229
 - Xreg 228
 - Yreg 228
 - response 230
 - syntax diagram 227
 - usage notes 229-230
- STORE event-type
 - of PER command 136
 - MASK 137
- storing
 - CPU status 227-230
 - into virtual storage 33-35
 - virtual machine data 227-230
 - virtual machine status 34
- summary of CP commands 263-269
 - by function 271-275
- suppressing automatic loading 24, 107-112
- SVC operand
 - of TRACE command 246
- syntax diagrams
 - ADSTOP command 55
 - ATTN command 57
 - BEGIN command 58
 - CHANGE command 59
 - CLOSE command 63
 - COMMANDS command 67
 - COUPLE command 68
 - CP commands 70
 - DEFINE command 72
 - DETACH command 83
 - DIAL command 86
 - DISCONN command 89
 - DISPLAY command 91
 - DUMP command 97
 - ECHO command 101

- EXTERNAL command 102
- INDICATE command 103
- IPL command 107
- LINK command 113
- LOADVFCB command 119
- LOGOFF command 122
- LOGON command 125
- NOTREADY command 130
- ORDER command 131
- PER command 134
- PURGE command 155
- QUERY command 158
- READY command 185
- REQUEST command 186
- RESET command 187
- REWIND command 188
- SCREEN command 189
- SEND command 192
- SET command 194
- SLEEP command 212
- SMSG command 214
- STORE command 227
- SYSTEM command 231
- TAG command 233
- TERMINAL command 238
- TRACE command 246
- TRANSFER command 253
- VMDUMP command 257
- SYSHOLD operand
 - of QUERY command 161
- system
 - loading by name 24
- system analyst
 - described 9
- SYSTEM command
 - described 231
 - operands
 - CLEAR 231
 - RESET 231
 - RESTART 231
 - responses 231
 - usage notes 231
- system load
 - displaying 103-106
- system messages vs. CP command responses 51
- SYSTEM operand
 - of SPOOL command 218
 - of VMDUMP command 259
- system programmer
 - described 9
- system resource operator
 - described 9
- system resources
 - displaying use of 103-106
- System/370 Extended (hardware) special feature 194-211
- systemname operand
 - of IPL command 109
- S370E operand

- of QUERY command 162

T

- TABCHAR operand
 - of TERMINAL command 239
- TABLE sub-command
 - of PER command 142
- TAG command
 - altering TAG information 233-236
 - described 233
 - operands
 - CONSOLE 233
 - DEV 234
 - DEV PRT 233
 - FILE 234
 - FILE spoolid 233
 - PCH 233
 - QUERY 234
 - tagtext 233
 - vaddr 233
 - responses 235
 - syntax diagram 233
 - transmitting files to remote location 233-236
 - usage notes 234-235
 - use with SPOOL command 216
- tagtext operand
 - of TAG command 233
- tapes
 - rewinding 188
- TAPES operand
 - of QUERY VIRTUAL command 159
- TBL operand
 - of QUERY command 161
- temporary disk space
 - clearing 72-82
 - formatting 72-82
- temporary disks
 - defining 16
 - detaching 16
 - releasing 16
- temporary virtual disks 16
- TERM operand
 - of SPOOL command 221
- terminal
 - connecting remote 86-87
 - control of attention handling 237-245
 - controlling input/output processing 237-245
 - controlling line length 237-245
 - disconnecting from virtual machine 89-90
 - testing 101
- TERMINAL command
 - described 237
 - determining current values 157-184
 - operands
 - APL 240
 - ATTN 241
 - BREAKIN 242

BRKKEY 243
 CHARDEL 238
 CONMODE 242
 ESCAPE 239
 HILIGHT 241
 LINEDEL 239
 LINEND 239
 LINESIZE 242
 MODE 241
 SCRNSAVE 241
 TABCHAR 239
 TEXT 240
 syntax diagram 238
 usage notes 244-245
 TERMINAL operand
 of QUERY command 159
 of TRACE command 247
 TERMINAL option
 of PER command 140
 terminal session
 setting up 13
 summary of commands used to control 263-269
 terminating
 console input/output 216-226
 processing of files 63
 testing
 controlling CCW 194-211
 programs 29
 terminal 101
 TEXT
 control of translation tables 237-245
 text format
 of CP commands 51
 TEXT operand
 of SEND command 192
 of TERMINAL command 240
 TFB-512 devices
 defining fixed-block mode 72-82
 TFB-512 operand
 of DEFINE command 78
 time
 determining length of a terminal
 session 157-184
 TIME operand
 of QUERY command 159
 timer
 controlling virtual 194-211
 TIMER operand
 of DEFINE command 74
 of SET command 200
 TO operand
 of SPOOL command 218
 of TRANSFER command 253
 of VMDUMP command 259
 TRACE
 turning off 36
 TRACE command
 described 246
 operands
 ALL 247
 BOTH 248
 BRANCH 247
 CCW 247
 CSW 247
 END 247
 EXTERNAL 246
 I/O 246
 INSTRUCT 247
 NORUN 248
 OFF 248
 PRINTER 247
 PRIV 246
 PROGRAM 246
 RUN 248
 SIO 247
 SVC 246
 TERMINAL 247
 responses 250-252
 syntax diagram 246
 usage notes 248-250
 TRACE output
 closing 36
 tracing
 branches 246-252
 input/output operations 246-252
 interruptions 246-252
 privileged instructions 246-252
 status words 246-252
 virtual machine activity 35-36, 246-252
 TRANSFER command
 described 253
 operands
 ALL 253
 CLASS 253
 FORM form 253
 FROM 253
 PRINTER 253
 PUNCH 253
 READER 253
 spoolid 253
 TO 253
 userid 253
 responses 255-256
 syntax diagram 253
 usage notes 254-255
 transferring files
 to a specified queue 253-256
 to a specified user 253-256
 to another user 216-226
 transmitting
 spool files locally 224
 spool files to remote locations 225-226
 turning off an ADSTOP 30
 T2305 operand
 of DEFINE command 78
 T2314 operand
 of DEFINE command 78
 T2319 operand
 of DEFINE command 78
 T3310 operand

- of DEFINE command 78
- T3330 operand
 - of DEFINE command 78
- T3340 operand
 - of DEFINE command 78
- T3350 operand
 - of DEFINE command 78
- T3370 operand
 - of DEFINE command 78
- T3375 operand
 - of DEFINE command 78
- T3380 operand
 - of DEFINE command 78

U

- underscore
 - use of 49
- unit record devices
 - virtual 17
- uppercase
 - use of 49
- uppercase letters and words 47
- UR operand
 - of QUERY VIRTUAL command 159
- usage notes
 - #CP command 53-54
 - ADSTOP command 55-56
 - ATTN command 57
 - BEGIN command 58
 - CHANGE command 61
 - CLOSE command 65
 - COMMANDS command 67
 - COUPLE command 68
 - CP command 70
 - DEFINE command 79-81
 - DETACH command 83
 - DIAL command 86-87
 - DISCONN command 89
 - DISPLAY command 94-95
 - DUMP command 98
 - ECHO command 101
 - event types
 - BRANCH 151
 - INSTRUCT 151
 - STORE 148
 - EXTERNAL command 102
 - INDICATE command 103
 - IPL command 110-112
 - LINK command 115-117
 - LOADVFCB command 119-121
 - LOGOFF command 122
 - LOGON command 125-126
 - MESSAGE command 128-129
 - NOTREADY command 130
 - options
 - NORUN 148
 - PASS 151

- RUN 148
- STEP 148
- ORDER command 131-132
- PER command 147-152
- PURGE command 155-156
- READY command 185
- REQUEST command 186
- RESET command 187
- REWIND command 188
- SCREEN command 190-191
- SEND command 192
- SET command 211
- SLEEP command 212
- SMSG command 214-215
- SPOOL command 222-226
- STORE command 229-230
- SYSTEM command 231
- TAG command 234-235
- TERMINAL command 244-245
- TRACE command 248-250
- TRANSFER command 254-255
- VMDUMP command 260-261
- USER operand
 - of INDICATE command 103
- USERHOLD operand
 - of QUERY command 161
- userid
 - explained 49
- userid operand
 - of COUPLE command 68
 - of DIAL command 86
 - of LINK command 113
 - of LOGON command 125
 - of MESSAGE command 128
 - of SEND command 192
 - of SMSG command 214
 - of SPOOL command 218
 - of TRANSFER command 253
 - of VMDUMP command 259
- users
 - determining current number on system 157-184
 - listing names currently on system 157-184
- using virtual forms control buffer
 - with 3262 printer 119-121
 - with 3289E printer 119-121

V

- vaddr
 - explained 49
- vaddr operand
 - of CLOSE command 63
 - of COUPLE command 68
 - of DEFINE command 77
 - of DETACH command 83
 - of DIAL command 86
 - of IPL command 107

- of LINK command 113
- of LOADVFCB command 119
- of NOTREADY command 130
- of QUERY VIRTUAL command 159
- of READY command 185
- of RESET command 187
- of REWIND command 188
- of SPOOL command 217
- of TAG command 233
- virtual
 - input devices
 - action taken on a CLOSE 65
 - output devices
 - action taken on a CLOSE 65
- virtual addresses 72-82
- virtual console
 - CP commands
 - used for 273
 - disconnecting 89-90
 - spooling 20, 216-226
- virtual devices
 - closing 216-226
 - defining 72-82
 - detaching 83-85
 - determining current number of 157-184
 - determining the status of 157-184
 - removing from your virtual machine 83-85
 - setting a device-end interruption for 185
 - simulating not ready status for 130
- virtual disks
 - managing 15
 - managing permanent 15
 - managing temporary 16
 - sharing 16
- virtual I/O devices
 - defining 72-82
- virtual machine
 - command environment 5
 - configuration
 - altering 72-82
 - determining status of 157-184
 - device
 - linking to 113-118
 - directory entry 14
 - disconnecting 122-124
 - environment
 - altering 72-82
 - entering CP commands 70
 - execution
 - resuming 58
 - logging off 122-124
 - logging on 125-127
 - placing in a dormant state 212
 - reconfiguring 28
 - resetting with preferred machine assist 76
 - restarting 24
 - setting control functions 194-211
 - status
 - storing 34
 - stopping execution 29
 - storing information from 227-230
 - tracing activity 35-36
 - tracing activity of 246-252
- VIRTUAL operand
 - of QUERY command 159
- virtual operating system
 - passing parameters to 107-112
- virtual printer
 - display current attributes 157-184
 - specifying forms controls for 119-121
- virtual punch
 - display current attributes 157-184
- virtual reader
 - display current attributes 157-184
- virtual storage
 - altering 33-35
 - changing 227-230
 - clearing 24
 - displaying 30
 - dumping 32-33, 257-261
- virtual timer
 - controlling 194-211
- virtual unit record devices 17
- virtual volumes (3330V)
 - defining 72-82
- VM/VS handshaking
 - control of 194-211
- VMCONIO operand
 - of SET command 199
- VMDUMP command
 - described 257
 - operands
 - * 259
 - *dumpid 260
 - bytecount 259
 - DSS 259
 - END 258
 - FORMAT vmttype 259
 - hexloc 258
 - SYSTEM 259
 - TO 259
 - userid 259
 - responses 261
 - syntax diagram 257
 - usage notes 260-261
- VMOUT operand
 - of SCREEN command 190
- VMSAVE operand
 - of QUERY command 162
 - of SET command 208
- valid
 - explained 49

W

warning messages
controlling receipt 194-211
WNG operand
of SET command 196
writing comments 6

X

XMEM operand
of SET command 210
Xreg operand
of DISPLAY command 93
of STORE command 228

Y

Yreg operand
of DISPLAY command 93
of STORE command 228

Z

zeros
clearing storage to 107-112, 231

Numerics

1 operand
of ECHO command 101
1403 operand
of DEFINE command 74
1443 operand
of DEFINE command 74
2501 operand
of DEFINE command 74
2540P operand
of DEFINE command 74
2540R operand
of DEFINE command 74
3036 operand
of DEFINE command 76
3066 operand
of DEFINE command 76
3088 operand

of DEFINE command 74
3138 operand
of DEFINE command 76
3148 operand
of DEFINE command 76
3158 operand
of DEFINE command 76
3178
size of input line 128
3203 device
LOADVFCB restrictions 119-121
3203 operand
of DEFINE command 74
3203 printer
specifying forms controls for virtual 119-121
3211 operand
of DEFINE command 74
3211 printer
specifying forms controls for virtual 119-121
3262 operand
of DEFINE command 74
3262 printer
loading a virtual forms control buffer 119-121
3270 device message length 214
3270 operand
of DEFINE command 76
3270 terminal
power off vs. logoff precautions 122-124
3270 terminal display
highlighting features available 244
3289E operand
of DEFINE command 74
3289E printer
loading a virtual forms control buffer 119-121
3505 operand
of DEFINE command 75
3525 operand
of DEFINE command 75
370E feature
enabling 194-211
370E operand
of SET command 210
3800 operand
of DEFINE command 75
3800 printer
modifying device attributes 216-226
modifying spool file attributes 62
print file duplication 59-62
spooling output to 223
using SPOOL command 216-226
3880 Storage Subsystem device
limits to defining 81
40 operand
of EXTERNAL command 102
4245 operand
of DEFINE command 76
4248 operand
of DEFINE command 76

Virtual Machine/
System Product
High Performance Option

READER'S
COMMENT
FORM

CP Command Reference
for General Users

Order No. SC19-6227-5

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.

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.*

How did you use this publication?

- | | |
|--|---|
| <input type="checkbox"/> As an introduction | <input type="checkbox"/> As a text (student) |
| <input type="checkbox"/> As a reference manual | <input type="checkbox"/> As a text (instructor) |
| <input type="checkbox"/> For another purpose (explain) _____ | |

Is there anything you especially like or dislike about the organization, presentation, or writing in this manual? Helpful comments include general usefulness of the book; possible additions, deletions, and clarifications; specific errors and omissions.

Page Number:

Comment:

What is your occupation? _____

Newsletter number of latest Technical Newsletter (if any) concerning this publication: _____

If you wish a reply, give your name and address: _____

IBM branch office serving you _____

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 automatic mail-sorting equipment. Please use pressure-sensitive or other gummed tape to seal this form.

Reader's Comment Form

Fold and Tape

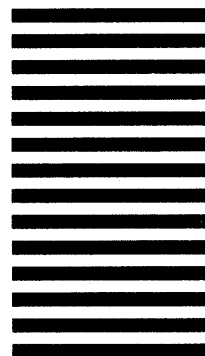
Please Do Not Staple

Fold and Tape



NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL
FIRST CLASS PERMIT NO. 40 ARMONK, N.Y.



POSTAGE WILL BE PAID BY ADDRESSEE

International Business Machines Corporation
Department 52Q MS 458
Neighborhood Road
Kingston, New York 12401



Fold and Tape

Please Do Not Staple

Fold and Tape

PRINTED IN U.S.A. SC19-6227-5



SC19-6227-5

VM/SP HPO CP Command Reference for General Users File No. S370-36 Printed in U.S.A. SC19-6227-5



SC19-6227-05

