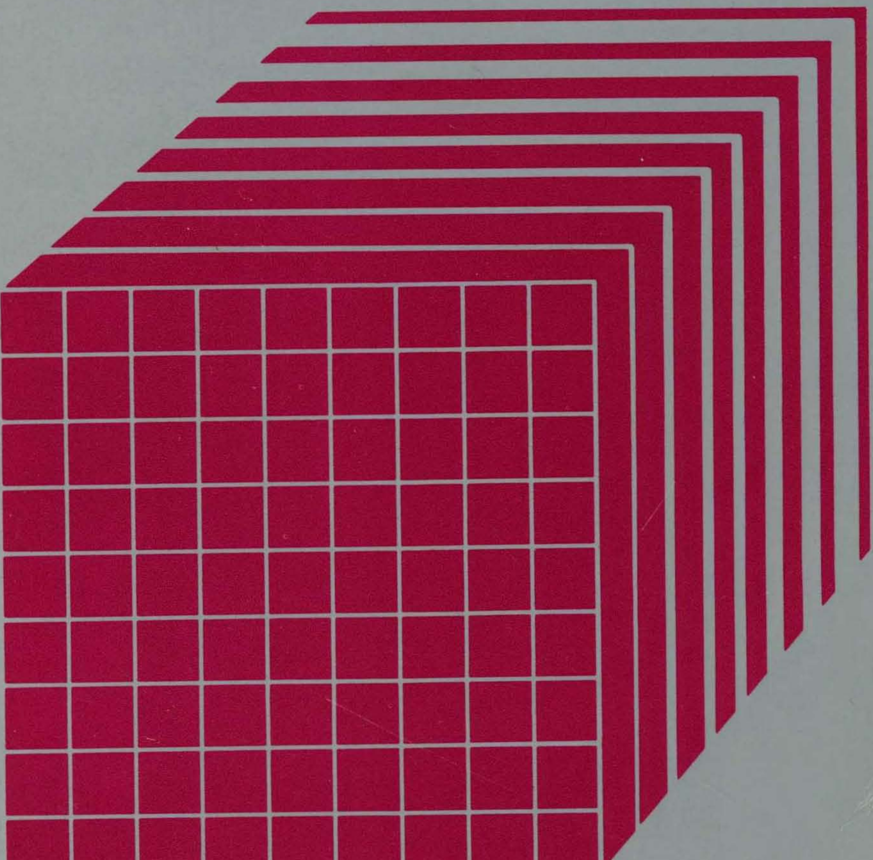
IBM

Virtual Machine/ System Product

CP Command Reference for General Users

Release 4

SC19-6211-3



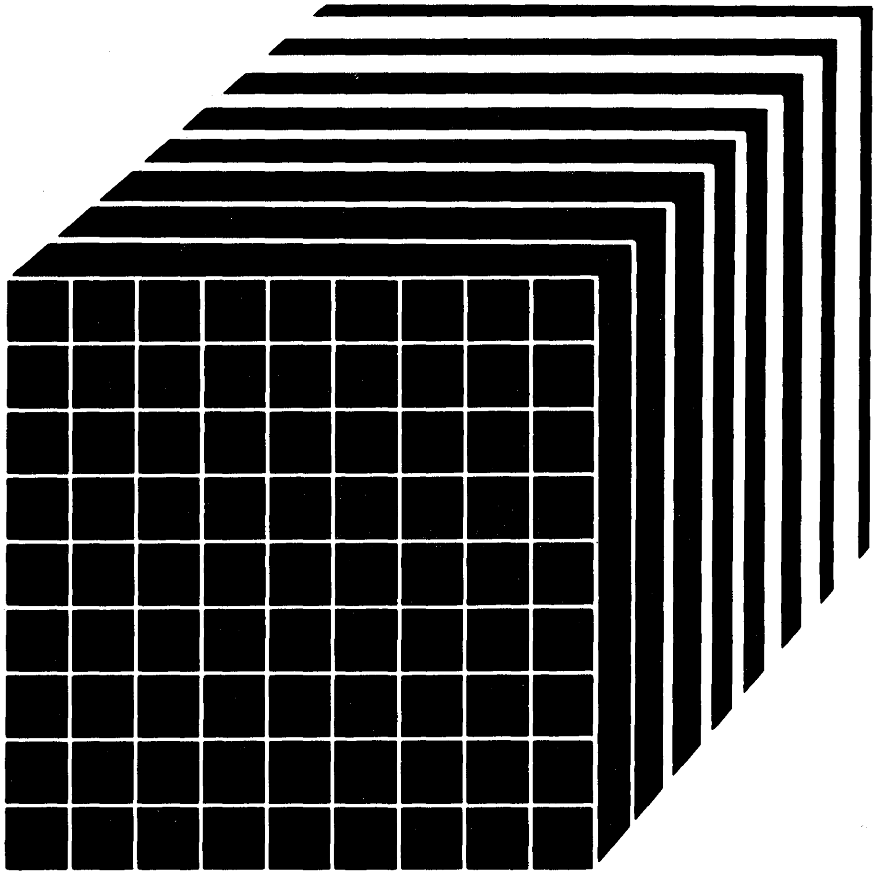
IBM

Virtual Machine/
System Product

CP Command Reference for General Users

Release 4

SC19-6211-3



Notice: The term Virtual Machine/System Product (abbreviated VM/SP), as used in this publication, refers to VM/SP when used in conjunction with VM/370 Release 6.

Fourth Edition (December 1984)

This edition, (SC19-6211-3) is a major revision of SC19-6211-2. This edition applies to the Virtual Machine/System Product Release 4, program number 5664-167, unless otherwise indicated in new editions or Technical Newsletters. Changes are periodically made to the information contained herein; before using this publication in connection with the operation of IBM systems, consult the latest *IBM System/370 and 4300 Processors Bibliography*, GC20-0001, for the editions that are applicable and current.

Summary of Changes

For a detailed list of changes, see page iii.

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

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

Ordering Publications

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

A form for reader's comments is provided at the back of this publication. If the form has been removed, comments may be addressed to IBM Programming Publications, Dept. G60, P.O. Box 6, Endicott, NY, U.S.A. 13760. IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you.

Summary of Changes

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

Release 3, order STOO-1359
Release 2, order SQ19-6211
Release 1, order ST19-6211

Summary of Changes
for SC19-6211-3
for VM/SP Release 4

VMDUMP Enhancements

Diagnose function X'94' is available to allow a virtual machine to request dumping of its virtual storage. Also, the three address range restriction has been removed from the VMDUMP command.

SHARED/NONSHARED Restriction

With the addition of this support, any attempt to construct a virtual device configuration that would mix SHARED and NONSHARED device types on the same virtual control unit is rejected.

TERMINAL Enhancements

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

The GRAF and CONSOLE options of the QUERY command contain additional function.

The SIO instruction can be used with attached remote printers and dedicated graphics terminal devices.

Summary of Changes
for SC19-6211-2
for VM/SP Release 3

PER COMMAND

The PER command is added to allow monitoring of certain events in the user's virtual machine as they occur during program execution.

CMS PERFORMANCE ENHANCEMENT

The discontiguous saved segment, CMSSEG, has been removed by merging modules from the CMSSEG segment into the CMS shared nucleus.

DIAGNOSE '14'

DIAGNOSE X'14' has been modified so that subcodes X'OFFE' and X'OFFF' respect the SFBINUSE flag when the file is not in use by the reader. These subcodes wait for a file being used by a system function. If, however, the file is not available within the 25-millisecond time limit a condition code of 3 and a return code of 24 is returned. This condition shows system problems because of performance or errors in the spooling area.

4245 LINE PRINTER SUPPORT

The 4245 Line Printer is a channel-attached, high-speed line printer (2000 LPM) using print band technology. It combines 3203 and 3262 technologies with newly developed hardware. The programming support for this printer is similar to the 3262 programming support. The 4245 is supported as a virtual, dedicated, and system printer. In addition, FCB lengths for the 4245 will be allowed the maximum length that hardware supports (256 byte length). It is not limited to a 180-byte maximum length as are previous printers.

3088 MULTISYSTEM CHANNEL COMMUNICATIONS UNIT

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

VM/SP 3800 MODEL 3 COMPATIBILITY

VM users have access to the 3800 Model 1 and Model 3 Printing Subsystems. 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.

MISCELLANEOUS

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

Summary of Changes for SC19-6211-1 for VM/SP Release 2

PROGRAMMABLE OPERATOR FACILITY

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

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

DIAL COMMAND SUPPORT FOR REMOTE BSC 3270 USERS

This facility provides the ability for remote BSC 3270 users to issue the DIAL command. The remote 3270 appears to be a locally attached terminal to the dialed virtual machine.

ENHANCED 3800 SUPPORT

This facility allows the RSCS Networking Release 3 program product to pass to CP the 3800 attributes defined in a NJI header created by a system using the Network Job Interface (NJI) protocol. This will also allow files created in these systems to print properly on a 3800 controlled by VM/SP Release 2. Copy group and burst attributes remain unsupported.

COMMAND RETRIEVE CAPABILITY

A RETRIEVE option is added to the CP SET program function command. When specified, the system will save input lines. When the specified program function key is pressed, the saved input lines are redisplayed in the reverse sequence of entry.

QUERY COMMAND ENHANCEMENT

The QUERY command is enhanced to allow the user to specify the operands "USERID" or "CPLEVEL." The QUERY USERID command responds with the userid and with the system identifier of the system that the user is logged on to. The QUERY CPLEVEL command provides information relative to the software product's name (e.g. VM/SP), release number, service level number, nucleus creation date and time, along with the IPL date and time. Time zone information is also included.

ENHANCED ASCII SUPPORT

This support provides the 3101 terminal user with the following capability:

- Screen Management - controlled scrolling of output
- Improved Communication Line Utilization - compression of contiguous blanks

- Program Function Key Support (1-8)

In addition, support for other ASCII terminal users includes the following:

- Prompting - Line feed prompting can be selected to allow data to be entered starting in column 1

- Read operations can be command-chained to the write operations

- Line size control option has been added

MISCELLANEOUS

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

Preface

This publication is a reference manual that discusses the IBM-Defined Class G VM/SP CP commands available to the general user.

The user should be knowledgeable of the concepts of VM/SP as outlined in the *Virtual Machine/System Product Introduction*, Order No. GC19-6200. In addition, the user should be aware of the logon process, terminal-initiated interrupts, terminal modes, and the method of switching from CP to virtual machine mode and vice versa. This information is detailed in the *Virtual Machine/System Product Terminal Reference*, Order No. GC19-6206.

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.

Other VM/SP CP commands, available only to system operators, system programmers and analysts, and service representatives, are described in the publication *Virtual Machine/System Product Operator's Guide*, Order No. SC19-6202.

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 command environments and the general structure of the CP command language.

"Section 2. The CP Command Language" provides additional information on the CP command set, including the optional 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. Notational Conventions" describes the syntax that is used in this publication to display command formats and command truncation and abbreviation.

"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 Command Summary" contains all of the VM/SP 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 command structure.

In this publication, the following terminology is used:

- The term 'cylinder' is used to describe DASD space on Direct Access Storage Devices (count-key-data devices) supported by the VM/SP System Control Program. The term 'block' is used to describe DASD space on FB-512 devices. Specifically, FB-512 devices are the IBM 3310 and 3370 Direct Access Devices employing fixed block mode. The term 'DASD space' may appear in text when there is no need to differentiate between count-key-data devices and FB-512 devices.

- "2305" refers to the IBM 2305 Fixed Head Storage, Models 1 and 2.

- The term "3088" refers to the IBM 3088 Multisystem Communications Unit Models 1 and 2.

- The term, "3270," is used in this publication to refer to a series of display devices, namely, the IBM 3275, 3276, 3277, 3278, and 3279 Display Stations, and the 3290 Information Panel. A specific device type is used only when a distinction is required between device types. Information about display terminal usage also applies to the IBM 3138, 3148, 3158, 3033, and 3036 Display Consoles when used in display mode, unless otherwise noted.

- Any information pertaining to the IBM 3284 or 3286 printer also pertains to the IBM 3262, 3287, 3288 and 3289 printers unless otherwise noted.

- "3330" refers to the IBM 3330 Disk Storage Models 1, 2, 11; the IBM 3333 Disk Storage and Control Models 1 and 11; and the 3350 Direct Access Storage operating in 3330/3333 Model 1 or 3330/3333 Model 2 compatibility mode.

- "3340" refers to the IBM 3340 Disk Storage, Models A2, B1, and B2; and the 3344 Direct Access Storage, Model B2.

- "3350" refers to the IBM 3350 Direct Access Storage, Models A2 and B2, in native mode. :

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

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

- "3380" refers to IBM 3380 Direct Access Storage.

- "VM/370 hardware assist" refers to both the virtual machine assist function and VM/370 Extended Control-Program Support. Some form of the VM/370 hardware assist is available on all VM/370 supported System/370 systems except the 155 II, 165-III, and 4331. Refer to *Virtual Machine/System Product Planning Guide and Reference*, Order No. SC19-6201, for further device information on the VM/370 hardware assist and ECPS features.

- Any information pertaining to the IBM 2741 terminal also applies to the IBM 3767 terminal, Model 1, operating as a 2741, unless otherwise specified.

- The term "processor" used in this publication means the main processor which is the processor with I/O handling capabilities. The term "attached processor" means a processor that has no I/O capabilities; an attached processor is always linked to a main processor for utilization. VM/SP also supports real multiprocessor (MP) configurations on the 158MP, 168MP, and 3033MP processors. In a MP system, the term "IPL processor" refers to the processor that is initialized first during system generation. The term "nonIPL processor" refers to the processor that is initialized after the IPL processor. Both the IPL and nonIPL processors in a real MP configuration have I/O capabilities.

- "3800" refers to the IBM 3800 Printing Subsystem, Models 1 and 3. A specific device type is used only when a distinction is required between device types.

- "3850" refers to the IBM 3850 Mass Storage System.

- "4248" refers to the IBM 4248 Printer.

- Unless otherwise noted, the term VSE refers to the combination of the DOS/VSE system control program and the VSE/Advanced Functions program product.

  In certain cases, the term DOS is still used as a generic term. For example, the disk packs initialized for use with VSE or any predecessor DOS or DOS/VS system may be referred to as DOS disks.

For a glossary of VM/SP terms, see the *Virtual Machine/System Product Library Guide and Master Index*, Order No. GC19-6207.

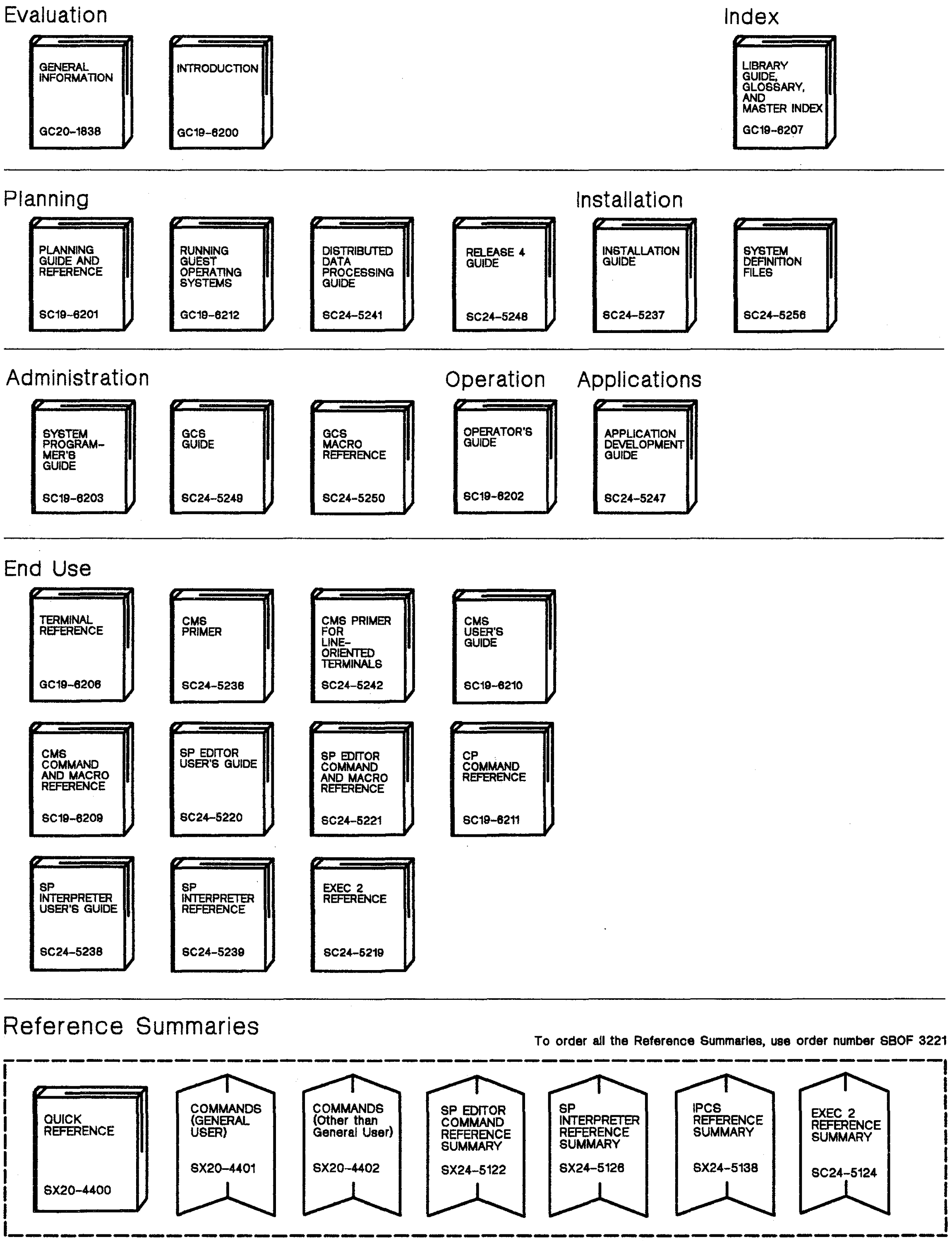
Prerequisite Publications

The concepts of VM/SP are described in the *Virtual Machine/System Product Introduction*, Order No. GC19-6200.

The physical operation of the virtual machine console is covered in the *Virtual Machine/System Product Terminal Reference*, Order No. GC19-6206.

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.

The VM/SP Library



Evaluation

GENERAL INFORMATION GC20-1838

INTRODUCTION GC19-6200

Index

LIBRARY GUIDE, GLOSSARY, AND MASTER INDEX GC19-6207

Planning

PLANNING GUIDE AND REFERENCE SC19-6201

RUNNING GUEST OPERATING SYSTEMS GC19-6212

DISTRIBUTED DATA PROCESSING GUIDE SC24-5241

RELEASE 4 GUIDE SC24-5248

Installation

INSTALLATION GUIDE SC24-5237

SYSTEM DEFINITION FILES SC24-5256

Administration

SYSTEM PROGRAM-MER'S GUIDE SC19-6203

GCS GUIDE SC24-5249

GCS MACRO REFERENCE SC24-5250

Operation

OPERATOR'S GUIDE SC19-6202

Applications

APPLICATION DEVELOPMENT GUIDE SC24-5247

End Use

TERMINAL REFERENCE GC19-6206

CMS PRIMER SC24-5236

CMS PRIMER FOR LINE-ORIENTED TERMINALS SC24-5242

CMS USER'S GUIDE SC19-6210

CMS COMMAND AND MACRO REFERENCE SC19-6209

SP EDITOR USER'S GUIDE SC24-5220

SP EDITOR COMMAND AND MACRO REFERENCE SC24-5221

CP COMMAND REFERENCE SC19-6211

SP INTERPRETER USER'S GUIDE SC24-5238

SP INTERPRETER REFERENCE SC24-5239

EXEC 2 REFERENCE SC24-5219

Reference Summaries

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

QUICK REFERENCE SX20-4400

COMMANDS (GENERAL USER) SX20-4401

COMMANDS (Other than General User) SX20-4402

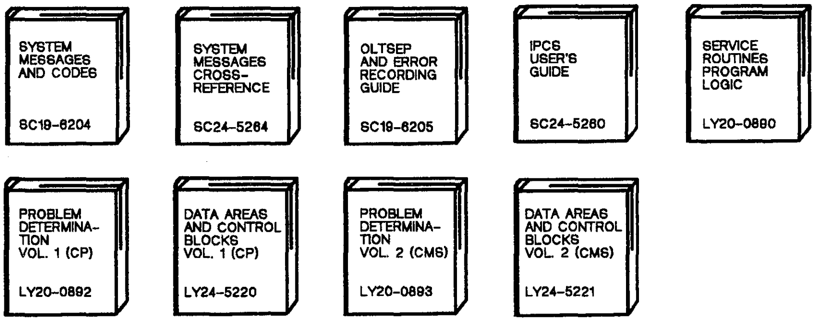
SP EDITOR COMMAND REFERENCE SUMMARY SX24-5122

SP INTERPRETER REFERENCE SUMMARY SX24-5126

IPCS REFERENCE SUMMARY SX24-5138

EXEC 2 REFERENCE SUMMARY SC24-5124

Diagnosis



SYSTEM MESSAGES AND CODES SC19-6204

SYSTEM MESSAGES CROSS-REFERENCE SC24-5264

OLTSEP AND ERROR RECORDING GUIDE SC19-6205

IPCS USER'S GUIDE SC24-5260

SERVICE ROUTINES PROGRAM LOGIC LY20-0890

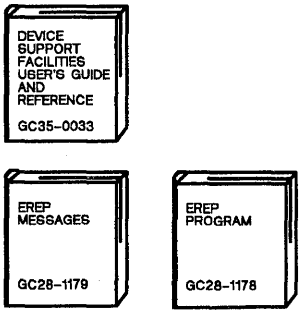
PROBLEM DETERMINATION VOL. 1 (CP) LY20-0892

DATA AREAS AND CONTROL BLOCKS VOL. 1 (CP) LY24-5220

PROBLEM DETERMINATION VOL. 2 (CMS) LY20-0893

DATA AREAS AND CONTROL BLOCKS VOL. 2 (CMS) LY24-5221

Auxiliary Service Support



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

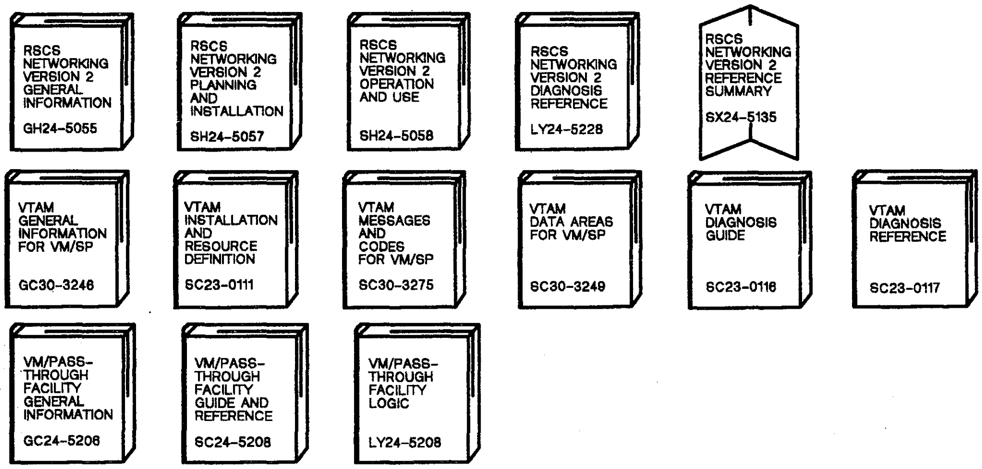
Device Support Facilities
5748-XX9

EREP MESSAGES GC28-1179

EREP PROGRAM GC28-1178

Environmental Recording 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 Communication Function For VTAM (ACF/VTAM)
5664-280

VM/PASS-THROUGH FACILITY GENERAL INFORMATION GC24-5206

VM/PASS-THROUGH FACILITY GUIDE AND REFERENCE SC24-5208

VM/PASS-THROUGH FACILITY LOGIC LY24-5208

VM/Pass-Through Facility
5748-RC1

Associated Publications

CP error messages and return codes that may be issued to a user during a terminal session are documented in *Virtual Machine/System Product System Messages and Codes*, Order No. SC19-6204.

General users intending to use CMS should refer to the following publications for information on the CMS command language and its implementation:

Virtual Machine/System Product

CMS Command and Macro Reference, Order No. SC19-6209

CMS User's Guide, Order No. SC19-6210

The RSCS component of VM/370 remains unmodified by the VM/SP program product. If the general user intends to use the unmodified VM/370 component, he should refer to the following publication for information on the RSCS command language:

Virtual Machine Facility/370:

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

In addition, VM/SP users can have enhanced networking facilities provided in the RSCS Networking program product (5748-XP1). The following publication contains information about the command languages of the RSCS Networking program product.

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

VM/SP has an enhanced interactive problem control system (VM/SP IPCS) component. This component replaces the unmodified VM/370 interactive problem control system. Details of this major component are found in the *VM/SP Interactive Problem Control System Guide*, SC24-5260.

Note: VM/SP IPCS provides extended function over the IPCS Extension Program Product (5748-SA1).

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

Contents

Part 1. Guidance Information 1

The CP Command Syntax 3
VM/SP Command Environments 3
VM/SP CP Command Structure 4
The Command Name 4
The Command Operands 4
Comments in the CP Command Language 4
Character Set Usage 5

The CP Command Language 7
Privilege Classes for CP Commands 7
The Class G Commands 9
The Class Any Commands 10

CP Command Usage 11
Controlling Input and Output Functions 13
Virtual Disks 15
Permanent Virtual Disks 15
Temporary Virtual Disks 15
Sharing Virtual Disks 16
Virtual Unit Record Devices 16
Virtual Unit Record Spooling 16
Spool File Characteristics 17
Virtual Console Spooling 19
Reordering and Purging Spool Files 20
Dedicated Devices 20
Dedicated Channels 21
Controlling the Virtual Machine 22
Loading an Operating System 22
Simulating Interrupts 24
Entering CP Commands 24
Entering CP Commands when the Virtual Machine Is Running 25
Entering CP Commands from the Virtual Console Read Environment 26
Entering CP Commands from CP Console Function Mode 26
Reconfiguring the Virtual Machine 27
Testing and Debugging of Programs 27
Stopping Execution of Your Virtual Machine 28
Displaying Virtual Storage 28
Terminal Output 29
Byte Alignment on Terminal Output 29

- Printer Output 30
- Altering Virtual Storage 31
- Tracing Virtual Machine Activity 33
- Using the PER Command 35
- Selectivity 36
- Terminating PER 37
- Suspending PER 37
- Additional Program Debugging Using PER 38
- The Branch Traceback Table 38
- The PER COUNT Sub-Command 39
- The PER Command Option 41
- Storage Alteration 42
- GUESTR and GUESTV 43

Part 2. Reference Information 45

Notational Conventions 47

Format of CP Commands 51

- * 52
- #CP 53
- ADSTOP 55
- ATTN 57
- BEGIN 58
- CHANGE 59
- CLOSE 64
- COMMANDS 68
- COUPLE 69
- CP 71
- DEFINE 73
- DETACH 83
- DETACH CHANNEL 86
- DIAL 87
- DISCONN 90
- DISPLAY 92
- DUMP 98
- ECHO 102
- EXTERNAL 103
- INDICATE 104
- IPL 108
- LINK 113
- LOADVFCB 119
- LOGOFF 122
- LOGON 125
- MESSAGE 129
- NOTREADY 131
- ORDER 132
- PER 134
  - Merging Trace Elements 146
  - Changing Options on Trace Elements 146
  - Changing Specific Trace Elements 147
- PURGE 157
- QUERY 159

QUERY Command for Class G Users 161
QUERY Command for All Classes of Users (Except Class Any) 183
READY 187
REQUEST 188
RESET 189
REWIND 190
SCREEN 191
SEND 194
SET 196
SLEEP 213
SMSG 215
SPOOL 217
STORE 228
SYSTEM 232
TAG 234
TERMINAL 238
TRACE 246
TRANSFER 253
VMDUMP 257

Appendix A. CP Command Summary 263

Index 271

Figures

1. Character Sets and Their Contents 5
2. IBM-Defined and Function Type CP Privilege Class Descriptions 8
3. Controlling the Virtual Machine During a Terminal Session 12
4. System Action to #CP Command 54
5. Action Taken on CLOSE for a Virtual Input Device. 66
6. Action Taken on CLOSE for a Virtual Output Device 67
7. Available Form Width Codes 81
8. Illustration of the Type of Link Established Under Every Condition . . . 116
9. 3270 High Intensity Features 245

Part 1. Guidance Information

This part of the publication contains general information about the Virtual Machine/System Product (abbreviated VM/SP) system. It contains three sections.

"The CP Command Syntax" describes the CP command environments and the general structure of the command languages.

"The CP Command Language" provides additional information on the CP command set.

"CP Command Usage" describes a variety of user requests via class G CP commands.

The CP Command Syntax

The CP component of VM/SP has its own command language. Much like any other language, the CP command language has a set a rules and structures. This section describes the CP command language environment, syntax and structure, giving general descriptions of command names, operands and their position on the command line, and character set usage in CP. A further discussion of CP commands is included in *VM/SP Operating Systems in a Virtual Machine.*

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 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. The procedure for creating your own CP commands is detailed in *VM/SP System Programmer's Guide.*

VM/SP 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 log on to VM/SP and issue CP commands.

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

If you are operating under CMS, you can determine which command environment you are in by entering a null line (that is, pressing the enter key, or equivalent, with no data). VM/SP responds to a null line by displaying the current command environment, CMS or CP.

The CP Command Syntax

VM/SP CP Command Structure

A VM/SP 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 alphameric 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 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 alphameric 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

VM/SP 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 1.

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

Figure 1. Character Sets and Their Contents

The CP Command Syntax

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 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 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 2 on page 8 lists each CP privilege class along with the associated type of user and function performed. Figure 2 on page 8 also identifies the specific publications in which each class subset of CP commands is described in detail.

For a complete list of CP commands grouped by privilege class, refer to *VM/SP Operator's Guide.*

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.

IBM-DEFINED CLASS	FUNCTION TYPE	USER AND FUNCTION
A*	O	***Operations:*** The primary system operator can issue all Class A commands. The class A user controls the VM/SP system. Class A is assigned to the user at the VM/SP system console during IPL. The primary system operator is responsible for the availability of the VM/SP 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. The system operator controls operation of the real machine using the system control panel and console device. *Note:* The class A system operator who is automatically logged on during CP initialization is designated as the primary system operator. 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.
B*	R	***Resource:*** 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 VM/SP system, except those controlled by the primary system operator and spooling operator.
C*	P	***Programmer:*** The system programmer can issue all Class C commands. The class C user updates certain functions of the VM/SP system. The system programmer can modify real storage in the real machine.
D*	S	***Spooling:*** 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.

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

IBM-DEFINED CLASS	FUNCTION TYPE	USER AND FUNCTION
E*	A	***Analyst:*** 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 discontiguous 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.
F*	C	***Customer Engineer:*** 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 VM/SP 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.
G+	G	***General:*** 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.
ANY+		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 VM/SP system.
		*Described in the VM/SP Operator's Guide. +Described in this publication.

Figure 2 (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 VM/SP 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.

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 system with console keys and switches. Figure 3 on page 12 lists the steps a user must follow in preparation for a terminal session (steps 1 and 2), and operation of a virtual machine (step 3).

CP Command Usage

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 VM/SP 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[2]	Enter CP commands to perform the function required	Available class G commands (Figure 9)
	a. activate terminal	LOGON command (Figure 9)
	b. transfer and store data	LINK, SPOOL, STORE, and TRANSFER commands (Figure 9)
	c. record and control data	CHANGE, CLOSE, DISPLAY, DUMP, ORDER, PURGE, QUERY, and SPOOL commands (Figure 9)

[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.

[2] Throughout the rest of this publication, the text is interspersed with examples of VM/SP commands and system responses. All user input is shown in lower case, and all system responses are shown in uppercase.

Figure 3 (Part 1 of 2). Controlling the Virtual Machine During a Terminal Session

STEP ACTION	FUNCTION	COMMAND DESCRIPTION
	d. reconfigure the virtual machine	DEFINE and DETACH commands (Figure 9)
	e. communicating with the virtual machine	#CP, ADSTOP, ATTN, BEGIN, EXTERNAL, IPL, REQUEST, SET commands (Figure 9)
	f. handling errors	TRACE, DISPLAY, and DUMP commands (Figure 9)
	g. end session	LOGOFF command (Figure 9)

Figure 3 (Part 2 of 2). 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
         ACCOUNT 87321 BIN14           (2)
(3)      OPTION ECMODE
         CONSOLE 01F 3215              (4)
(5)      SPOOL 00C 2540 R
         SPOOL 00D 2540 P
         SPOOL 00E 1403
(6)      MDISK 130 2314 050 099 VDISK1 WR RWPASSWD
         MDISK 191 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 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 corresponds 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 99 cylinder read/write minidisk with a virtual address of 130 is located on cylinders 50 through 149 of a real 2314 volume labeled VSDISK1. Similarly, a 20 cylinder read/write minidisk with a virtual address of 191 is located on cylinders 1 through 20 of a real 3330 volume labeled VDISK2. The LINK entry makes the virtual disk at virtual address 440 available to your virtual machine in read-only mode at virtual address 190. The last entry provides a link to a sharable CMS system volume so the user can use such CMS functions as the CMS editor or the EXEC processor.

Notes:

1. *The read-only access mode (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 LINK command means that you get write access to the disk as long as no one else has a link to it; if someone else links to it before you log on, you only have read-only access.*

3. *For detailed information on directory entries, see the section "Directory Control Statements" in the VM/SP Planning and System Generation Guide.*

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

Under VM/SP, 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, to VM/SP, are 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 VM/SP directory. 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 log on to the system.

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 t2319 as 133 cyl 15
```

In the previous example, a virtual 2319 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 (DSF) program.

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 log off.

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 either when you release that space or log off 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 same command line. For further information, refer to discussion of the LINK command in Section 5 of this publication.*

Virtual Unit Record Devices

The VM/SP 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 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 organizes the output data into a spool file. VM/SP 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 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. Some of these characteristics such as spooling class and distribution code are assigned in the VM/SP 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 identifies individual spool files by assigning each file a spool identification number (spoolid). This number ranges from 0001 through 9900 after 9900, VM/SP 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 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 assigns the number when the file is opened. Also, if you close a console file without stopping console spooling, VM/SP 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
```

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 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 VM/SP directory entry for a virtual machine at system generation or a device can be dedicated dynamically via the system operator's ATTACH command.

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 log on to the system. It remains dedicated to your virtual machine until you either log off or release the device via the DETACH command:

```
detach 495
```

The format of the DEDICATE control statement is described in detail in "Part 2: Defining Your VM/SP System" of the *VM/SP 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 log on 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 tying you down 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, 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) 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 CHANNEL 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
```

and 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 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 via the IPL command or automatically at logon time if an IPL control statement is included in your virtual machine's directory entry.

When a virtual machine runs the same operating system with very few exceptions, it may be expedient to use automatic loading. If the VM/SP 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 log on to VM/SP 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 via 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 can be 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 via the IPL command. The data is entered following the keyword PARM:

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

VM/SP 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 the “Usage Notes” discussion of the IPL command in Section 5.

Simulating Interrupts

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 interrupt code of X'40' is assumed. This code corresponds to pressing the external interrupt 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 VM/SP 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.

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 that you can interrupt the running of your virtual machine in order to execute 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; and, you can stop a virtual machine and place it directly into the CP environment.

CP Command Usage

Entering CP Commands when the Virtual Machine Is Running

If your terminal mode is set to VM, pressing the attention key (or its equivalent) once stops the virtual machine and places it into the virtual console read environment. 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 sign (#) in these examples represents the logical line end character currently in effect.

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 interrupt 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.

To restart the virtual machine, enter:

```
begin
```

CP Command Usage

Entering CP Commands from the Virtual Console Read Environment

When your virtual machine is stopped and in a console read environment, you can either enter CP commands directly or you can place the virtual machine in the CP console function mode.

To execute CP commands directly, enter one or more CP command lines as operands of the #CP command. For example, if you enter:

```
#cp query time#query users
```

the virtual machine executes the QUERY TIME and QUERY USERS commands and returns to the virtual console read environment. You can restart the virtual machine by entering the command:

```
#cp begin
```

If you enter the #CP command without any operands, the virtual machine is placed in CP console function mode. You can then enter CP commands such as:

```
query time
 -- or --
query users
```

directly. After each command, the virtual machine returns to the CP console function mode.

Entering the command:

```
begin
```

returns the virtual machine to the environment from which the #CP command was issued — in this case, the virtual console read environment.

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 VM/SP 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.

Testing and Debugging of Programs

In addition to the testing and debugging facilities that are contained in your virtual machine's operating system, VM/SP has several CP commands that you can use. These commands can be used 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, TRACE, and PER 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:

- The requested address is reached.
- The next ADSTOP command is issued.
- An IPL or a system reset is performed.
- The ADSTOP OFF command is issued.

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 via the DISPLAY command. All but the CAW and CSW can be printed on the spooled virtual printer via 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.

Terminal Output

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:

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

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 2K segment of the specified storage area are displayed. Contiguous 2K segment with identical storage keys are combined; for example, the response could have been:

```
001000 TO 0027FF  KEY=F0
002800 TO 003800  KEY=E0
```

To display all storage keys, enter:

```
display k
```

If your virtual machine is in extended control mode (ECMODE), you can interrogate 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
```

As each operand in the command line is processed, VM/SP 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.

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 Dec.	Address Hex	No. of Bytes	Data
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.

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, VM/SP trace output may have gaps in the tracing sequences. Each gap represents a period when VM/SP tracing was interrupted.

VM/SP 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 intermixed 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

The PER command can be used 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 allows PER to monitor events that occur as a result of 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. For example, 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 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

PER options can be used 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'18'), enter:

```
per instruct data 18
```

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    50D0C342 020348 CC=0
   02000A LA    41D0C33E 020344 CC=0
   020012 0C    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, the need arises to frequently clear your screen. The frequency with which events are displayed can be changed by using the PASS option. Ordinarily, every successful event is displayed. However, using the PASS option makes it possible 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 00000023 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

The PER SAVE subcommand can be used 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.

If you save a traceset under the same name as an existing traceset, the two are appended. PER does not overwrite a saved traceset.

Additional Program Debugging Using PER

TRACE can be used 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. PER, used in conjunction with TRACE, 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 Branch Traceback Table

The first step is to trace program interrupts using TRACE:

```
   trace prog
```

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

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

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

```
per branch 24600
```

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

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

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

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

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

The PER COUNT Sub-Command

Another method of finding the failing instruction is to use the PER COUNT sub-command with TRACE. This method, as well as the use of the PER TABLE command, is well suited for problems other than just operation exceptions. If the program is abending with any sort of program exception, 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 020000-0204FF TERMINAL NORUN
```

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

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

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

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

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

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

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

The PER Command Option

The PER CMD option can be used 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, the CMD option can be used 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. The CMD option can also be used 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

PER can be used 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.

It is also possible to 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.

It is also possible to monitor only alterations of storage when the storage value actually changes. For example:

```
per mask into 20100 data ffffffffffffff
```

monitors instructions that actually change one or more bytes in the doubleword field starting at location 20100. PER MASK can also be used 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. Using the GUESTR and GUESTV options, it is possible to choose which level activity will be traced. For example, when debugging VM/SP within a virtual machine it is sometimes helpful to limit trace output to either just second or just third level activity. 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 the command:

```
per i range 20000-21000
```

is issued, then both second and third level activity is traced (that is, both CP and CMS). To only trace the CP activity (second level) in the range 20000-21000, enter:

```
per i range 20000-21000 guestr
```

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

```
per i range 20000-21000 guestv
```

It is also possible to selectively trace 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
```

CP Command Usage

Part 2. Reference Information

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

"Notational Conventions" describes the syntax that is used in this publication to display command formats.

"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 and subcommand exists in the CP and XEDIT environments:

RESET.

Notational Conventions

The notation used to define the command syntax for VM/SP 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 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.

Notational Conventions

- 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	:

- 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.

- 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

$$\begin{matrix}\mathtt{A}\\ \underline{\mathtt{B}}\\ \mathtt{C}\end{matrix}$$

  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.

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

  Example:

  The representation

$$\begin{Bmatrix}\mathtt{A}\\ \mathtt{B}\\ \mathtt{C}\end{Bmatrix}$$

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

  $$\begin{bmatrix} \mathtt{A} \\ \mathtt{B} \\ \mathtt{C} \end{bmatrix}$$

  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 repeated more than once in succession.

  Example:

  The representation

  ```
  {spoolid1 spoolid2...}
  ```

  indicates that more than one spoolid may be coded within the braces.

Notational Conventions

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 VM/SP system messages. The command responses are not prefixed and, therefore, they are not contained in *VM/SP System Messages and Codes*. Only CP command responses applicable to the class G and class Any user are contained in this publication.
- ***Messages***: Lists the messages issued by the commands. Refer to *VM/SP System Messages and Codes* for detailed information about the messages. Refer to *VM/SP System Messages Cross-Reference* for a listing of messages.

*

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.

Messages

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. However, your installation may wish to redefine the logical linend 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.

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 virtual machine and the virtual machine enters CP environment.
#CP query files	QUERY command executed.
#CP query files#query users	Two separate QUERY commands executed.
data entered¢#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 entered¢#CP	"Data entered" is ignored. You enter CP environment.
#CP query files¢data entered	QUERY command is not executed; console input (data entered) passes to the virtual machine.

Figure 4. System Action to #CP Command

Responses

If you enter #CP without a commandline, you receive no response.

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

Messages

None.

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     { hexloc }
           { OFF    }
```

where:

`hexloc` is the 6-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 removed (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 6-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. Issue the SET RUN OFF command before the ADSTOP command.

6. Return codes for CP commands are the same as the error message numbers.

Response

```
ADSTOP AT xxxxxx
```

The instruction whose address is xxxxxx 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 xxxxxx, or at any other location desired.

Messages

```
DMKCFD004E INVALID HEXLOC - hexloc
DMKCFD026E OPERAND MISSING OR INVALID
DMKCFD160E HEXLOC hexloc EXCEEDS STORAGE
DMKCFD164E HEXLOC hexloc NON-ADDRESSABLE STORAGE
```

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 interrupts 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.

Messages

None.

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

The virtual machine begins execution.

Messages

```
DMKCFJ004E INVALID HEXLOC - hexloc
```

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   ⎧ Reader   ⎧ CLass c1   ⎫ ⎧ CLass c2        ⎫1⎫  ⎡      ⎧fn (ft)⎫⎤
         ⎪ Printer  ⎨ spoolid    ⎬ ⎪ ┌       ┐       ⎪ ⎪  ⎢NAme ⎨       ⎬⎥
         ⎪ PUnch    ⎪ FORM form1 ⎪ ⎪ │HOld   │       ⎪ ⎪  ⎣      ⎩dsname ⎭⎦
         ⎪          ⎩ ALL        ⎭ ⎨ │NOHold │       ⎪ ⎪
         ⎪                         ⎪ └       ┘       ⎪ ⎪
         ⎨                         ⎪ DIst dist       ⎬ ⎬
         ⎪                         ⎪ COpy [*]nnn     ⎪ ⎪
         ⎪                         ⎩ FLash name nnn  ⎪ ⎪
         ⎪ ⎧ MOdify name [n]                         ⎪ ⎪
         ⎪ ⎪ CHars name1 [name2 [name3 [name4]]]     ⎪ ⎪
         ⎪ ⎨ CHars name1 [CHars name2 [CHars         ⎪ ⎪
         ⎪ ⎪    name3 [CHars name4]]]                ⎪ ⎪
         ⎪ ⎪ FCB name                                ⎪ ⎪
         ⎩ ⎩ FORM form2                              ⎭ ⎭
```

[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, except for NAME which, if specified, must be the last entry in the command line. This is contrary to the notation normally used in this publication.

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 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 alphameric 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 user HOLD status.

`DIST dist`

changes the distribution code specified in the VM/SP 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 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.

6. In order to print on the 3800, the name you specify in the FLASH option must match the FLASH name the VM/SP 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, 12, or (for the 3800 Model 3 only) 10 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 calculates the proper FCB 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 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. The MVS Character Conversion Aid may also be used to convert existing customized character sets to the 3800 Model 3 pel density.

Response

```
nnnn  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.

Messages

```
DMKCSU003E INVALID OPTION - option
DMKCSU006E INVALID DEVICE TYPE - devtype
DMKCSU008E INVALID SPOOLID - spoolid
DMKCSU013E CONFLICTING OPERAND - operand
DMKCSU026E OPERAND MISSING OR INVALID

DMKCSU027E SPOOLID MISSING OR INVALID
DMKCSU028E CLASS MISSING OR INVALID
DMKCSU029E FNAME FTYPE MISSING OR INVALID
DMKCSU030E COPIES MISSING OR INVALID
DMKCSU032E DIST CODE MISSING OR INVALID
```

```
DMKCSU035E DEVICE TYPE MISSING OR INVALID
DMKCSU042E SPOOLID nnnn DOES NOT EXIST
DMKCSU243E FLASH MISSING OR INVALID
DMKCSU245E CHARS MISSING OR INVALID
DMKCSU246E FCB MISSING OR INVALID
DMKCSU253E MODIFY MISSING OR INVALID
```

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:

```
Close    ┌ ┌Reader           ┐                                                        ┐
         │ │vaddr   ┌HOld   ┐│                                                        │
         │ └        └NOHold ┘┘                                                        │
         │                                                                            │
         │ ┌CONsole   ┌PUrge                                                       ┐┐ │
         │ │Printer   │FORM form                                                   ││ │
         │ │PUnch     │                                                            ││ │
         │ │vaddr     │┌HOld   ┐ [DIst distcode] ┌NAme  ┌fn [ft]┐┐                ││ │
         └ └          └└NOHold ┘                 └      └dsname ┘┘                ┘┘ ┘
```

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 5 on page 66). 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 5 on page 66). 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 user 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 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. The form is changed for this file only and does not affect other files.

DIST distcode

uses the one- to eight-character alphameric identification (distcode) on the output separators of printer and punch instead of the identification specified in the VM/SP directory. The distribution code is changed for this file only and does not affect other files or change the VM/SP 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 alphameric name assigned to the file for identification. The filetype, ft, is a one- to eight-character alphameric 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. 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 a 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.

5. You can specify a filename and filetype and an optional distribution code to aid in later identification of the file and its contents.

6. Figure 5 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.

	SPOOL Command Options Set for a Virtual Input Device	
(CLOSE Command Setting)	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

Figure 5. Action Taken on CLOSE for a Virtual Input Device.

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

7. Figure 6 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.

(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]

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

[1]No 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.

Responses

```
{PRT}                {TO }                    {HOLD  }
{PUN} FILE spoolid   {FOR}  userid COPY nnn   {NOHOLD}
{CON}
```

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 USER HOLD status. However, if the SET MSG option is OFF, this message is not issued even though the preceding conditions are met.

Messages

```
DMKCSQ003E INVALID OPTION - option
DMKCSQ006E INVALID DEVICE TYPE - devtype
DMKCSQ013E CONFLICTING OPERAND - operand
DMKCSQ022E VADDR MISSING OR INVALID
DMKCSQ029E FNAME FTYPE MISSING OR INVALID

DMKCSQ032E DIST CODE MISSING OR INVALID
DMKCSQ040E DEV vdev DOES NOT EXIST
```

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 Notes

1. 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 operands 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.

Messages

None.

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.

Messages

```
DMKDIB006E INVALID DEVICE TYPE - devtype
DMKDIB011E INVALID DEVICE TYPE - userid vdev
DMKDIB020E USERID MISSING OR INVALID
DMKDIB022E VADDR MISSING OR INVALID
DMKDIB040E DEV vdev DOES NOT EXIST

DMKDIB045E userid NOT LOGGED ON
DMKDIB047E userid vdev DOES NOT EXIST
DMKDIB058E CTCA vdev BUSY ON userid
```

CP

Privilege Class: Any

Use the CP command as an optional header on commands to CP in a 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 commandline with at least one blank.

Note:

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.

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.

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.

4. An attention key that is pressed while in fullscreen mode (user mode) is not interrogated by CP, but is passed to the fullscreen application for processing. Exceptions to this are the PA1 key and the break key (SCRNSAVE). Therefore, if '#CP command' is entered while in fullscreen mode, the input line will be passed to the application, and the command will not be processed by CP. To execute a CP command while in fullscreen mode, PA1 or the break key must first be used to put the terminal into 'CP read' mode.

Responses

If you enter CP without a commandline in the CP environment, you receive no response.

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

Messages

None.

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
            Printer
            PUnch     [As]  vaddr
            CONsole
            CTCa
            TIMer
            1403
            1443
            2501
            2540P
            2540R
            3088
            3203
            3211
            3262
            3289E
            3505
            3525
            4245

            3800 [As] vaddr [Size ww ll] [42cgm ] [CFS] [NODatck]
                                         [2Wcgm ] [BTS] [DAtck  ]

            [Printer] [As] vaddr    [1403 ]
            [Prt    ]               [1443 ]
            [PRT    ]               [3203 ]
                                    [3211 ]
                                    [3262 ]
                                    [4245 ]
                                    [3289E]
                                    [3800 ]

            CHANnels [As]  [SEL]
                           [BMX]

            LIne     [As] vaddr [IBM[1] ]
                                [TELE[2]]
```

```
GRAF          cuu    ┌3033┐
                     │3036│
                     │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.

`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                                ┌2WCGM┐ ┌BTS┐ ┌DAtck  ┐
3800-1  [As] vaddr [Size ww 11]     └4WCGM┘ └CFS┘ └NODatck┘
3800-3
```

adds a spooling 3800 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 printer is assumed. The vaddr specifies the address to the virtual machine configuration. The SIZE keyword specifies the size of the forms. The ww specifies the width of the form as a hexadecimal code (see Usage Note 6. for the available form width codes.) The 11 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 3800 Model 1 printer, specify:

```
define 3800-1 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. The 2WCGM option is not valid for the 3800 Model 3 printer. 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-Trimmer-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 provided 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 depress the reflection of some load checks that are the result of invalid translation 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 ignores 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 ignores 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: The virtual machine is immediately reset when this set of operands is executed.

```
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 cuu   ┌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 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 5 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.

Redefining any virtual unit record device will result in any open output files being closed and available for further processing as determined by previously issued spooling commands.

A VM/SP 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 OS/VS system generation process because VM/SP 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 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 so that VM/SP 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, 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 via 120 through 12F since these are the addresses predefined in the mass table create program and by the OS/VS 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.

```
TFB-512
T3310     [AS] vaddr [BLK] [nnnnnn]
T3370

T2305
T2314
T2319     [AS] vaddr [CYL] [nnn]
T3330
T3340
T3350
T3375
T3380
```

adds a temporary virtual disk of the specified type to the virtual machine configuration. See Usage Note 5.

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 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] {nnnnnK}
             {nnM   }
```

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 VM/SP 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. Changing the virtual storage size (increasing or decreasing) causes a virtual system reset and clears all virtual storage to binary zeros.

Usage Notes

1. When you alter the configuration or channel operating mode of your virtual machine via the DEFINE command, the changes are temporary and are in effect for the current terminal session only.

2. If either storage or channel operating mode is altered, the virtual machine is immediately reset and the IPL procedure must be performed again.

3. If storage is redefined, any pages that have been locked via the LOCK command become unlocked.

4. If your virtual machine is operating in the 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.

5. Defining a shared and nonshared device on the same virtual control unit will result in an error message being issued (DMKDEF331E). Define shared and nonshared devices on separate subchannels.

6. The figure below 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 7. Available Form Width Codes

7. If the optional keyword "AS" is omitted, the vaddr to be assigned cannot be "A."

8. When using the virtual channel-to-channel adapter, it is possible to receive a spurious attention interrupt after receiving attention plus busy in response to a data transfer operation. The spurious attention may occur if both the X and Y sides of the VCTCA are doing the same data transfer operation (for example, both doing writes or both doing reads).

9. To find the status of the virtual devices that have been defined the QUERY VIRTUAL ALL command can be used.

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.

Messages

```
DMKDEF003E INVALID OPTION - option
DMKDEF022E VADDR MISSING OR INVALID
DMKDEF024E CYLS|BLKSS MISSING OR INVALID
DMKDEF025E STORAGE MISSING OR INVALID
DMKDEF026E OPERAND MISSING OR INVALID

DMKDEF040E DEV vdev DOES NOT EXIST
DMKDEF091E DASD vdev NOT DEFINED; TEMP SPACE NOT AVAILABLE
DMKDEF092E DEV vdev NOT DEFINED; devtype vdev ALREADY DEFINED
DMKDEF095E DEV vdev NOT DEFINED; INSUFFICIENT FREE STORAGE
DMKDEF136E DEVICE {vdev|raddr} NOT {DEFINED|VARIED};
           CHANNEL XX DEDICATED

DMKDEF153E DEVICE {vdev|RNG} EXCEEDED MAX NUMBER OF
           VIRTUAL DEVICES
DMKDEF331E vaddr NOT DEFINED - USE A {SHARED|NONSHARED} VCU
           INSTEAD
DMKDEG025E STORAGE MISSING OR INVALID
DMKDEG026E OPERAND MISSING OR INVALID
DMKDEG094E STORAGE EXCEEDS ALLOWED MAXIMUM

DMKDEI075E DEVICE 'volid' IS NOT AVAILABLE
DMKDEI274E DEV rdev HAS A MOUNTED VOLUME. MSS ID NOT
           AVAILABLE TO DEMOUNT IT
DMKDEI275E rdev ALREADY SYSVIRT
DMKDEI276E rdev NOT 3330V
DMKDEI277E rdev ERROR DEMOUNTING VOLUME

DMKDEI278E DEVICE NOT A DASD
DMKDEI279E DASD NOT VUA
DMKDEI280E rdev ALREADY VIRTUAL
```

DETACH

Privilege Class: G

Use the DETACH command to remove a virtual device from the virtual machine.

The format of the DETACH command is:

```
DETach      { [vaddr [vaddr...]] }
            { [vaddr-vaddr     ] }
```

where:

vaddr [vaddr...]
vaddr-vaddr

the term vaddr is the virtual address (cuu) of the device to be detached from your virtual machine. To detach multiple devices on one command line, a blank must be inserted between the addresses specified. To detach a range of devices, a hyphen (-) must be inserted between the addresses specified. Multiple addresses and a range of addresses cannot be intermixed on the same command line.

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 VM/SP 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.

   Note: A maximum of 48 multiple addresses may be specified on the detach command line. Attempts to detach more than 48 multiple addresses will result in processing only the first 48 addresses requested. The remaining addresses will be ignored, with no error messages issued for these device addresses.

7. It is possible for a class B user to DETACH real devices from another user or other users.

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 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.

```
CTC vaddr DROP FROM userid vaddr
```

This is the response if the device detached was a virtual channel-to-channel device connected (via 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
```

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

Messages

```
DMKNEA020E USERID MISSING OR INVALID
DMKVDD006E INVALID DEVICE TYPE - devtype
DMKVDD020E USERID MISSING OR INVALID
DMKVDD021E RADDR MISSING OR INVALID rdev
DMKVDD022E VADDR MISSING OR INVALID

DMKVDD040E DEV rdev DOES NOT EXIST
DMKVDD045E userid NOT LOGGED ON
DMKVDD046E devtype rdev OFFLINE
DMKVDD121E devtype rdev NOT ATTACHED TO userid
DMKVDD123E DASD rdev CP OWNED

DMKVDD124E DASD rdev IN USE BY nnn USERS
DMKVDD135E devtype rdev NOT DETACHED; CHAN x DEDICATED
DMKVDD140E devtype rdev ATTACHED TO userid
DMKVDD142E devtype rdev NOT DRAINED
DMKVDD143E devtype rdev IN USE BY SYSTEM
DMKVDD273E DETACH REJ VIRTUAL MACHINE TRACING I/O
```

DETACH CHANNEL

Privilege Class: G

Use the DETACH CHANNEL command to detach 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 CHANNEL command is:

`DETach`	`CHANnel c`

where:

`CHANNEL c`
is the real address of the channel that is to be detached.

Usage Note

If DETACH CHANNEL is invoked prior to the normal completion of I/O activities on the dedicated channel, those activities are aborted.

Response

`CHANNEL c DETACHED userid`
is sent to the user who issued the command and to the primary system operator.

Messages

```
DMKVCH034E CHANNEL MISSING OR INVALID
DMKVCH048E CHANNEL x DOES NOT EXIST
DMKVCH130E CHANNEL x [PROC xx] NOT ATTACHED TO userid
DMKVCH138E CHANNEL x NOT ATTACHED; INSUFFICIENT STORAGE
DMKVCH188E SPECIFIED PROCESSOR UNAVAILABLE

DMKVDD020E USERID MISSING OR INVALID
DMKVDD034E CHANNEL MISSING OR INVALID
DMKVDE039E PROCESSOR ADDRESS MISSING OR INVALID
```

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.

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 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 locally and remotely attached BSC 3270 display devices. If the DIAL command is issued from a local 3270 terminal, the virtual system user must issue the CP command RESET to drop the dialed connection.

   Additionally, 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 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.

Responses

```
DIALED TO userid vaddr
```

is the message sent to the user indicating that a logical connection has been made.

```
{GRAF raddr}
{LINE raddr} DIALED TO userid DIALED = nnn
{DEV  resid}
```

is the response to the primary system operator. It indicates a successful connection to the virtual machine (userid) and the total number of VM/SP lines (nnn) currently connected to other virtual machines. 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.

```
{GRAF raddr}
{LINE raddr} DROP FROM userid DIALED = nnn
{DEV  resid}
```

is the message sent to the primary system operator when the line is disabled.

Messages

```
DMKDIA011E INVALID DEVICE TYPE - userid vdev
DMKDIA017E INSUFFICIENT SCREEN SPACE - CLEAR SCREEN AND REDIAL
DMKDIA020E USERID MISSING OR INVALID
DMKDIA022E VADDR MISSING OR INVALID
DMKDIA045E userid NOT LOGGED ON

DMKDIA047E userid vdev DOES NOT EXIST
DMKDIA055E LINE(S) NOT AVAILABLE ON userid
DMKDIA056E LINE vdev BUSY ON userid
DMKDIA098E DEV rid MODE SWITCH NOT POSSIBLE
DMKDIA199E COMMAND REJECTED; DEVICE rdev PENDING OFFLINE

DMKDIA206E CANNOT CONNECT TO HOST VITUAL MACHINE
DMKDIA707A DIAL FUNCTION NOT AVAILABLE
```

DISCONN

Privilege Class: Any

Use the DISCONN command to disconnect your terminal from the VM/SP 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. If specified, control returns to CP and the "VM/370 online" message is displayed.

Usage Notes

1. The terminal remains disconnected until it is reconnected via 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 unless SET RUN ON has been previously specified. If you log on within 15 minutes, your virtual machine can continue operating. Since many operands of the SET and TERMINAL commands default to OFF when you log on again, you have to reissue the SET and TERMINAL commands. If you do not log on 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 via 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. Local console options 'APL', 'ATTN', 'CONMODE 3270', 'HILIGHT', 'LINESIZE', 'SCRNSAVE', 'SCROLL', 'TEXT', and 'TYPE 3101', for the TERMINAL command are session valid options only, and are not supported across DISCONNECT of the virtual machine. These options must be reset, if desired, upon reconnect.

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.

```
{GRAF raddr}
{LINE raddr} DISCONNECT userid USERS = nnn
{DEV  resid}
```

is the response to the primary system operator informing him that the user represented by userid has been disconnected from the VM/SP system. The response, nnn, is the total number of users remaining in the system. DEV resid indicates the resource identification of a 3704/3705/3725 line.

Messages

```
DMKUSO003E INVALID OPTION - option
DMKUSO197E SECONDARY USER IS NOT AVAILABLE
```

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 Notes")
- 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 content of 3704/3705/3725 storage.

The format of the DISPLAY command is:

```
Display  { [ hexloc1]   {-}  [hexloc2]                 }
         { [Khexloc1]   {:}  [END    ]                 }
         { [Lhexloc1]                                  }
         { [Thexloc1]                                  }
         {              {.}  [bytecount]               }
         {                   [END      ]               }
         {                                             }
         { Greg1     [{-}  [reg2]            ]         }
         { Yreg1     [{:}  [END ]            ]         }
         { Xreg1     [                       ]         }
         {           [{.}  [regcount]        ]         }
         {           [     [END     ]        ]         }
         {                                             }
         { Psw                                         }
         { CAW                                         }
         { CSW                                         }
```

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 the following "Usage Notes" for a discussion on discontiguous saved segments.) The address, 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 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 the following "Usage Notes" for a discussion on discontiguous saved segments.) If this address is not on a fullword boundary, it is rounded up to the next higher fullword. The value, bytecount, must have a value of at least one and may be from one to six hexadecimal digits; leading zeros are optional.

`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.

{.} regcount
END

is a decimal number from 1 to 15 or a hexadecimal number from 1 to F 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 15 or a hexadecimal number from 1 to F. If display type Y is specified, regcount must be 1, 2, 3, or 4. The sum of reg1 and regcount must be a number that does not exceed the maximum register number for the type of registers being displayed.

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

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 (via the DISPLAY command) at the terminal.

8. When running with a discontiguous 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.

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  .xxxxxxxxxxxxxxxxx  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.

Messages

```
DMKCDB003E INVALID OPTION - option
DMKCDB004E INVALID HEXLOC- hexloc
DMKCDB009E INVALID RANGE - range
DMKCDB010E INVALID REGISTER - register
DMKCDB026E OPERAND MISSING OR INVALID
DMKCDB160E HEXLOC hexloc EXCEEDS STORAGE
```

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 VM/SP directory entry)
- Storage keys
- Virtual storage locations (second-level virtual storage only; see "Usage Notes")

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

The format of the DUMP command is:

```
DUmp    [Lhexloc1] { {-}  hexloc2       }     [*dumpid]
        [Thexloc1] { {:}  END           }
        [ hexloc1] {                    }
                   { {.}  [bytecount]   }
                   {      [END      ]   }
```

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.

```
{-} hexloc2
{:} END
```

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 VM/SP 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 discontiguous 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.

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.

Messages

```
DMKCDM004E INVALID HEXLOC - hexloc
```

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]
            [1 ]
```

where:

nn is the number of times that the line is to be sent.

1 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.

Messages

None.

EXTERNAL

Privilege Class: G

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

The format of the EXTERNAL command is:

EXTernal	[code 40]

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 VM/SP directory, you can also code X'1004' (clock comparator interrupt) or X'1005' (CPU timer interrupt). The default is the external interrupt button on the system console, X'40'.

Usage Note

This simulates pressing the interrupt key on the real computer console, or other functions which cause an external interrupt. 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.

Messages

```
DMKCPB005E INVALID HEXDATA - data
```

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 of VM/SP 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%
└PROC xx-nn%  PROC yy-nn%┘

EXPAN-nnn  PAGING-nnnn/SEC  STEAL-nnn%  LOAD-nnn%
```

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, VM/SP indicates this response field as zero.

`PROC xx-nn%`
(in MP mode only) nn 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-nn%`
(in MP mode only) nn 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 System Programmer's Guide.*

`PAGING-nnnn/SEC`
indicates the average number of page reads and writes performed per second.

`STEAL-nnn%`
indicates the percentage of page reads that required a real storage page to be stolen from another in-queue virtual machine.

`LOAD-nnn%`
is an artificial value. It attempts to measure the elapsed time as a percentage of the system time devoted to paging that results from real storage contention. This calculation includes the estimated processor

time involved in paging, the amount of time spent in page waits, and the percentage of pages stolen (STEAL).

Response for INDICATE USER

```
PAGES: RES-nnnn WS-nnnn READS=nnnnnn WRITES=nnnnnn MH-nnnn FH-nnnn
VTIME=mmm:ss TTIME=mmm:ss SIO=nnnnnn RDR-nnnnnn
PRT-nnnnnn PCH-nnnnnn
```

where:

`RES-nnnn` is the current number of your virtual storage 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 working set size.

`READS=nnnnnn`
is the total number of page reads that have occurred for you since you logged on or since the last ACNT command was issued for your virtual machine.

`WRITES=nnnnnn`
is the total number of pages written for you since you have logged on or since the last ACNT command was issued for your virtual machine.

`MH-nnnn` is the current number of virtual pages allocated for you in the preferred moveable head paging area of the system owned paging disk(s). This number is taken at an instant in time during the execution of the INDICATE command.

`FH-nnnn` is the current number of virtual pages allocated for you in the preferred fixed head paging area of the system owned paging disk(s). This number is taken at an instant in time during the execution of the INDICATE command.

`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) that you have used since you logged on or since the last ACNT command was issued for your virtual machine.

`SIO=nnnnnn`
is the total number of nonspooled I/O requests that you have issued since you logged on or since the last ACNT command was issued for your virtual machine. 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.

RDR-nnnnnn

is the total number of virtual cards read since you logged on or since the last ACNT command was issued from your virtual machine.

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.

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.

Messages

```
DMKTHI002E INVALID OPERAND - operand
DMKTHI003E INVALID OPTION - option
DMKTHI026E OPERAND MISSING OR INVALID
DMKTHI045E userid NOT LOGGED ON
```

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 ⎡cylno ⎤ ⎡CLear   ⎤ [STOP] [ATTN] ⎫ [PARM p1 [p2...]]
      ⎨       ⎣nnnnnn⎦ ⎣NOCLear ⎦               ⎬
      ⎩ systemname                              ⎭
```

where:

```
vaddr ⎡cylno ⎤ ⎡CLEAR   ⎤ [STOP]
      ⎣nnnnnn⎦ ⎣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.

cylno
: is the cylinder containing the IPL data. If this operand is specified, CP loads the IPL data from the specified virtual cylinder instead of from virtual cylinder zero. Virtual cylinder zero is the default when cylno is not specified. This operand is valid only for virtual direct storage devices.

nnnnnn
: is the block address containing the IPL data. Virtual block zero is the default when nnnnnn is not specified. This operand is valid only for FB-512 devices.

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 operand is invalid if 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' operand on the IPL command.

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 System Programmer's Guide.*

Note: You cannot load a shared system or one that uses discontiguous 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 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 embedded 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 PARM is specified, the remaining characters on the command line are treated as parameters to be passed to your virtual machine; therefore, 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. 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.

2. 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.

3. 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.

4. 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.

5. 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.

6. 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.

7. 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.

8. CMS users can use the PARM operand to pass the AUTOCR (automatic carriage return) option to the CMS initialization routine. AUTOCR 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.

9. AUTOCR should not be used for AUTOLOG users if SET AUTO READ OFF is used in the user's profile.

   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 AUTOCR command, the PROFILE EXEC executes initial setup procedures that allow the service virtual machine to reestablish itself.

10. 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.

11. When CMS is IPLed by name and the virtual storage size is too small to contain all of the saved CMS system pages, the CMS saved system is treated by CP as a discontiguous 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.

12. When IPL'ing the loader from your virtual machine in order to create a CP nucleus, CONMODE should be set 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.

Messages

```
DMKCFG002E INVALID OPERAND - operand
DMKCFG003E INVALID OPTION - option
DMKCFG013E CONFLICTING OPTION - option
DMKCFG022E VADDR MISSING OR INVALID
DMKCFG026E OPERAND MISSING OR INVALID

DMKCFG040E DEV vdev DOES NOT EXIST
DMKCFG044E SYSTEM sysname DOES NOT EXIST
DMKCFG083E VMSAVE AREA DOES NOT CONTAIN A VALID SAVED SYSTEM
DMKCFG170E SYSTEM sysname EXCEEDS STORAGE
DMKCFG171E SYSTEM sysname VOLID volid NOT MOUNTED

DMKCFG172E SYSTEM sysname DASD vdev INCOMPATIBLE SYSRES
DMKCFG173E SYSTEM sysname REQUIRES DASD vaddr
DMKCFG174E PAGING I/O ERROR; IPL FAILED
DMKCFG177E PARM EXCEEDS 64 CHARACTERS
DMKCFG203E V=R AREA REQUIRES IPL BY DEVICE ADDRESS
DMKCLK437E SYSTEM PAGING SPACE NOT AVAILABLE, IPL TERMINATED

DMKCLK970W TOD CLOCK SYNC CHECK RECEIVED
DMKCPJ329E rdev IMPROPERLY GENNED, DEVICE CANNOT BE USED AS
           SYSOWN
DMKCPJ330I MISMATCHED DEVICE TYPES AT ADDRESS rstor
DMKVMI022E VADDR MISSING OR INVALID
DMKVMI230E IPL SIO ERROR

DMKVMI231E IPL TIO ERROR
DMKVMI232E IPL UNIT ERROR; CSW csw SNS sense
DMKVMI233E IPL OVERLAY ERROR
DMKVMI234E IPL SENSE ERROR
```

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 VM/SP directory entry.

The format of the LINK command is:

`LINK`	`[To] userid vaddr1 [As] vaddr2 [mode] [[PASS=] password`[1]`]`

[1] If 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 VM/SP directory is to be searched for device vaddr1. An asterisk (*) is used to specify that the device is in your own VM/SP directory. If the keyword TO is omitted, the userid may not be "TO" or "T."

`vaddr1`

is the virtual device address (cuu) in the VM/SP directory 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 "AS" or "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. See Usage Note 6.

`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 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 a read-only link is given.

M Primary multiple access. A write link is established unless another user already 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 already has write access to the disk, in which case a read link is given since it was the alternate access requested.

Note: Unpredictable results can occur when one user has a read-only (R or RR) link to a device that is being updated by a user who has the device in write status (W or WR).

MW Primary multiple access or alternate write access. A write link is established in all cases.

Caution: 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. 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 and the userid is that of another user, then the default mode is R. If you are linking to one of your own disks, the default mode is the access mode as specified in the VM/SP directory for your disk (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 VM/SP 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. Up to three access mode passwords can exist for each disk; these are the read, write, and multiple passwords.*
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. 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 is not guaranteed by CP or CMS.
2. 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 password must be provided. If a LINK command is issued from a CMS batch job, a password must be provided even if the access password is ALL.
3. 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.
4. Virtual reserve/release processing cannot be requested by appending a V to the mode operand of the LINK command. Virtual reserve/release processing can only be requested by means of the MDISK statement of the directory program.

5. The link established by the LINK command to the vaddr1 device belonging to userid is summarized in the following chart where R = read-only, W = write, M = multiple, and - = none:

Primary access requested	R						W						M								
Alternate access requested	-			R			-			R			-			R			W		
Current existing links	-	R	W	-	R	W	-	R	W	-	R	W	-	R	W	-	R	W	-	R	W
LINK ESTABLISHED	R	R	-	R	R	R	W	-	-	W	R	R	W	W	-	W	W	R	W	W	W
where: R=read; W=write; M=multiple																					

Figure 8. Illustration of the Type of Link Established Under Every Condition

6. When issuing the LINK command be aware that VM/SP 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 so, the command is not executed and an error message is sent. Refer to Appendix B in the *VM/SP 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 PASSWORD:
XXXXXXXX
```

Type the read password over the mask to request read-only access to the desired disk. The mask will only appear on line-mode terminals.

```
ENTER WRITE PASSWORD:
XXXXXXXX
```

Type the write password over the mask to request write access to the desired disk. The mask will only appear on line-mode terminals.

```
ENTER MULT PASSWORD:
XXXXXXXX
```

Type the multiple password over the mask to obtain write access to a disk to which other users may already have access. The mask will only appear on line-mode terminals.

Note: CP keeps track of unsuccessful link attempts due to invalid passwords. When a user exceeds the maximum number of incorrect password attempts allowed (a system threshold value which defaults to 10), the LINK command is disabled for the remainder of the terminal session.

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.

```
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   } [        {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   } [        {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 with virtual devices that have the same starting cylinder or block on the same real device.

Messages

```
DMKLNK020E USERID MISSING OR INVALID
DMKLNK022E VADDR MISSING OR INVALID
DMKLNK052E ERROR IN CP DIRECTORY
DMKLNK053E userid NOT IN CP DIRECTORY
DMKLNK101W DASD vdev FORCED R/O; R/O by nnn USERS

DMKLNK102W DASD vdev FORCED R/O; R/W by nnn USERS
DMKLNK103W DASD vdev FORCED R/O; R/W by nnn USERS;
           R/O BY nnn USERS
DMKLNK104E userid vdev NOT LINKED; R/O BY nnn USERS
DMKLNK105E userid vdev NOT LINKED; R/W BY nnn USERS
DMKLNK106E userid vdev NOT LINKED; R/W BY nnn USERS;
           R/O BY nnn USERS

DMKLNK107E userid vdev NOT LINKED; NOT IN CP DIRECTORY
DMKLNK108E userid vdev NOT LINKED; VOLID volid NOT MOUNTED
DMKLNK109E userid vdev NOT LINKED; INVALID LINK DEVICE
DMKLNK110E userid vdev NOT LINKED; devtype vdev ALREADY DEFINED
DMKLNK111E userid vdev NOT LINKED; NO READ PASSWORD

DMKLNK112E userid vdev NOT LINKED; NO WRITE PASSWORD
DMKLNK113E userid vdev NOT LINKED; NO MULT PASSWORD
DMKLNK114E userid vdev NOT LINKED; PASSWORD INCORRECT
DMKLNK115E LINK INVALID; EXCESSIVE INCORRECT PASSWORDS
DMKLNK116E USERID xxx NOT LINKED; CP DIRECTORY IN USE

DMKLNK117E USERID userid NOT LINKED; VOLID volid CONFLICT
DMKLNK118E userid rdev NOT LINKED; COMMAND FORMAT NOT VALID
DMKLNK119E USERID vdev NOT LINKED; INSUFFICIENT FREE STORAGE
DMKLNK137E DASD vdev NOT LINKED; CHAN x DEDICATED
DMKLNK139E DASD rdev NOT LINKED; 3330V NOT ACCEPTED AS SYSTEM
           VOLUME

DMKLNK153E DEVICE {vdev|RNG} EXCEEDED MAX NUMBER OF VIRTUAL
           DEVICES
DMKLNK199E COMMAND REJECTED; DEVICE rdev PENDING OFFLINE
DMKLNK331E vaddr NOT LINKED - USE A {SHARED|NONSHARED} VCU INSTEAD
DMKSSS070E DEVICE NOT AVAILABLE - MSS DEVICES ALLOCATED
DMKSSS071E DEVICE NOT AVAILABLE - MSS CANNOT BE ACCESSED

DMKSSS073E DEVICE NOT AVAILABLE - MSS ERROR
DMKSSS078I MSS MOUNT PROCEEDING FOR VOLUME 'volid'
```

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 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 and 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 VM/SP 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.

2. VM/SP provides two FCB images (FCB1 and FCB8) with the following format:

Line spacing = 6 lines/inch						FCB1						
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 that replaces 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 can 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 loaded into the real printer according to the FILEFCB, CFILEFCB, or DEFFCB option specified on the start command. The spool file will then print properly on the real printer.

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.

Messages

```
DMKCSB006E INVALID DEVICE TYPE - devtype
DMKCSB022E VADDR MISSING OR INVALID
DMKCSB026E OPERAND MISSING OR INVALID
DMKCSB031E BUFFER MISSING OR INVALID
DMKCSB031E BUFFER MISSING OR INVALID: FCB TOO LONG

DMKCSB031E BUFFER MISSING OR INVALID:
           EXTENDED FCB NOT SUPPORTED
DMKCSB036E INDEX MISSING OR INVALID
DMKCSB040E DEV rdev DOES NOT EXIST
DMKCSB043E BUFFER name DOES NOT EXIST
```

LOGOFF

Privilege Class: Any

Use the LOGOFF command to terminate a virtual machine session and disconnect your virtual machine from the VM/SP system.

The format of the LOGOFF command is:

LOGoff LOGout	[HOld]

where:

HOLD retains the connection for a switched communication line to enable you to log on without redialing the VM/SP system.

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 log off 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 discovers that the terminal has been turned off and then only if a read is outstanding (that is, VM/SP attempts to send a message to the terminal, but gets back an error code indicating that the terminal is turned off). Many hours may pass before VM/SP discovers 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.hs 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 CPU time used in the current terminal session in minutes:seconds.hundredths of seconds.

`TOTCPU= mmm:ss.hs`
the total CPU 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.

The 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.

```
{GRAF raddr}
{LINE raddr} LOGOFF AS userid USERS = nnn
{DEV rid   }
```

is the normal response to the primary system operator. DEV rid specifies the resource identification of a 3704/3705/3725 line.

```
{GRAF raddr}
{LINE raddr} LOGOFF AS userid USERS = nnn FORCED
{DEV  rid  }
```

is the response to the primary system operator if the logoff is forced by a line timeout or a terminal power-off. DEV rid 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.

Messages

```
DMKUSO003E INVALID OPTION - option
```

LOGON

Privilege Class: Any

Use the LOGON command to identify yourself to the VM/SP system and to access that system.

The format of the LOGON command is:

```
Logon      userid   [password]   [Noipl]
Login
```

where:

`userid` is the identifier assigned to you in the VM/SP 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 VM/SP 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 system and not to some other user's virtual machine.

   If your terminal is a SNA terminal supported through the Virtual Machine/VTAM Communications Network Application (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 system with a new session.

2. Upon successful logon, VM/SP 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 log on again. If you do not log on 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 VM/SP system generated for attached processing operation but operating in uniprocessor mode, a message is sent to the user with attached processor AFFINITY set (via 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. 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 attempt by the user will result in the VM logo being displayed.

7. CP automatically masks your password on typewriter terminals to insure installation security. For installations migrating from a VM/370 operating system to VM/SP, 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.

Responses

```
ENTER PASSWORD  (IT WILL NOT APPEAR WHEN TYPED):
```

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.

```
Enter one of the following commands:

LOGON userid         (Example: LOGON VMUSER1)
DIAL userid          (Example: DIAL VMUSER2)
MSG userid message   (Example: MSG VMUSER2 GOOD MORNING)
   LOGOFF
```

is issued immediately following any of the messages:

```
DMKLOG003E INVALID OPTION - option
DMKLOG020E USERID MISSING OR INVALID
DMKLOG050E LOGON UNSUCCESSFUL -- INCORRECT PASSWORD
DMKLOG052E ERROR IN CP DIRECTORY
DMKLOG053E userid NOT IN CP DIRECTORY

DMKLOG054E ALREADY LOGGED ON {LINE|GRAF|LUNAME} raddr
DMKLOG067E COMMAND FORMAT NOT VALID
```

```
{luname     }
{GRAF raddr }
{LINE raddr } LOGON AS userid     USERS = nnn
{DEV  resid }

     -- or --

{luname     }
{GRAF raddr }
{LINE raddr } RECONNECT userid    USERS = nnn
{DEV  resid }
```

is the response to the primary system operator. DEV resid specifies the resource identification of a 3704/3705/3725 line. An luname response specifies the logical unit name of a user who logs on to VM/SP through a SNA network.

Messages

```
DMKLOG003E INVALID OPTION - option
DMKLOG020E USERID MISSING OR INVALID
DMKLOG050E LOGON UNSUCCESSFUL -- INCORRECT PASSWORD
DMKLOG052E ERROR IN CP DIRECTORY
DMKLOG053E userid NOT IN CP DIRECTORY

DMKLOG054E ALREADY LOGGED ON {LINE|GRAF|LUNAME} rdev
DMKLOG067E COMMAND FORMAT NOT VALID
DMKLOG080E NO PREALLOCATED VMSAVE AREA
DMKLOG081W VMSAVE AREA ALREADY IN USE, MUST USE SET COMMAND
DMKLOG082E MORE THAN ONE VMSAVE AREA ALLOCATED,
                     MUST SPECIFY AREA NAME

DMKLOG090E DEV vdev NOT DEFINED; DEV {rdev|volid} NOT AVAILABLE
DMKLOG091E DASD vdev NOT DEFINED; TEMP SPACE NOT AVAILABLE
DMKLOG092E DEV vdev NOT DEFINED; devtype vdev ALREADY DEFINED
DMKLOG093E DEV vdev NOT DEFINED; ERROR IN CP DIRECTORY
DMKLOG095E DEV vdev NOT DEFINED; INSUFFICIENT FREE STORAGE

DMKLOG131E INSUFFICIENT FREE STORAGE AVAILABLE FOR LOGON REQUEST
DMKLOG188E SPECIFIED PROCESSOR UNAVAILABLE
DMKLOG189E AFINITY NOT SET; SYSTEM IN UNIPROCESSOR MODE
DMKLOG250E 370E IS NOT AVAILABLE {ON PROC xx}
DMKLOG251E 370E IS NOT ACTIVE

DMKLOH196E SECONDARY USER IS NOT IN CP DIRECTORY
DMKLOH206E CANNOT CONNECT TO HOST VIRTUAL MACHINE
DMKSSS070E DEVICE NOT AVAILABLE - MSS DEVICES ALLOCATED
DMKSSS071E DEVICE NOT AVAILABLE - MSS CANNOT BE ACCESSED
DMKSSS072E DEVICE NOT AVAILABLE - 'rdev' NOT PROPER TYPE

DMKSSS073E DEVICE NOT AVAILABLE - MSS ERROR
DMKSSS078I MSS MOUNT PROCEEDING FOR VOLUME 'volid'
```

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:

```
Message  | userid    msgtext
Msg      | *
         | OPerator
```

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.

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. 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, DMKMSG045E '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.

Responses

```
hh:mm:ss
MSG FROM OPERATOR:  msgtext
```

is the message received by the user from the system operator.

```
hh:mm:ss
MSG FROM {LOGONxxx}: msgtext
         {userid  }
         {LOGNxxxx}
```

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, LOGON or LOGN 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.

Messages

```
DMKMSG003E INVALID OPTION - option
DMKMSG020E USERID MISSING OR INVALID
DMKMSG045E userid NOT LOGGED ON
DMKMSG057W userid NOT RECEIVING; {DISCONNECTED|MSG OFF|NOT
           DISC|NO CONSOLE|SMSG OFF| NOT AUTHORIZED|WNG
           OFF|VMCF/IUCV ERROR nnn}
```

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.

Messages

```
DMKCPB006E INVALID DEVICE TYPE - devtype
DMKCPB022E VADDR MISSING OR INVALID
DMKCPB040E DEV vdev DOES NOT EXIST
```

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     (Reader )   (CLass c1 CLass c2...     )   1
          (Printer)   (FORM form1 FORM form2... )
          (PUnch  )   (spoolid1 spoolid2...     )
```

[1] A combination of CLASS, FORM, and spoolid specifications may be entered.

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.

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.

Messages

```
DMKCSV003E INVALID OPTION - option
DMKCSV006E INVALID DEVICE TYPE - devtype
DMKCSV008E INVALID SPOOLID - spoolid
DMKCSV024E FORM NUMBER MISSING OR INVALID
DMKCSV026E OPERAND MISSING OR INVALID

DMKCSV027E SPOOLID MISSING OR INVALID
DMKCSV028E CLASS MISSING OR INVALID
DMKCSV035E DEVICE TYPE MISSING OR INVALID
DMKCSV042E SPOOLID nnn DOES NOT EXIST
```

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 the user's 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 SECTION 3. CP COMMAND USAGE under *Testing and Debugging of Programs.*

The format of the PER command is:

```
PER         EVENT TYPES:

              [ Instruct   [DATA] hex-data]                             [options] ]
              | 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]   ]                         [       ] ]

            OPTIONS:

                Range instruction-addr-range
                FRom instuction-addr-range

                PAss [0]
                     [n]

                CMd{text}

                [ Printer      [RUN]                    ]
                |                                       |
                | [TErminal] [ NORun       ]            |
                | [BOth    ] | RUN         |            |
                |            |             |            |
                |            | STEp [1]    |            |
                [            [      [n]    ]            ]

                 GUESTR
                 GUESTV
                 DATOFF
                 DAT

            SUB-COMMANDS:

                COunt
                TAble
                SAve    traceset-name [APpend]
                GET     traceset-name [APpend]
                ENd    { ALL            }
                       { COunt          }
                       { CUrrent        }
                       { element-number }
                       { event-type     }
                       { traceset name  }
```

where:

EVENT-TYPES

The PER command supports 5 different EVENT keywords. More than one EVENT keyword can be placed on a single line and they can be 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 the DATA keyword is specified, then hex-data must be specified. Hex-data must be expressed in complete bytes. If hex-data is not specified, 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 the INTO keyword is specified, then an into-addr-range must be specified.

The into-addr-range is optional. See Usage Note 1 for the various ways that this address range can be expressed. If an into-addr-range is not specified, then all successful branch instructions are displayed. If an into-addr-range is specified (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 the INTO keyword is specified, then an address range must be specified. If the DATA keyword is specified, then hex-data must be specified.

If no operands are given, then all instructions causing alterations to storage are traced.

If the storage-addr-range is specified, trace output will be produced only when an instruction alters storage in the indicated range.

If addr hex-data is specified, then addr must be 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 storage at the indicated address is not available, 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. This value can be expressed in either decimal or hexadecimal. If no register number is supplied, then the default is all 16 general purpose registers. If the register number (reg1) is specified, 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 reg2 is specified, then all the registers from reg1 to reg2 will be monitored. reg2 must be greater than or equal to reg1. If regcount is specified, 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 the DATA keyword is specified, then hexword must be specified. 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
```

trace register 12, 13, 14, or 15 if any of their values change to 00000006

OPTIONS

The PER command supports 14 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 for the various ways that the range can be specified. 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 RANGE is not specified, then the full range of virtual storage is assumed. If the RANGE keyword is specified, then the instruction-addr-range is required, and follows the RANGE keyword. The RANGE keyword can be specified 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 multiple ranges are specified in this case, then each range will apply to all the event types on that command line. The maximum number of ranges that can be specified 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.

The RANGE keyword can be specified on a command line without 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 the section of the PER command on *Changing Options on Trace Elements.*

The RANGE keyword can also be used 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.

Setting up multiple address stops with PER is accomplished 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` $\begin{bmatrix}\underline{0}\\ n\end{bmatrix}$

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. PASS cannot be specified more than once on a command line.

`CMd {text}`

specifies a CP command or commands to be executed when the given event occurs. If the CMD option is used, 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.

The SLEEP command should not be used with the CMD option. (See Usage Note 23.)

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 and cannot be specified on the same command line unless TERMINAL is also specified. If PRINTER is specified for any trace elements where NORUN or STEP is already in effect, then the PRINTER option is applied to that trace element and the RUN option is turned on.

Note: "BOTH and NORUN" and "BOTH and STEP" are valid combinations. If BOTH is specified on the command line, PRINTER and TERMINAL appear in the QUERY display.

```
RUN
NORun
```

```
STEp ┌ 1 ┐
     └ n ┘
```

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 n is not specified, then 1 is the default value for n. STEP may be turned off by specifying either the

RUN or NORUN option. STEP cannot be specified more than once on a command line.

STEP, RUN, and NORUN are conflicting options and cannot be specified on the same command line.

"NORUN and PRINTER" or "STEP and PRINTER" are an invalid combination of options and cannot be specified on the same command line unless TERMINAL is also specified. If NORUN or STEP is specified for any trace element where 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.

GUESTR
GUESTV
DATOFF
DAT

PER traces virtual machine activity in both second and third level storage. (See usage note 21.) 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.

GUESTV (or DAT) causes only third level activity to be traced.

If both GUESTR and GUESTV are specified, or if neither are 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.

SUB-COMMANDS

The PER command supports 5 SUB-COMMAND keywords. Only one sub-command can be specified on a command line. A sub-command must be the first keyword following PER on the command line, and cannot be specified on a command line that contains event or option keywords.

COunt

causes the successful events to be counted. While the PER COUNT option is in effect, no trace output is produced. Whenever a valid PER command is entered, the value of the current count is displayed and then the PER COUNT is terminated.

If COUNT is in effect and PER COUNT is entered, 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
APPEND
BRANCH
COUNT
CURRENT
G
INSTRUCT
MASK
NAMES
STORE
```

If APPEND is specified, then the current traceset is merged with the named saved traceset. (See the section of the PER command on *Merging Trace Elements.*)

If APPEND is not specified, 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 the following PER command is issued:

```
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 APPEND is specified, then the named traceset is merged with the current traceset. (See the section of the PER command on *Merging Trace Elements.*)

If APPEND is not specified, 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. More than one operand can be specified 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 (e.g. 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 the following is entered:

```
   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 any of the options (RANGE, FROM, PASS, CMD, PRINTER, TERMINAL, BOTH, RUN, NORUN, STEP, GUESTR, GUESTV, DATOFF or DAT) are specified 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 the command:

```
per range 24000-25000 pass 3 printer run
```

is issued, 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
```

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 the command:

```
per store 23034-23100 range 20000-24000 pass 4
```

is issued, 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. All address range values must be specified in hexadecimal. The address range can be specified 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 no range is specified, 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 interruptible 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. 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.

6. 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

7. 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 (i.e. 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.

8. 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 may also be displayed.

9. The default range 0-FFFFFF or its equivalent (for example, RANGE 5-4) is not displayed in the QUERY PER output even if explicitly specified in the PER command. The default PASS 0 is not displayed in the QUERY PER output even if explicitly specified.

10. 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.

11. 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.

12. 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.

13. 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.

14. In order to use the line end character for multiple commands, line edit must be on before issuing the PER command.

    When multiple commands have been entered properly, then QUERY PER always includes the current line end character in the display of multiple commands.

15. 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.

16. 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.
```

17. 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.

18. PASS suppresses the display of successful events. PASS *does not* make the event unsuccessful. Therefore, the PASS option has no effect on the count.

19. 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 the command:

    ```
    per i r 20000-22000 r 22000-24000 terminal norun
    ```

    is issued, 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
    ```

    and the command

    ```
    per i r 20000-21000 r 21000-22000 terminal norun
    ```

    is issued, then the resulting current traceset would be:

    ```
    1 INSTRUCT RANGE 020000-022000 TERMINAL RUN
    2 INSTRUCT RANGE 020000-022000 TERMINAL NORUN
    ```

    See Usage Note 5.

20. 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.

21. 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 (for example, MVS), then the addresses are third level addresses.

22. 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.

23. The SLEEP command should not be used 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.

24. 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
tttttt	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 tttttt 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 xxxxxxxxxxxx oooooo oooooo CC=n
```

EXECUTE INSTRUCTIONS:

```
lvvvvvv EX    xxxxxxxx yy (instruction output as above)
```

BRANCH TRACEBACK:

TRACEBACK TABLE:

```
:lvvvvvv mnem  xxxxxxxx tttttt  n TIMES
:(up to 6 entries are included)
```

Note: Entries are put into the bottom of the traceback table. When the table is full, the new entry is put in 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 hex-data is specified 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.

Messages

```
DMKPEI002E INVALID OPERAND - operand
DMKPEI003E INVALID OPTION - option
DMKPEI010E INVALID REGISTER - register
DMKPEI026E OPERAND MISSING OR INVALID
DMKPEI047E THE CURRENT TRACESET DOES NOT EXIST

DMKPEI312E MORE THAN {8 RANGES|1 PASS|1 STEP} SPECIFIED
           ON COMMAND LINE
DMKPEL013E CONFLICTING OPTION - option1 AND option2
DMKPEL013W CONFLICTING OPTION-{NORUN CONFLICTS WITH PRINTER|PRINTER
           CONFLICTS WITH NORUN|STEP CONFLICTS WITH PRINTER|PRINTER
           CONFLICTS WITH STEP} - {''BOTH''|''RUN''} OPTION FORCED
DMKPEL047E THE CURRENT TRACESET DOES NOT EXIST
DMKPEL311E MAXIMUM NUMBER OF TRACE ELEMENTS EXCEEDED

DMKPEN002E INVALID OPERAND - operand
DMKPEN003E INVALID OPTION - option
DMKPEN026E OPERAND MISSING OR INVALID
DMKPEN047E {THE CURRENT TRACESET|THE TRACESET traceset} DOES NOT
           EXIST
DMKPEN180E PER TRACE NOT IN EFFECT

DMKPEN311E MAXIMUM NUMBER TRACE ELEMENTS EXCEEDED
DMKPET003E INVALID OPTION - option
DMKPET180E PER TRACE NOT IN EFFECT
```

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    [ CLass c1 CLass c2...     ] 1 }
        { Printer   [ FORM form1 FORM form2... ]   }
        { PUnch     [ spoolid1 spoolid2...     ]   }
        { ALL       [ ALL                      ]   }
```

[1] A combination of CLASS, FORM and spoolid 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).

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.

Messages

```
DMKCSV003E INVALID OPTION - option
DMKCSV006E INVALID DEVICE TYPE - devtype
DMKCSV008E INVALID SPOOLID - spoolid
DMKCSV024E FORM NUMBER MISSING OR INVALID
DMKCSV026E OPERAND MISSING OR INVALID

DMKCSV028E CLASS MISSING OR INVALID
DMKCSV035E DEVICE TYPE MISSING OR INVALID
DMKCSV042E SPOOLID nnnn DOES NOT EXIST
DMKTCS240E PRT rdev, FCB-FORMS MISMATCH, FILEID nnnn {HELD|PURGED}
DMKTCS241E PRT rdev, INVALID LOAD MODULE mmmm SPECIFIED
           FILE nnnn {HELD|PURGED}
```

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 VM/SP installation: uniprocessor mode (UP), attached processor mode (AP), or multiprocessor mode (MP).
- 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 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.

For ease of use, the QUERY command and operands described in this section have been separated into the operands available for general users (class G) and those available to all users except class Any.

QUERY Command for Class G Users

The format of the Class G QUERY command is:

```
Query    { Time                                                              }
         { Set                                                               }
         { TERMinal                                                          }
                                                                             
Query    {                                         [          ] 1            }
         {                                         [ NOHold   ]              }
         { Files [CLass c] [FORM form]             [ HOld     ]              }
         {                                         [ USERhold ]              }
         {                                         [ SYShold  ]              }
         {                                                                   }
         { {             [ CHANnels       ] }                                }
         { {             [ GRaf           ] }                                }
         { {             [ CONsole        ] }                                }
         { {             [ DAsd           ] }                                }
         { {[Virtual]    [ TApes          ] }                                }
         { {             [ LINES          ] }                                }
         { {             [ UR             ] }                                }
         { {             [ STORage        ] }                                }
         { {             [ ALL            ] }                                }
         { {             [ vaddr[-vaddr]  ] }                                }
         {                                                                   }
         { Links vaddr                                                       }
         { CPUid                                                             }
         { SECuser                                                           }
         {                                                                   }
         { Reader   [                           [ NOHold   ] 1 ]             }
         { Printer  [ [CLass c]  [FORM form]    [ HOld     ]   ][ALL] [TBL]1 }
         { PUnch    [       spoolid             [ USERhold ]   ]             }
         {          [                           [ SYShold  ]   ]             }
         {                                                                   }
         { PF[nn]                                                            }
         { SCREen                                                            }
         { PROCess                                                           }
         { VMSAVE                                                            }
         { S370E                                                             }
         { SPMODE                                                            }
         { USERID                                                            }
         {                                                                   }
         {       [ CUrrent         ]                                         }
         { PER   [ Names           ]                                         }
         {       [ traceset-name   ]                                         }
         {       [ ALL             ]                                         }
```

[1] These options are mutually exclusive.

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.

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.

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 locally and remotely attached display devices and graphics devices dedicated 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 communication lines.

UR displays the status of all your unit record devices.

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.

`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(s)
- Vertical format (FCB)

- Copy modification module and trc number
- 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 the TBL option is used.

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 to VM/SP.

`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 support is active or inactive (not available) on the system.

SPMODE displays whether single processor mode is ON or OFF for VM/SP.

VMSAVE displays information about all of the target VMSAVE areas specified in module DMKSNT that this user is either the owner or receiver of.

USERID displays the userid and system identifier (if applicable).

PER 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.

QUERY Command Responses for Class G Users

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.

VIRTCPU= mmm:ss.hs is the virtual CPU time used in the current terminal session in minutes:seconds.hundredths of seconds.

TOTCPU= mmm:ss.hs is the total CPU time (virtual and overhead) used in the current terminal session in minutes:seconds.hundredths of seconds.

QUERY SET

```
                                      |ON  |
                                      |OFF |
    {ON  }       {ON  }               |CODE|
    {OFF }       {OFF }               |TEXT|           {ON }        {ON }
MSG {IUCV}, WNG  {IUCV}, EMSG         |IUCV|, ACNT     {OFF}, RUN   {OFF}

         {ON }          {ON  }          {ON }           {ON }
LINEDIT  {OFF}, TIMER   {OFF }, ISAM    {OFF}, ECMODE   {OFF}
                        {REAL}

       {ON          {SVC   } {TMR   }}
       {NOT ACTIVE  {NOSVC } {NOTMR }}           {ON }                {ON }
ASSIST {OFF                           }, PAGEX   {OFF}, AUTOPOLL      {OFF}

     {ON  }        {ON  }
     {OFF }        {OFF }                {NONE}               {ON }
IMSG {IUCV},SMSG   {IUCV}, AFFINITY      {nn  },NOTRAN        {OFF},

       {ON }       {ON }
VMSAVE {OFF},370E  {OFF}

          {VR    }
          {NNNNNK}             {N/Q xx/yy}
STBYPASS  {OFF   }, STMULTI    {OFF      }

    {ON }           {OFF }           {OFF }
MIH {OFF},VMCONIO   {IUCV},CPCONIO   {IUCV}
```

Q in the STMULTI response is the actual number of shadow tables being used. With the exception of Q, all of the response values are the result of and explained with the SET command.

The settings of all functions controlled by the SET command and the VM/SP directory ISAM and ECMODE options are displayed. Refer to the discussion of the SET command for explanation of these functions.

QUERY TERMINAL

```
                                                                            {ON  }
        {n  }           {n  }            {n  }          {n  }             {OFF }
LINEND  {OFF}, LINEDEL  {OFF}, CHARDEL   {OFF}, ESCAPE  {OFF}, TABCHAR    {CHAR}

          {nnn}       {ON }       {ON }        {ON }        {CP}             {ON }
LINESIZE  {OFF}, ATTN {OFF}, APL  {OFF}, TEXT  {OFF}, MODE  {VM}, HILIGHT    {OFF}

                                                  |PA1 |
         {3215}            {IMMED    }            |PF1 |               {ON }
CONMODE  {3270}, BREAKIN   {GUESTCTL }, BRKKEY    | .  |, SCRNSAVE     {OFF}
                                                  | .  |
                                                  |PF24|

      {3101}           {VM }           {nnn }
TYPE  {TTY }, PROMPT   {TTY}, SCROLL   {CONT}
```

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= aabbbbbbccccdddd

where:

`aabbbbbbccccdddd`
is the 16-digit processor identification associated with the virtual machine.

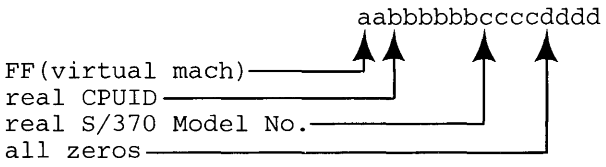
`aa` is the version code; these two digits are forced to X'FF' to identify that the virtual machine is running under VM/SP.

`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 VM/SP 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 303X or 4300 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.

`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     }
           { ON DEV raddr  }  VSMvvvvvvvv
           { NOT READY     }
```

The status of all locally and remotely attached display devices and graphics devices dedicated 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 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.
`vvvvvvvv`	is the userid of the VSM controlling the SNA device.

QUERY VIRTUAL CONSOLE

```
                 luname
                 GRAF                TERM     STOP
CONS vaddr   ON  LINE      raddr   NOTERM     START
             DISCONNECTED

                 CONT      HOLD                     READY
vaddr CL c     NOCONT    NOHOLD    COPY nnn      NOTREADY   FORM form

        TO
vaddr   FOR   userid   DIST distcode FLASHC ccc

vaddr FLASH ovly CHAR ctab MDFY 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 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.

`fcb` 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. Luname is a one-to-eight character identifier assigned to a SNA terminal user and use by CP instead of raddr. It is valid for SNA terminals connected to the host VM/SP system by the VTAM Communications Network Application (VM/VCNA) program product or by VM SNA Console Services (VSCS).

`vvvvvvvv` is the userid of the VTAM service machine (VSM).

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

```
                        { R/W } { nnnnnn BLK }
DASD vaddr type volser  { R/O } { nnn CY     }
```

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.

Notes:

1. *A display of 3340 represents either a 3340 or 3344 device.*
2. *For a dedicated DASD device, 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 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.

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 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 of the line.

```
           { ENABLED  }
LINE vaddr { DISABLED }
```

The status of virtual communication lines at virtual address vaddr is displayed.

QUERY VIRTUAL UR

```
                  {   CONT }{   HOLD }{   EOF }{    READY }
RDR vaddr CL c    { 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.

The default settings for a reader are:

RDR vaddr CL * NOCONT NOHOLD READY EOF

```
                 {   CONT }{   HOLD }             {    READY }
PUN vaddr CL c   { NOCONT }{ NOHOLD } CPY nnn     { NOTREADY } FORM form

      vaddr  { TO  }
             { FOR } userid   DIST distcode

                 {   CONT }{   HOLD }                 {    READY }
PRT vaddr CL c   { NOCONT }{ NOHOLD } CPY [*] nnn     { NOTREADY } FORM form

      vaddr  { TO  }
             { FOR } userid DIST distcode FLASHC ccc

      vaddr FLASH ovly  CHAR ctab [char[char[char]]] MDFY cmod n FCB fcb

                      { 2WCGM }{ BTS }{ DATCK   }  3800-1
      vaddr SIZE ww ll { 4WCGM }{ CFS }{ NODATCK }  3800-3
```

The status of all the virtual printers and punches attached to your virtual machine is displayed.

Note: The last line of the response 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.

Note: Further information can be obtained in the "Usage Notes" discussion of the SPOOL 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.

`fcb` 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 VM/SP directory.

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 0F 11 4WCGM CFS NODATCK 3800-1
```

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

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 XXXXXXXX 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 XXXXXXXX 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 XXXXXXXX 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 READER, QUERY PRINTER, QUERY PUNCH

```
                                      Additional Information

                              ┌── Query ───┐
                              CPY  HOLD  ┌FORM┐
                              nnn  stat  └form┘
┌─── Basic Information ────┐  ┌──────────── Query All ─────────────┐
ORIGINID  FILE  CLASS RECORDS CPY  HOLD DATE   TIME     NAME TYPE DIST
userid    file  c typ norecs  nnn  stat mm/dd  hh:mm:ss name type code
                                ┌──────────── Query Tbl ──────────────────────┐
                                FLASH FCB MOFY  FLSHC LOAD   CHARS
                                ovly  fcb mod n ccc  ⎧NO ⎫  cat [cat [cat [cat]]]
                                                     ⎨BEG⎬
                                                     ⎩ANY⎭
```

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 to identify the file.

a
: is the spool file class.

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.

name 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.

type 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.

fcb 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 could be 6, 8, or 12.

mod 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.

cat is the name of the character arrangement table(s) used to generate output.

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(s), 3800 vertical formatting, 3800 copy modifications, copy modification trc, 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

```
     ⎧       ⎫ ⎡tab...      ⎤
     ⎨ IMMED ⎬ ⎢copy...     ⎥
PFnn ⎨ DELAY ⎬ ⎣pfdata1...  ⎦
     ⎩       ⎭
  RETRIEVE
```

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 pfdatal 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     NO        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 has 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 or when VM/SP itself terminates. See the discussion on the VMSAVE option in the *VM/SP System Programmer's Guide* for more information regarding the specification and use of the VMSAVE option.

QUERY S370E

```
S370E {ON } [PROC xx, {ON } PROC yy]
      {OFF}           {OFF}
```

This command tells you whether or not the System/370 Extended (hardware) special feature has been installed and, if installed, whether or not the feature has been enabled by the VM/SP system operator. When installed and enabled, an MVS virtual machine may use the feature after the MVS operator of that virtual machine has issued the SET 370E command.

With attached processors, the VM/SP system operator may enable the feature for either or both processors. When enabled for only one processor, you must SET AFFINITY ON for that processor before VM/SP accepts the SET 370E command.

QUERY PROCESSOR

```
PROCESSOR mm ONLINE[, PROCESSOR nn ONLINE, SYSTEM IN {AP} MODE]
                                                     {MP}
```

where:

`mm` is the processor ID number of the main processor if VM/SP 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 VM/SP is running in AP mode; in MP mode, it is the processor ID number of the non-IPL processor.

If VM/SP is running in uniprocessor mode, the bracketed portion of the response is omitted. If VM/SP is running in AP or MP modes, the bracketed portion of the response is displayed.

QUERY SPMODE

```
SPMODE {ON }
       {OFF}
```

In a VM/SP system, this response indicates whether or not VM/SP is operating in single processor mode. If it is, VM/SP uses only the main processor in the configuration; the V=R virtual machine has exclusive use of the other processor.

Note: When VM/SP 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 has resumed AP or MP mode.

QUERY USERID

```
userid [AT sysid]
```

where:

`userid` is the name of the virtual machine.

`sysid` is a one to eight character name that identifies the VM system. The name is assigned by the system programmer; if no name has been assigned to the system being used, the word 'AT' and the system identifier are omitted.

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 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.

QUERY Command for All Classes of Users (Except Class Any)

This form of the QUERY command is for all classes of users except those in the Any category. Use this form of the QUERY command to:

- Display the log messages.
- List all the users that are logged on.
- Display the number of users that are logged on or dialed to VM/SP.

The format for this QUERY command is:

```
Query        { CPLEVEL               }
             { LOGmsg                }
             { Names                 }
             { Users    [userid]     }
             { userid                }
```

where:

CPLEVEL displays the system software product name, release number, service level number, 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.

QUERY Command Responses for All Classes of Users (Except Class Any)

QUERY CPLEVEL

```
VM/SP 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. This number indicates which service tape has been applied to the CP software; it cannot identify which individual updates have been incorporated intobCP.

`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 that 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 NAMES

```
userid - (DSC   ) , ...
   .     <raddr >
   .     (resid )
   .
userid - (DSC   ) , ...
         <raddr >
         (resid )

VSM    - {userid}, ...
userid - {luname}, ...
userid - Lnnn
```

A list of all logged-on users is displayed; if the user is currently connected, the real address (raddr), or the resource identification of a 3704/3705/3725 line (resid) to which he is connected is displayed; if he is not connected to the system, DSC is displayed. The userid - luname response displays the userids and logical unit names (lunames) of all users accessing the VM/SP system in a SNA network through the VTAM Service Machine (VSM). The VSM - userid displays the userids of all users connected through that particular VSM to the host VM/SP system. If your installation has the VM/Pass-through program product, Lnnn displays the logical device name of the user accessing the host VM/SP system through the Pass-through virtual machine.

QUERY USERS

```
nnn USERS, mmm DIALED, xxx NET
```

The number of users logged on and dialed to VM/SP is displayed.

where:

nnn	is the total number of logged-on users.
mmm	is the total number of users attached via 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  }
         { resid  }
         { luname }
         { Lnnn   }
```

The real address (raddr) or the resource identification of a 3704/3705/3725 line (resid) to which the specified user is connected is displayed. The luname is displayed instead of the raddr or resid if that user is logged on through a SNA network using the VTAM Service Machine. If your installation has the VM/Passthrough program product, Lnnn displays the logical address of that user.

Messages

```
DMKCFJ026E OPERAND MISSING OR INVALID
DMKCQG020E USERID MISSING OR INVALID
DMKCQG022E VADDR MISSING OR INVALID
DMKCQG027E SPOOLID MISSING OR INVALID
DMKCQG040E DEV vdev DOES NOT EXIST

DMKCQG042E SPOOLID nnnn DOES NOT EXIST
DMKCQG045E userid NOT LOGGED ON
DMKCQP003E INVALID OPTION - option
DMKCQP023E VOLID MISSING OR INVALID
DMKCQP040E DEV rdev DOES NOT EXIST

DMKCQP466I CTLR rdev IN BUFFER SLOWDOWN MODE
DMKCQQ006E INVALID DEVICE TYPE - devtype
DMKCQQ021E RADDR MISSING OR INVALID
DMKCQQ022E VADDR MISSING OR INVALID
DMKCQQ023E VOLID MISSING OR INVALID

DMKCQQ040E DEV rdev DOES NOT EXIST
DMKCQQ045E userid NOT LOGGED ON
DMKCQQ466I CTLR rdev IN BUFFER SLOWDOWN MODE
DMKCQR003E INVALID OPTION - option
DMKCQR020E USERID MISSING OR INVALID

DMKCQR026E OPERAND MISSING OR INVALID
DMKCQR028E CLASS MISSING OR INVALID
DMKCQR045E userid NOT LOGGED ON
DMKCQS003E INVALID OPTION - option
DMKCQT002E INVALID OPERAND - operand

DMKCQT006E INVALID DEVICE TYPE - devtype
DMKCQT021E rdev MISSING OR INVALID
DMKCQT747A DEVICE rdev VARY IN PROGRESS, PLEASE RE-ENTER THIS
           COMMAND
DMKCQT747A CSW=(dsw)[, SENSE=(sense)], PLEASE RE-ENTER THIS
           COMMAND
DMKCQT747A ERROR ENCOUNTERED, NO IOERBLOK CAN BE FOUND,
           PLEASE RE-ENTER THIS COMMAND

DMKCQT747A DEVICE UNIT CHECK - NO SENSE DATA IS FOUND,
           PLEASE RE-ENTER THIS COMMAND
DMKCQY020E USERID MISSING OR INVALID
DMKCQY026E OPERAND MISSING OR INVALID
DMKCQY045E userid NOT LOGGED ON
DMKPEI047E {userid {vdev|segname} | THE CURRENT TRACESET | THE
           TRACESET traceset | SAVED TRACESET} DOES NOT EXIST

DMKPEQ003E INVALID OPTION - option
DMKPEQ180E PER TRACE NOT IN EFFECT
```

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.

Messages

```
DMKCPB006E INVALID DEVICE TYPE - devtype
DMKCPB022E VADDR MISSING OR INVALID
DMKCPB040E DEV vdev DOES NOT EXIST
```

REQUEST

Privilege Class: G

Use the REQUEST command to make an attention interrupt 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 interrupts are pending. This condition may result if the user issues the REQUEST command by means of DIAGNOSE X'8'.

Responses

None.

Messages.

None.

RESET

Privilege Class: G

Use the RESET command to clear all pending interrupts 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 interrupts, 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.

3. 8809 dedicated tapes are reset to the default mode settings of slow speed and normal gap. However, if a tape is not mounted and ready when the reset is issued, it will not be reset to the defaults.

Responses

```
DEVICE RESET
```

is the response you receive upon the successful completion of the command.

Messages

```
DMKCPB022E VADDR MISSING OR INVALID
DMKCPB040E DEV vdev DOES NOT EXIST
```

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 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.

Messages

```
DMKCPB006E INVALID DEVICE TYPE - devtype
DMKCPB022E VADDR MISSING OR INVALID
DMKCPB040E DEV vdev DOES NOT EXIST
```

SCREEN

Privilege Class: G

Use the SCREEN command to alter or change any extended 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 is only valid when the Extended Color Feature has been applied to the terminal controller.

You can assign extended 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 refer to "Section 2. Display Terminals" in *VM/SP Terminal Reference.*

The format of the SCREEN command is:

```
SCREen   { area { [ { extcolor }   { exthilight } ] } 1 }
                    { DEFault  }   {    NONe    }
```

[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 VM/SP 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 color attribute you want to assign to an 'area' of your screen. Color can be:

BLUe — blue
RED — red
GREen — green
YELlow — yellow
TURquois — turquoise
PINk — pink
WHIte — white
DEFault — monochrome is the default.

`exthilight`

specifies the extended highlight value you want to assign to an area of your screen. The exthilight 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. These 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.

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 completed.

Messages

```
DMKCFW002E INVALID OPERAND - operand
DMKCFW026E OPERAND MISSING OR INVALID
```

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 System Programmer's Guide.

The format of the SEND command is:

`SEND`	`[CP] userid [text]`

where:

`CP` indicates that CP executes the text in 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 userid of the secondary user defined 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 also applies when '#CP command' is the command text because the logical linedit character is in fact a carriage return, implying multiple lines.

Usage Notes

1. The issuer of the SEND command must be authorized by the secondary user by means of the CONSOLE statement in the directory entry of the receiving virtual machine.

2. The receiver of the SEND command input must be running disconnected.

3. When a logical linedit character is sent in the SEND command text, the character must be that of the sender, not the disconnected machine.

Responses

None.

Messages

```
DMKSND020E USERID MISSING OR INVALID
DMKSND045E userid NOT LOGGED ON
DMKSND068E SEND COMMAND FAILED - RECEIVER {IS NOT
           DISCONNECTED|HAS NO AUTHORIZED SENDER|HAS NO VIRTUAL
           CONSOLE|HAS CONSOLE INPUT WAIT| IS EXECUTING A
           CONSOLE FUNCTION}
```

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
         EMSG         ON
                      OFF
                      CODE
                      TEXT
                      IUCV
         IMSG         ON
         MSG          OFF
         SMsg         IUCV
         WNG
        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#...pfdatan]
     [DELayed]

PFnn RETrieve
PFnn (TAB n1 n2 ...  )

PFnn (COPY (resid))

PFnn (COPY (cuu))

PFnn (COPY (luname))

VMSAVE {ON  }
       {OFF }
       {name}

STBypass { [nnnnnK    [NOVERIFY]] }
         { [nnM       [        ]] }
         {                        }
         {  VR                    }
         {  OFF                   }

STMulti  { [n  [[USEG xx]  CSEG yyy]] }
         { [ON [[       ]          ]] }
         {                            }
         {  OFF                       }

370E  {ON }
      {OFF}

MIH   {ON }
      {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 log on VM/SP, 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, via 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 log on VM/SP, 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 ON is specified, the virtual machine will receive special messages via VMCF. 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 via the CP WARNING command from the system operator are received at the terminal. If OFF is specified, no warning messages are received. When you log on to VM/SP, 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 data 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 log on to VM/SP, 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 data 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 key (causing a read of a CP command) without stopping your virtual machine. When the CP command is entered, it is immediately executed and the virtual machine resumes execution. OFF places the virtual machine in the normal CP environment, so that when the attention key is pressed, the virtual machine stops. When you log on VM/SP, RUN is set off.

```
LINEDIT { ON  }
        { OFF }
```

controls the line editing functions. ON specifies that the line editing functions and the symbols of the VM/SP 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 log on VM/SP, 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 System/370 extended control mode and control registers 1 through 15. Control register 0 may be used with ECMODE either ON or OFF. When you log on VM/SP, ECMODE is set according to the user's directory option; ON if ECMODE was specified and OFF if not.

Note: Execution of the SET ECMODE {ON | OFF} command always causes a virtual system reset. 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 log on VM/SP, ISAM is set according to the user's directory options; 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 via the IPL command. (IPL sets the NOTRANS option to an OFF condition.)

```
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 log on to VM/SP, 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.

```
EMSG { ON   }
     { OFF  }
     { CODE }
     { TEXT }
     { IUCV }
```

controls error message handling. ON specifies that both the error code and text are displayed at the terminal. OFF specifies that no error message is to be displayed. CODE specifies that only the error code is be displayed. TEXT specifies that only text is displayed. 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. When you log on to VM/SP, EMSG is set to TEXT.

If the console is being spooled for error messages from CP, the OFF setting is ignored for the spooled output and the full error message appears in the spooled output. The other four settings result in spooled output that matches the console printout.

It is the responsibility of the operating system running under CP to determine whether or not its own error messages will honor the EMSG setting in the spooled output.

Note: CMS always recognizes EMSG settings for all error (E), information (I), and warning (W) messages, but ignores the EMSG setting and displays the complete message (error code and text) for all response (R), severe error (S), and terminal (T) messages. Also, data written by means of 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 *MSG 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 CPU run time and also during virtual wait time. If the REALTIMER option is specified in your VM/SP directory entry, TIMER is set to REAL when you log on; otherwise it is set to ON when you log on.

```
AFFINITY ┌ OFF ┐
         └ ON  ┘
```

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. AFFINITY ON causes the directory to be interrogated for an affinity setting; if there isn't an affinity setting 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.

`CPUID bbbbbb`

CPUID is the keyword that provides a means of setting or changing the CPU identification for a virtual machine. The variable value bbbbbb represents a string of six hexadecimal digits that permit unique identification of the CPU. This command allows the user to provide a unique CPU 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 by VM/SP to ensure that all virtual machine using the SET CPUID command have specified unique CPU serials. The hexadecimal field 'bbbbbb' is the CPU identification number (CPU serial).

Note: The CPU 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:

aabbbbbbccccdddd

where:

`aa` version code; these two digits are forced to X'FF' to identify that the virtual machine is running under VM/SP.

`bbbbbb` CPU identification number; this field contains six hexadecimal digits. This is the only part of the CPUID that can be set by the DIRECTORY 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  ] }
        {       [NOSVC] [NOTMR] }
        {                       }
        {  OFF                  }
```

All functions of SET ASSIST operands pertain to VM/SP hardware assist. VM/SP hardware assist consists of processor hardware that assists the processor in the execution of certain instructions, series of instructions, privileged operations, and interrupt processing that are normally handled by the control program portion of VM/SP. In attached processor or multiprocessor systems, VM/SP hardware assist is available for installation on either or both processors.

All, some, or none of the function performed by VM/SP 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 hardware assist available on supported VM/SP processors, refer to the *VM/SP Operator's Guide.*) The extent of VM/SP 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 VM/SP 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 Operator's Guide* for details on these commands.

If VM/SP 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 log on unless your VM/SP 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 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 handles all the SVC interruptions.

If the user requests VM/SP hardware assist and it is available and active, the request is honored; if VM/SP hardware assist is not active but is available and the user attempts to enable hardware assist, the virtual machine remembers the enabling attempt and an appropriate message is sent to the user. When the system operator enables hardware assist, it is immediately utilized by the virtual machine. In the event that hardware assist is not available or active on the system, an appropriate error message is issued. 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 machines affinity must be set to match the processor that contains hardware assist.

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 System Programmer's Guide* for additional information on how to use VM/SP hardware assist.

```
PFnn [IMMED  ]   [pfdata1#pfdata2#...pfdatan]
     [DELAYED]
```

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 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. Care should be exercised when defining PFkey data for non-immediate mode of operation. The resulting write to the screen may result in a reset of VM/370's attribute bytes.

`pfdata1#pfdata2#...pfdatan`

defines the VM/SP 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 ("). For further explanation, see the "Usage Notes" section that follows. 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.

`PFnn [COPY [resid]]`

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 actuated. 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. The resource identification, resid, may be specified if more than one printer is connected to the same control unit as your display terminal. It is a two-character hexadecimal resource identification number assigned to a specific printer. If resid is entered, the printed copy is directed to a specific printer; if not, the copy is printed on the printer with the lowest resid number. The resid numbers of the printers available to your display terminal can be obtained from your system operator. If only one printer is available, as with the 3275 Display Station, resid need not be specified.

`PFnn [COPY [cuu]]`

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 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 service machine. The number, nn, 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 PFnn COPY function can be used for any printer managed by VM/VTAM, i.e. not restricted to those connected to the same controller, and works on any terminals controlled by VM/VTAM service machine.) 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.
   - The SET PFnn COPY command format is invalid.

   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.

2. To ensure 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.

3. 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 redisplays the most current input line from the input area, so that you can modify and re-enter the data.

VM/SP can actually keep track of several input lines. The number of lines kept depends on the length of the lines; VM/SP 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 redisplays the most current input line. If the key is pressed again, VM/SP displays the previous input line. Each time the key is pressed, VM/SP steps through the input lines, displaying them one at a time. When VM/SP reaches the oldest line that it has kept, it returns to the most current line once again.

VM/SP resets the RETRIEVE function, each time a new input line is entered, to retrieve that (the most current line). A simple way to quickly reset the RETRIEVE function to the most current line is: enter a single asterisk. Then press the RETRIEVE program function key once to get the asterisk redisplayed, and a second time to get the previous input line redisplayed.

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 nor 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. OFF disables the option for the user. ON is the default when the VMSAVE option is specified in that user's directory entry. 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 System Programmer's Guide* and the *VM/SP Planning Guide and Reference* for more information.

```
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 VM/SP V=R area.

OFF terminates the shadow table bypass function.

```
STMulti  { [n  [[USEG] xx   CSEG yyy]] }
         { [ON [[    ]              ]] }
         {                             }
         {  OFF                        }
```

allows users running in the system V=V area to have VM/SP maintain multiple shadow tables when a virtual operating system, such as MVS, SVS and VM/SP itself, uses multiple segment tables.

n defines the number of shadow tables between 1 and the maximum of 6.

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 0 through 99. 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.

The STMULTI option is accepted only if ECMODE is set ON. This occurs when a second level system (e.g. MVS, SVS, etc.) is running in a virtual machine.

```
370E {ON }
     {OFF}
```

provided the System/370 Extended (hardware) special feature has been installed and the VM/SP system operator has enabled it, this operand allows you to execute the OS/VS2 MVS/System Extensions Program Product (program number 5740-XE1) 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 VM/SP 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 accepts the SET 370E command.

```
MIH {ON }
    {OFF}
```

provided that DMKDID is installed on the system, MIH may be set on or off. When MIH is on, the interrupt is simulated by CP. When MIH is off, the missing interrupt is detected and a message is written to the operator, but the interrupt is not simulated by CP.

Note: When MIH is on, an interface control check (IFCC) will be reflected to the user if an interrupt cannot be simulated.

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. If an interrupt-driven system, such as Customer Information Control System (CICS), is to be run in a virtual machine, SET TIMER REAL must be specified on systems which use the interval timer.

3. 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.

4. In the case of systems with attached processors or multiprocessors, the VM/SP 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, with the AFFINITY setting of a virtual machine, the user can receive no response indicating that VM/SP hardware assist is installed and active on his processor. In attached processor or multiprocessor setups where the user has no processor preference (AFFINITY=NONE) upon entering SET ASSIST he too receives no response if either processor has VM/SP hardware assist active. Any error message is indicative that VM/SP hardware assist is not available or active for the user. If the user receives a message indicating that VM/SP 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 hardware, need only change his affinity setting, then initiate the SET ASSIST command.

5. 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.

6. 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:

7. If you use the set run on, it is possible that command output may be interrupted by a guest virtual machine read or write.

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.

8. If a terminal has an inhibited (non-display) read up and either a delayed PFkey or an undefined PFkey is used, the input will be rewritten without the inhibited attribute byte, therefore displaying any data typed at that point. The clear key can be used following the PFkey to rewrite the input area with the inhibit attribute.

Responses

None.

Messages

```
DMKCFJ003E INVALID OPTION - option
DMKCFO020E USERID MISSING OR INVALID
DMKCFO026E OPERAND MISSING OR INVALID
DMKCFO045E userid NOT LOGGED ON
DMKCFO175E {FAVORED|RESERVE} ALREADY IN USE BY userid

DMKCFO184E VM ASSIST NOT AVAILABLE {ON PROCESSOR xx}
DMKCFO186E CP ASSIST NOT AVAILABLE
DMKCFO188E SPECIFIED PROCESSOR UNAVAILABLE
DMKCFO250E 370E IS NOT AVAILABLE {ON PROC xx}
DMKCFO255W 370E FEATURE NOW DISABLED

DMKCFS003E INVALID OPTION - option
DMKCFS006E INVALID DEVICE TYPE - devtype
DMKCFS026E OPERAND MISSING OR INVALID
DMKCFS045E userid NOT LOGGED ON
DMKCFS046E DEV devtype rdev OFFLINE

DMKCFS183E VM ASSIST NOT ACTIVE
DMKCFS184E VM ASSIST NOT AVAILABLE
DMKCFS187E TIMER ASSIST NOT AVAILABLE
DMKCFS250E 370E IS NOT AVAILABLE {ON PROC xx}
DMKCFS251E 370E IS NOT ACTIVE

DMKCFU003E INVALID OPTION - option
DMKCFU006E INVALID DEVICE TYPE - devtype
DMKCFU013E CONFLICTING OPTION - option
```

```
DMKCFU021E RADDR MISSING OR INVALID
DMKCFU026E OPERAND MISSING OR INVALID

DMKCFU040E DEV rdev DOES NOT EXIST
DMKCFU046E devtype rdev OFFLINE
DMKCFU140E devtype rdev ATTACHED TO userid
DMKCFU143E devtype rdev IN USE BY SYSTEM
DMKCFU199E COMMAND REJECTED; DEVICE rdev PENDING OFFLINE

DMKCFU547E MISSING INTERRUPTION MONITORING NOT AVAILABLE
DMKCFU925I SYSTEM AUTO DUMP REALLOCATED TO DASD rdev
DMKCFU953I UNABLE TO ALLOCATE SYSTEM AUTO DUMP
DMKCFV062E ECMODE NOT SET ON
DMKCFV063E SHADOW TABLE BYPASS: NOT SET

DMKCFV080E NO PREALLOCATED VMSAVE AREA
DMKCFV082E MORE THAN ONE VMSAVE AREA ALLOCATED; MUST SPECIFY AREA NA
DMKCFY052E ERROR IN CP DIRECTORY
DMKCFY053E userid NOT IN CP DIRECTORY
DMKCFY183E VM ASSIST NOT ACTIVE

DMSCFY184E VM ASSIST NOT AVAILABLE
DMKCFY187E TIMER ASSIST NOT AVAILABLE
DMKCFY188E SPECIFIED PROCESSOR UNAVAILABLE
DMKCFY189E AFFINITY NOT SET; SYSTEM IN UNIPROCESSOR MODE
DMKCFY190I NO AFFINITY SET IN DIRECTORY

DMKMCI003E INVALID OPTION - option
DMKMCI026E OPERAND MISSING OR INVALID
DMKMSG003E INVALID OPTION - option
DMKVAT064E SHADOW TABLE BYPASS; RESET
```

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    ┌    ┌     ┐ ┐
         │    │ SEC │ │
         │ nn │ MIN │ │
         │    │ HRs │ │
         └    └     ┘ ┘
```

where:

```
┌    ┌     ┐ ┐
│    │ SEC │ │
│ nn │ MIN │ │
│    │ HRs │ │
└    └     ┘ ┘
```

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 specify no time unit, the value of nn is taken to be in minutes.

Usage Notes

1. During the dormant period, the virtual machine does not run but connection time is still being counted.

2. 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 be reset if the LOGON command is chained to it.

8. If the SLEEP command is issued 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.

Messages

```
DMKCFJ002E INVALID OPERAND - operand
```

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:

$$\underbrace{\texttt{smsg pd58gh1c}}_{\texttt{14 characters used}} \qquad \underbrace{\texttt{special message}}_{\texttt{125 characters left}}$$

$$\underbrace{\texttt{sm u}}_{\texttt{5 characters used}} \qquad \underbrace{\texttt{special message}}_{\texttt{134 characters left}}$$

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, 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.

   Note: For more information on VMCF and IUCV, refer to the *VM/SP System Programmer's Guide.* For more information on the ALLOW option of IUCV, refer to the *VM/SP Planning Guide and Reference.*

Responses

None.

Messages

```
DMKMSG002E USERID MISSING OR INVALID
DMKMSG045E userid NOT LOGGED ON
DMKMSG057W userid NOT RECEIVING; {DISCONNECTED|SMSG OFF|NOT
           AUTHORIZED|WNG OFF}
```

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 below in the section "Transmitting Files to Remote Locations." The default values for the SPOOL command appear under the "Usage Notes" following the SPOOL operand descriptions.
The format of the SPOOL command is:

```
SPool   {Reader} {[CLass {c}] [CONt  ] [HOld  ] [EOF  ]  }1
        {vaddr } {[      {*}] [NOCont] [NOHold] [NOEof]  }

        {Printer} ( [To ] {userid} ] [HOld  ][CONt  ]                      1 )
        {PUnch  } ( [For] {  *   } ] [NOHold][NOCont] [CLass c] [COpy[*]nnn] )
        {CONsole} ( [     {SYSTEM} ]                                         )
        {vaddr  } ( [OFF           ]                                         )
                  (                                                          )
                  (            3                                             )
                  ( [CLOSE] [FLash name nnn] [FORM {form}] [Dist {distcode}] )
                  ( [PURGE] [MOdify name[n]] [     {OFF }] [     {OFF     }] )
                  (         [CHars name1   ]                                 )
                  (         [     [name2   ]                                 )
                  (         [     [name3   ]                                 )
                  (         [     [name4]]]]                                 )
                  (         [CHars name1   ]                                 )
                  (         [ [CHars name2 ]                                 )
                  (         [ [CHars name3 ]                                 )
                  (         [ [CHars       ]                                 )
                  (         [     name4]]] ]                                 )
                  (         [FCB name      ]                                 )
                  (                                                          )
                  (                          [STArt] 2    [TErm   ]2         )
                  (                          [STop ]      [NOTErm ]          )
```

[1] At 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.

[2] These options apply only to a virtual spooled console.

[3] These options can only be used to modify a virtual spooling printer. These options 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, the input files for the specified reader are not deleted after they are used, but 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 via 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.

$\texttt{FOR} \left\{\begin{matrix}\texttt{userid}\\ *\\ \texttt{SYSTEM}\end{matrix}\right\}$

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 VM/SP 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 Notes") 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.

Usage Notes

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 VM/SP 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

If the user specifies a FLASH name, it must match the FLASH name specified by the operator on the START command for the file to print on a 3800.

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 a spool file containing 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.

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 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. *VM/SP 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 via 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 recode customized 3800 Model 1 character sets and build new modules through the use of the GENIMAGE command. 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 VM/SP 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 via 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 readers; 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 installations' 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 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, 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 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 log on 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 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.

Messages

```
DMKCQH013E CONFLICTING OPTION - option
DMKCQH024E {CYLS|BLKS|FORM NUMBER} MISSING OR INVALID
DMKCSO024E {CYLS|BLKS|FORM NUMBER} MISSING OR INVALID
DMKCSP003E INVALID OPTION - option
DMKCSP006E INVALID DEVICE TYPE - devtype
DMKCSP007E INVALID USERID - userid

DMKCSP013E CONFLICTING OPERAND - operand
DMKCSP020E USERID MISSING OR INVALID
DMKCSP022E VADDR MISSING OR INVALID
DMKCSP024E {CYLS|BLKS|FORM NUMBER} MISSING OR INVALID
DMKCSP026E OPERAND MISSING OR INVALID

DMKCSP028E CLASS MISSING OR INVALID
DMKCSP030E COPIES MISSING OR INVALID
DMKCSP040E DEV rdev DOES NOT EXIST
DMKCSP053E userid NOT IN CP DIRECTORY
DMKCSP243E FLASH MISSING OR INVALID

DMKCSP245E CHARS MISSING OR INVALID
DMKCSP246E FCB MISSING OR INVALID
DMKCSP253E MODIFY MISSING OR INVALID
DMKCSQ024E {CYLS|BLKS|FORM NUMBER} MISSING OR INVALID
DMKCSU024E {CYLS|BLKS|FORM NUMBER} MISSING OR INVALID

DMKCSV013E CONFLICTING OPTION - option
DMKCSV024E {CYLS|BLKS|FORM NUMBER} MISSING OR INVALID
DMKSPT024E {CYLS|BLKS|FORM NUMBER} MISSING OR INVALID
```

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 Notes")
- 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  ⎧ hexloc   ⎫
       ⎪ Lhexloc     hexword1  [hexword2...]  ⎪
       ⎪                                       ⎪
       ⎪ Shexloc     hexdata...                ⎪
       ⎪                                       ⎪
       ⎪ ⎧Greg⎫                                ⎪
       ⎨ ⎩Xreg⎭      hexword1  [hexword2...]   ⎬
       ⎪                                       ⎪
       ⎪ {Yreg}      hexdword1  [hexdword2...] ⎪
       ⎪                                       ⎪
       ⎪ Psw        [hexword1]  hexword2       ⎪
       ⎪                                       ⎪
       ⎩ STATUS                                ⎭
```

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 hexword1 operand represents 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...]operand each represents 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...] operand each represents 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 more more hexdwords 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 only hexword2 is specified, it is stored into the second word of the PSW. The [hexword1] hexword2 operand each represents 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 CPU 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 via the DISPLAY command. The procedure is the only way to get timer information at the terminal using a CP command.

Response

STORE COMPLETE

is the response at the successful completion of the command.

Messages

```
DMKCDS004E INVALID HEXLOC - hexloc
DMKCDS005E INVALID HEXDATA - hexdata
DMKCDS010E INVALID REGISTER - register
DMKCDS012E INVALID PSW - psw
DMKCDS026E OPERAND MISSING OR INVALID

DMKCDS033E HEXLOC MISSING OR INVALID
DMKCDS160E HEXLOC hexloc EXCEEDS STORAGE
DMKCDS162E INVALID ECR x - xxxxxxxx
DMKCDS163E STORE EXCEEDS MAXIMUM REGISTER
DMKCDS164E HEXLOC hexloc NON-ADDRESSABLE STORAGE
DMKCDS166E CHANGED SHARED PAGE NOT WRITTEN TO AUXILIARY STORAGE
```

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 interrupts 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. Interrupt 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 interrupts, 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.

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'.

Messages

```
DMKCPB012E INVALID PSW - psw
DMKCPB026E OPERAND MISSING OR INVALID
```

TAG

Privilege Class: G

Use the TAG command to associate file descriptive information with a VM/SP spool file.

The format of the TAG command is:

```
TAg        ( Printer )
           ) PUnch   (
      DEv  ( CONsole ( [tagtext]
           ( vaddr   )

      FIle  spoolid  [tagtext]

                  (      ( Printer ) )
                  (      ) PUnch   ( )
                  ) DEv  ) CONsole ( (
      QUery       )      ( vaddr   ) (
                  (                  )
                  ( FIle  spoolid    )
```

where:

```
DEV ( PRINTER  )
    ( PRT      )
    ) PUNCH    (
    ) PCH      (
    ( CONSOLE  )
    ( vaddr    )
```

specifies a 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 and assigned by CP when the spool file was closed.

`tagtext` is the information (up to 136 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 RSCS can be specified in this field. For details on how to use the tagtext operand when transmitting files across the RSCS network, refer to the following "Usage Notes" section.

```
        /      / PRINTER \  \
       |       | PRT      | |
       | DEV   < PUNCH    > |
       |       | PCH      | |
      <        | CONSOLE  |  >
QUERY  |       \ 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 alphameric 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 VM/SP 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 VM/SP 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 via the TRANSFER command.

4. Although the buffer that contains TAG information is 136 characters in length including imbedded blanks, only 126 characters of it are usable because of input command line limitations.

5. The following spool file types cannot be tagged:

   - Monitor files
   - Accounting files
   - Dump files
   - Real card reader (input files)

6. The TAG command works only on files that are not currently in use.

Responses by Command Operands

```
              {PRINTER}
TAG QUERY DEV {PUNCH  }
              {CONSOLE}
```

Displays the contents of the tags associated with each device of the specified class, as follows:

```
{PRT }                 repeated for
{PUN } vaddr TAG:      each device in
{CONS}                 the specified
tagtext.........       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.

Messages

```
DMKCST003E INVALID OPTION - option
DMKCST006E INVALID DEVICE TYPE - devtype
DMKCST022E VADDR MISSING OR INVALID
DMKCST026E OPERAND MISSING OR INVALID
DMKCST027E SPOOLID MISSING OR INVALID

DMKCST040E DEV vdev DOES NOT EXIST
DMKCST042E SPOOLID nnnn DOES NOT EXIST
DMKCST424E RDR FILE SPOOL ERROR; FILE UNCHANGED
```

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.

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   }}
          {            {GUESTctl}}
          {                      }
          { BRKkey     {PA1 }    }
          {            {PF1 }    }
          {            { .  }    }
          {            { .  }    }
          {            {PF24}    }
          {                      }
          { TYpe       {3101}    }
          {            {TTY }    }
          {                      }
          { PROMpt     {VM }     }
          {            {TTY}     }
          {                      }
          { SCROll     {nnn }    }
          {            {CONT}    }
```

[1]More 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 VM/SP 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 log on, CHARDEL ON is in effect; if your virtual machine is logged on by the system operator via 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 ¢, but depends on what is specified in your VM/SP 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 log on, LINEDEL ON is in effect; if your virtual machine is logged on by the system operator via 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 VM/SP 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 log on, LINEND ON is in effect; if your virtual machine is logged on by the system operator via 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 VM/SP 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 log on, ESCAPE ON is in effect; if your virtual machine is logged on by the system operator via 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.

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.

2. The setting of the SET LINEDIT command is independent of the TERMINAL TEXT setting.

```
ATTN     {ON }
         {OFF}
```

controls signaling of an attention interrupt. If ATTN ON is specified, the exclamation point is displayed when an attention interrupt occurs. The OFF option suppresses the displaying of the exclamation point (!) and carriage 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 log on VM/SP, 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 interrupt. OFF specifies that CP does not save the guest screen (that is, a CLEAR interrupt is posted by CP to the guest when CP gives control back to the guest). OFF is the default.

Note: SCRNSAVE ON must be specified if running a guest SCP such as MVS with CONMODE 3270. Otherwise, if guest screen is not saved, unpredictable results can occur due to message traffic from CP. Thus, no problems should occur (with SCRNSAVE OFF) 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 users except the primary system operator. For more information on this, see "Controlling the Virtual Machine" in Section 3 of the VM/SP CP Command Reference for General Users.

```
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.

```
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 as the break in controller that does not exist on the guest virtual machine console hardware can lock out CP mode. Pressing PA1 will still cause the CP environment to be entered when operating in line mode. Some applications (for example, the System Product Editor) also interpret PA1 as a request for CP mode. Once the break key has been hit, the "begin" command must be issued to return to the guest virtual machine console.

```
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 program function keys one through 24.
- Causes output to begin at column 1 of line 24 on the display.

```
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 on 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 log on, the system defaults are in effect. However, the settings you specify for line-editing (except LINEDEL) and MODE are still in effect when you log on after disconnecting. All the other operands (TEXT, APL, ATTN, HILIGHT, and LINESIZE) are reset if you log on after disconnecting.

2. 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 line-editing functions are on or off.

3. If an error occurs during processing of the command, all functions preceding the one with the error are in effect.

4. 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, ESCAPE, and TABCHAR operands.

5. Figure 9 on page 245 describes the other features available for highlighting because the TERMINAL command cannot activate or deactivate these features.

6. The SCROLL, TYPE, and PROMPT operands must be issued from a TTY device or an error message results.

Highlighted Feature	Designated By:
Certain data fields for application programs running under CMS	Application programmer
CMS editor session: Current line (adjacent to SYSTEM AVAILABLE indicator) Error messages (second line on screen) '?EDIT' messages (also second line on screen)	System
Logon logo	System programmer
Non-edit CMS and CP display sessions: Messages or warnings from the system operator or other users	System

Figure 9. 3270 High Intensity Features

7. CP automatically masks your password on typewriter terminals. For installations migrating from a VM/370 operating system to VM/SP, 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.

8. If your virtual machine is not accepting interrupts and SCRNSAVE is set OFF, you must hit the BRKKEY twice in order to enter CP mode.

9. CONMODE 3270 and SCRNSAVE are session valid options only. They are not supported across disconnect of the virtual machine. These options must be reset, if desired, upon reconnection.

Responses

None.

Messages

```
DMKCFT002E INVALID OPERAND - operand
DMKCFT006E INVALID DEVICE TYPE - devtype
DMKCFT026E OPERAND MISSING OR INVALID
DMKGRF457E TERMINAL SCRNSAVE UNSUCCESSFUL
```

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. INTERRUPTION 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.

The format of the TRACE command is:

```
TRace   { (SVC     )1 [PRINter                      ] }
        { (I/O     )  [                             ] }
        { (PROgram )  [ [TERMinal]  [NORun]         ] }
        { (EXTernal)  [ [BOTH    ]  [RUN  ]         ] }
        { (PRIV    )  [                             ] }
        { (SIO     )  [                             ] }
        { (CCW     )  [  OFf                        ] }
        { (BRanch  )                                  }
        { (INSTruct)                                  }
        { (ALL     )                                  }
        { (CSW     )                                  }
        {                                             }
        {  END                                        }
```

[1]More than one of these activities may be traced by using a single TRACE command. For example:

```
TRACE SVC PROGRAM SIO PRINTER
```

where:

SVC traces virtual machine SVC interrupts.

I/O traces virtual machine I/O interrupts.

PROGRAM traces virtual machine program interrupts.

EXTERNAL traces virtual machine external interrupts.

PRIV traces all virtual machine non-I/O privileged instructions.

SIO traces TIO, CLRIO, HIO, HDV, and TCH instructions for all virtual devices both spooled and non-spooled. SIO and SIOF instructions will not be traced for spooled devices or consoles.

It also traces SIO instructions to attached remote printers and dedicated remote graphics terminals.

CCW traces virtual and real CCWs for nonspool nonconsole device I/O operations. Virtual and real CCWs are not traced for remote and VM/VTAM dialed devices. When CCW tracing is requested, SIO and TIO instructions to all devices are also traced.

BRANCH traces virtual machine interrupts, PSW instructions, and successful branches.

INSTRUCT traces all instructions, virtual machine interrupts, and successful branches.

ALL traces all instructions, interrupts, successful branches, privilege instructions, and virtual machine I/O operations.

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 that virtual printer that has 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'08' 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 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 interrupts. 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 doing 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. 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.

14. There are some modules in CMS that do not issue SIOs. The I/O operation is initiated via DIAGNOSE instructions.

Responses

The following symbols are used in the responses received from TRACE:

Symbol	**Meaning**
vvvvvv	virtual storage address
tttttt	virtual transfer address or new PSW address
rrrrrr	real storage address
xxxxxxxx	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 vadd	virtual device address
R radd	real device address
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 interrupt
: : :	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 vadd CC n
```

TIO, CLRIO, HIO, or HDV

```
I/O vvvvvv mnem xxxxxxxx type vadd CC n type radd CSW xxxx
```

SIO or SIOF

```
I/O vvvvvv mnem xxxxxxxx type vadd CC n type radd 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 not data relocation. Only the 2 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     ttttttt  ttttttt  (WAIT bit on)
::: vvvvvv LPSW  xxxxxxxx ==> ttttttt  ttttttt  (WAIT bit not on)
::: vvvvvv mnem  xxxxxxxx                       (all others)
```

Executed Instructions:

```
vvvvvv EX xxxxxxxx zz vvvvvv mnem xxxx xxxxxxxx
```

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 ==> tttttt
```

INTERRUPTION (SVC, PROGRAM, or EXTERNAL)

```
*** vvvvvv int code ==> tttttt
```

I/O INTERRUPTION (First line given only if "CSW" was specified):

```
CSW V vadd xxxxxxxx xxxxxxxx R radd yyyyyyyy yyyyyyyy
*** vvvvvv I/O vadd ==> tttttt CSW xxxx
```

BRANCH TRACE: (ALL option selected)

Entry for 'branch from' instruction

```
vvvvvv mnem xxxxxxxx    tttttt
```

Entry for 'branch to' instruction

```
==> vvvvvv mnem xxxxxxxxxxxx
```

Messages

```
DMKTRA002E INVALID OPERAND - operand
DMKTRA003E INVALID OPTION - option
DMKTRA013E CONFLICTING OPERAND - operand
DMKTRA026E OPERAND MISSING OR INVALID
DMKTRA180E TRACE NOT IN EFFECT
```

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   [Printer]  { spoolid   }  { [To]   { *      } }  [Printer]
           [PUnch  ]  { CLass c   }  {        { userid } }  [PUnch  ]
           [Reader ]  { FORM form }  {                   }  [Reader ]
                      { ALL       }  {  From  { ALL    } }
                                     {        { userid } }
```

where:

PRINTER/PRT
PUNCH/PCH
READER/RDR
: specifies the queues to which and from which 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 alphameric 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.

Usage Note:

The TRANSFER command does not transfer any active spool files.

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 VM/SP 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.

		Queue			
	Spool Files	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 print 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

```
(RDR) FILE spoolid TRANSFERRED TO userid (RDR)
<PRT>                                    <PRT>
(PUN)                                    (PUN)
```

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.

```
(RDR) FILE spoolid TRANSFERRED FROM (userid) (RDR)
<PRT>                               <  *   > <PRT>
(PUN)                               (      ) (PUN)
```

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.

`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.

Messages

```
DMKCSV003E INVALID OPTION - option
DMKCSV007E INVALID USERID - userid
DMKCSV008E INVALID SPOOLID - spoolid
DMKCSV020E USERID MISSING OR INVALID
DMKCSV026E OPERAND MISSING OR INVALID

DMKCSV027E SPOOLID MISSING OR INVALID
DMKCSV028E CLASS MISSING OR INVALID
DMKCSV042E SPOOLID nnnn DOES NOT EXIST
DMKCSV053E userid NOT IN CP DIRECTORY
```

VMDUMP

Privilege Class: G

The VMDUMP command dumps virtual storage that VM/SP creates for the virtual machine user, See Usage Note 1. In addition to saving the storage contents of the virtual machine, the VMDUMP command also dumps:

- Virtual program status word (PSW)
- General registers
- Floating point registers
- Control registers
- Storage protection keys
- Virtual machine type identification
- Timer values.

The output produced by the VMDUMP command can be processed by the VM/SP IPCS component. The VM/SP IPCS component assists in debugging errors, as well as stores and maintains error information about the virtual machine. The VM/SP IPCS component displays this information interactively at the user's terminal. VMDUMP provides the VM/SP IPCS component with header information to identify the owner of the dump; it also maintains dump information and writes the dump to a reader file.

For more information on the VM/SP IPCS, refer to *VM/SP Interactive Problem Control System Guide,* Order No. SC19-6204.

VMDUMP also provides you with the ability to send dumps to other users.

Except for the *dumpid operand, you can specify the operands of the VMDUMP command in any order. However, you must specify the *dumpid operand as the last operand on the command line.

The format of the VMDUMP command is:

```
VMDump   ┌hexloc1┐ ┌ ┌ - ┐  ┌hexloc2┐  ┐
         │   0   │ │ │ : │  └  END  ┘  │
         │       │ │ └   ┘             │
         │       │ │ {.}    ┌bytecount┐│
         └       ┘ └        └   END   ┘┘

         ┌TO *     ┐
         │TO userid│
         └SYSTEM   ┘

         [FORMAT vmtype]

         [DSS]

         [*dumpid]
```

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. You may specify as many address ranges as can be contained on one command line each time you issue the VMDUMP command. See Usage Notes 2, 3, and 5.

{- 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 and 4 for an expanded explanation of the use of delimiters. You must specify a value for hexloc2 which 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 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 with the virtual machine type (vmtype) which VM/IPCS 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 validity check the vmtype. Any vmtype longer than eight bytes generates an error message and halts further VMDUMP processing. The dump header record includes your specific vmtype and the VM/IPCS uses the vmtype information to format the dump. If you enter FORMAT, you must also specify a vmtype. If you do

DSS specifies that CP take a dump of all discontiguous saved segments in use by your virtual machine. When 'DSS' is specified, only the discontiguous save 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 virtual machine.) Operating systems running in a virtual machine such as OS/VS, DOS/VS VSE/AF Rel 1, and VSE/AF Rel 2, 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. If the dump is taken using the FORMAT CP operand, the dumpid information will not be available.

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 vmtype 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. Only non-protected storage is dumped.

Responses

COMMAND COMPLETE

is the response indicating normal completion of the VMDUMP function.

Messages

```
DMKVMD009E INVALID RANGE - range
DMKVMD013E CONFLICTING OPTION - option
DMKVMD020E USERID MISSING OR INVALID
DMKVMD033E HEXLOC MISSING OR INVALID[1]
DMKVMD038E PARAMETER parameter FOR option OPTION MISSING OR
           INVALID

DMKVMD053E userid NOT IN CP DIRECTORY
DMKVMD160E HEXLOC hexloc EXCEEDS STORAGE
DMKVME146I VMDUMP FAILED, SPOOLING ERROR
DMKVME174E PAGING ERROR - {I/O|ADDRESS EXCEPTION}:
           {COMMAND COMPLETE|NO DUMP CREATED}
```

[1] This message is issued if the value specified for the number of address ranges in the address list is zero. Return Code = 1C is returned to the user with the message if the error conditions exists for the Diagnose X'94' command.

Appendix A. CP Command Summary

This appendix lists the following:

- The CP commands in alphabetical order
- The IBM-defined privilege classes that can execute the command
- The corresponding function type
- A blank column for you to record the user-defined classes
- A brief description of each command.

Note: Brackets indicate type is optional in the OVERRIDE statement.

Command	Function Type*	IBM-Defined Privilege Class	User-Defined Class	Usage
*	N/A	Any	None	Annotate the console sheet.
#CP	N/A	Any	None	Execute a CP command while remaining in the virtual machine environment.
ACNT	<O>	A		Create accounting records for logged on users, and reset accounting data. ACNT also closes the spool file that is accumulating accounting records.
ADSTOP	<G>	G		Halt execution at a specific virtual machine instruction address.
ATTACH	<R>	B		Logically connect a real device to a virtual machine for that machine's exclusive use or logically connect a DASD device for CP access and control. With CHANNEL operand, dedicate all devices on a particular channel to a specific user.
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.

Command	Function Type*	IBM-Defined Privilege Class	User-Defined Class	Usage
BEGIN	<G>	G		Continue or resume execution of the virtual machine at either a specific storage location or at the address in the current PSW.
CHANGE	S	D		Alter one or more external attributes of a closed spool file or files.
	G	G		Alter one or more attributes of a closed spool file.
CLOSE	<G>	G		Terminate spooling operations on a virtual card reader, punch, printer, or console.
COMMANDS	N/A	Any	None	Display the commands and diagnose codes you are authorized to use.
COUPLE	<G>	G		Connect channel-to-channel devices.
CP	N/A	Any	None	Execute a CP command while remaining in the CMS virtual machine environment.
CPTRAP	<P>	C		Create a reader file of selected trace table, CP interface, and virtual machine interface entries for problem determination.
DCP	<P>	C,E		Display the contents of real storage locations at the terminal.
DEFINE	R	B		Redefine the status of a 3330V volume.
	G	G		Reconfigure your virtual machine.
DETACH	R	B		Remove a real device from the CP system. With the CHANNEL operand, remove a dedicated channel from a user.
	G	G		Detach a virtual device from a virtual machine. Detach a channel from your virtual machine.
DIAL	N/A	Any	None	Connect a terminal or display device to the virtual machine's virtual communication line.
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.

Command	Function Type*	IBM-Defined Privilege Class	User-Defined Class	Usage
DRAIN	<S>	D		Stop spooling operations on a specified real unit record devices after the file currently being processed has been completed.
DUMP	<G>	G		Print the following on the virtual printer: Virtual PSW, general registers, floating-point registers, storage keys, and contents of specified virtual storage locations.
ECHO	<G>	G		Test terminal hardware by redisplaying data entered at the terminal.
ENABLE	<R>	A,B		Enable the previously disabled or nonenabled devices so users may access the system.
EXTERNAL	<G>	G		Simulate an external interruption for a virtual machine and return control to that machine.
FLUSH	<S>	D		Halt and immediately purge or hold the current output on a specified real unit record device.
FORCE	<O>	A		Force a logoff of any user on the system.
FREE	<S>	D		Remove a set of spool files belonging to a specified user from a system hold status.
HALT	<O>	A		Terminate any active channel program on a specified real device.
HOLD	<S>	D		Place user spool files in a system hold status.
INDICATE	O	A		Provides a list of statistics for all users who have the favored execution option.
	A	E		Display, at the console, the use of and contention for system processor and storage.
	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.

Command	Function Type*	IBM-Defined Privilege Class	User-Defined Class	Usage
LOADBUF	<S>	D		On a 1403 printer load the Universal Character Set(UCS) with a specified print chain/train image. On 3203, 3211, 3262, 4245, or 4248 printers, load UCS or Forms Control Buffer(FCB) with a specified image. On 3289 Model 4 printer, load the Font Offset Buffer (FOB) with the image print belt and FCB.
LOADVFCB	<G>	G		Load virtual forms control buffer for a virtual 3203, 3262, 3289E, 3211, 4245, or 4248 printer.
LOCATE	<P>	C,E		Find the addresses of CP control blocks associated with a particular user, a user's virtual device, or a real system device.
LOCK	<O>	A		Permanently locks in selected pages of real storage.
LOGOFF	N/A	Any	None	Disable access to CP.
LOGON	N/A	Any	None	Provide access to CP.
MESSAGE	N/A	Any	None	Transmit messages to other users.
	<O>	A,B		Send message text to a specified user, to primary system operator, or to one or all logged-on users.
MIGRATE	<O>	A		Activate the normal page/swap table migration routines or force a particular user's pages to the secondary device even if that user is currently active.
MONITOR	<O>	A,E		Initiate or override the system-generated monitor function or terminate the recording of events occurring in the real machine.
MSGNOH	<R>	B		Allow a service virtual machine to send messages to specified users without the standard header associated with the MESSAGE command.
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.

Command	Function Type*	IBM-Defined Privilege Class	User-Defined Class	Usage
	R	B		Load, dump, and control operation of a 3704/3705 and control operation of a 3725 control program operating in 270x emulation mode (EP).
NOTREADY	<G>	G		Simulate "not ready"for a device to a virtual machine.
ORDER	S	D		Place closed spool files (of a specified device type) in a different order.
	G	G		Rearrange closed spool files in a specific order.
PER	<G>	A,B,C,D, E,F,G		Monitors certain events in the user's virtual machine as they occur during program execution.
PURGE	S	D		Remove closed spool files from the system before they are printed or punched by the spooling devices or before they are read by a user.
	G	G		Remove closed spool file from the system.
QUERY	O	A		Provide status information on the real or virtual machine and miscellaneous CP functions. Also displays the status of MVS/System Extensions Support.
	R	B		Provide status information on the real or virtual machine and miscellaneous CP functions. Displays the status of the various devices.
	P	C		Provide system log messages and information about system users and processors.
	S	D		Provide system spooling information.
	A	E		Provide status information on the real or virtual machine and miscellaneous CP functions.
	C	F		Provide system log messages and information about system users.
	G	G		Request information about machine configuration and system status.
QVM	<O>	A		Request the transition from VM/SP environment to native mode for a particular virtual machine.

Command	Function Type*	IBM-Defined Privilege Class	User-Defined Class	Usage
READY	<G>	G		Simulate device end interruption for a virtual device.
REPEAT	<S>	D		Increase the number of copies of an output file or place the current output file in a HOLD status increasing or not increasing the number of copies to be created.
REQUEST	<G>	G		Make an attention interruption pending for the virtual machine console.
RESET	<G>	G		Clear and reset all pending interruptions for a specified virtual device and reset all error conditions.
REWIND	<G>	G		Rewind (to load point) a tape and ready a tape unit.
SAVESYS	<A>	E		Save a virtual machine storage space with registers and PSW as they currently exist. Used in the process of creating named systems.
SCREEN	<G>	G		Allows the user to change or alter the color and extended highlighting values for his virtual machine.
SEND	<G>	G		Pass commands and message replies to disconnected virtual machine for processing.
SET	O	A		Establish system parameters and perform various functions to control the CP system and virtual machine options.
	R	B		Change log message, designate the unit to receive system abend dump, change time interval for a specific device class, set off monitoring for a specified class, or terminate all monitoring of missing interruptions.
	A	E		Sets paging and sets the system resource management function.
	C	F		Set the recording mode for a device and for soft errors.
	G	G		Control various functions within the virtual machine.
SHUTDOWN	<O>	A		Systematically end all virtual machine functions and checkpoint the system for an eventual warm start.

Command	Function Type*	IBM-Defined Privilege Class	User-Defined Class	Usage
SPACE	<S>	D		Force the output on the specified printer to be single spaced for the current active spool file, regardless of the carriage control commands the actual file.
SPMODE	<O>	A		Establish or reset the single processor mode environment.
SPTAPE	<S>	D		Dump spool files to tape or load spool files from tape.
START	<S>	D		Restart a spooling device after it has been drained or to change the output class that it may service.
STCP	<P>	C		Alter the contents of real storage but not real PSW or real registers.
SMSG	<G>	G		Send special messages to specified virtual machine.
SPOOL	<G>	G		Alter spooling control options; direct a file to another virtual machine or to a remote location via the RSCS virtual machine.
STORE	<G>	G		Alter specified virtual storage locations and registers.
SYSTEM	<G>	G		Simulate RESET, CLEAR STORAGE, and RESTART buttons on a real system console.
TAG	<G>	G		Specify variable information to be associated with a spool file or output unit record device. Interrogate the current TAG text setting of a given spool file or output unit record device.
TERMINAL	<G>	G		Define or redefine the input and attention handling characteristics of your virtual console.
TRACE	<G>	G		Trace specified virtual machine activity at your terminal, spooled printer, or both.
TRANSFER	S	D		Direct one or more spool files to a specified user or reclaim reader spool files previously sent to one or more users.
	G	G		Transfer input files or reclaim input files from a specified user's virtual card reader.
UNLOCK	<O>	A		Unlock page frames previously locked by a LOCK command.

Command	Function Type*	IBM-Defined Privilege Class	User-Defined Class	Usage
VARY	<R>	B		Mark a device available or unavailable for use by a user or the control program.
VMDUMP	<G>	G		Dump virtual machine (use VM/SP IPCS to view dump.)
WARNING	<O>	A,B		Transmit high-priority messages to a specified user or to all users.

Index

Special Characters

- *
  - command
    - described 52
    - summary 263
- * operand
  - of MESSAGE command 129
  - of SPOOL command 217
- *dumpid operand, VMDUMP command 258
- #CP
  - command
    - commandline operand 71
    - described 71
    - summary 263
- 'extcolor' operand, of SCREEN command 192
- 'exthilight' operand, of SCREEN command 192

A

- abbreviation, use of 47
- accounting, information, displaying 197
- ACNT operand, of SET command 196
- ACNT, command, summary 263
- ADSTOP
  - command
    - described 55
    - hexloc operand 55
    - OFF operand 55
    - summary 263
- AFFINITY operand, of SET command 196
- affinity setting, relation to the logon process 126
- ALL operand
  - of CHANGE command 59
  - of PURGE command 157
  - of QUERY command 161
  - of TRACE command 246
  - of TRANSFER command 253
- altering
  - attributes of
    - virtual printer file 59
    - virtual punch file 59
    - virtual reader file 59
  - logical line-editing symbols 238
  - virtual machine environment 73
- annotating your console sheet 52
- Any user, described 8
- APL operand, of terminal command 239
- APL, control of translation tables 239
- ASSIST operand, of SET command 196
- attached processor application, use of SET ASSIST 196
- attached processor, INDICATE command use for monitor of system load 104
- attention
  - handling, terminal control of 238
  - interrupt
    - entering from your virtual console 57
    - for a virtual machine 188
- ATTN
  - command
    - described 57
    - performs same function as REQUEST command
    - summary 263
- ATTN operand
  - of IPL command 108
  - of TERMINAL command 238
- attributes
  - modifying, for 3800 printer 217
  - spool file, changing 59
  - virtual device, modifying 217
- AUTOCR option, of PARM operand, for IPL command 111
- AUTOPOLL operand, of SET command 196

B

- BACKSPAC, command, summary 263
- BATCH option, of PARM operand, for IPL command 111
- BEGIN
  - command
    - described 58
    - hexloc operand 58
    - summary 263
- BEGIN command, used with the DISCONN command 90
- blanks, as delimiters 4
- block address, used to load a system, at IPL 108
- BOTH operand, of TRACE command 246
- braces, use of 48
- brackets, use of 49
- BRANCH event type of PER command 134
- BRANCH operand, of TRACE command 246
- branch traceback table 39
- branches, tracing 134, 246
- BREAKIN operand, of TERMINAL command 238
- BRKKEY operand, of TERMINAL command 238
- byte alignment, on TERMINAL output 29
- bytecount operand, VMDUMP command 257

C

- CAW (Channel Address Word), displaying 92
- CAW operand, of DISPLAY command 92
- CCW operand, of TRACE command 246
- CCW translation, controlling 200
- CHANGE
  - command
    - ALL operand 59
    - CHARS operand 59
    - CLASS operand 59
    - COPY operand 59
    - COPY* operand 59
    - described 59
    - DIST operand 59
    - FCB operand 59
    - FLASH operand 59
    - FORM operand 59
    - HOLD operand 59
    - MODIFY operand 59
    - NAME operand 59
    - NOHOLD operand 59
    - PRINTER operand 59
    - PUNCH operand 59
    - READER operand 59
    - summary 263
- changing
  - control registers 228
  - filename of a spool file 59
  - floating-point registers 228
  - general registers 228
  - Program Status Word (PSW) 228
  - spool class of a file 59
  - spool file attributes 59
  - spoolid of a file 59
  - virtual storage locations 228
- channel c operand, of DETACH CHANNEL command 86
- channel-to-channel adapters, virtual, connecting 69
- CHANNELS operand
  - of DEFINE command 73
  - of QUERY command 161
- character
  - set 3
  - translation tables, APL, controlling 238
- CHARDEL operand, of TERMINAL command 238
- CHARS operand
  - of CHANGE command 59
  - of SPOOL command 217
- class
  - privilege, for CP commands 7
  - spool, changing for a file 59
  - virtual device spool, modifying 217
- class Any, described 10
- class G, described 9
- CLASS operand
  - of CHANGE command 59
  - of ORDER command 132
  - of PURGE command 157
  - of QUERY command 161
  - of SPOOL command 217
  - of TRANSFER command 253
- CLEAR operand
  - of IPL command 108
  - of SYSTEM command 232
- clearing
  - interrupts from a virtual machine 189
  - storage to zero 108
- CLOSE
  - command
    - CONSOLE operand 64
    - described 64
    - DIST operand 64
    - dsname operand 64
    - fnft operand 64
    - FORM operand 64
    - HOLD operand 64
    - NAME operand 64
    - NOHOLD operand 64
    - PRINTER operand 64
    - PUNCH operand 64
    - PURGE operand 64
    - READER operand 64
    - summary 263
    - vaddr operand 64
  - HOLD restrictions 64
  - NOHOLD restrictions 64
  - restrictions 64
- CLOSE operand, of SPOOL command 217
- closing
  - files 64
  - TRACE output 34
  - virtual devices 217
- CMD option of the PER command 41, 134
- code operand, of EXTERNAL command 103
- command
  - name 4
  - operands 4
- command environment
  - CP 3
  - defined 3
- commandline operand
  - of #CP command 53
  - of CP command 71
- commands
  - ADSTOP command, halting execution of virtual machine 28
  - ATTACH command, with dedicated devices 21
  - ATTN command, simulates real console function 25
  - CHANGE command, altering spool file attributes 17
  - CLOSE command, using with spool files 17
  - command
    - described 68
  - CP summarized 263
  - DEFINE command
    - reconfiguring virtual machine 27

temporary disks 15
DETACH CHANNEL command, with dedicated channels 21
DETACH command
removing dedicated devices 21
temporary disks 15
DISPLAY command, displaying virtual storage 29
DUMP command, printing virtual storage 31
EXTERNAL command, simulating interruptions 24
LINK command, sharing virtual disks 16
MESSAGE command, obtaining operator assistance 21
notational conventions 47
ORDER command, reordering spool files 20
PURGE command, deleting spool files 20
QUERY command, determining device or file attributes 17
REQUEST command, simulates real console function 25
SPOOL command, altering device attributes 17
STORE command, altering virtual storage 31
summary of use 263
TRACE command, follow virtual machine activities 33
used to control a terminal session, summary of 263
comments, how to write 4
CONMODE operand, of TERMINAL command 238
connecting
remote terminals to virtual machines 87
virtual channel-to-channel adapters 69
console
input/output, terminating 253
sheet, annotation of 52
spool file, modifying 254
spooling, controlling 254
CONSOLE operand
of CLOSE command 64
of DEFINE command 73
of QUERY command 183
of SPOOL command 217
CONT operand, of SPOOL command 217
continuous reading of an entire file 217
Control Program, console function mode 26
control, functions, setting for your virtual machine 196
controlling
a terminal session, summary of commands used 263
3270 display 239
controlling virtual machine, during terminals session 11
COPY operand
of CHANGE command 59
of SPOOL command 217
COPY* operand
of CHANGE command 59
of SPOOL command 217
COUNT sub-command of the PER command 39, 135
COUPLE
command
described 69
summary 263
userid operand 69
vaddr1 operand 69
vaddr2 operand 69
CP
command
commandline operand 71
described 71
summary 263
CP command environment 3
CP commands
entering
from CP console function mode 26
entering from the virtual machine environment 71
privilege class for 7
privilege class, described 8
text format of 51
CP operand, of SEND command 194
CPLEVEL operand, of QUERY command 183
CPTRAP, command, summary 263
CPUID operand, of QUERY command 161
CPUID operand, of SET command 196
creating disk space with DEFINE command, precautions 73
CSW (Channel Status Word), displaying 92
CSW operand
of DISPLAY command 92
of TRACE command 246
CTCA operand, of DEFINE command 73

D

DASD operand, of QUERY command 161
DCP, command, summary 263
debugging, programs 27
debugging, using PER 134
dedicated channel, detached 86
DEFINE
command
CHANNELS operand 73
CONSOLE operand 73
CTCA operand 73
described 73
GRAF operand 73
LINE operand 73
PRINTER operand 73
PUNCH operand 73
READER operand 73
STORAGE operand 73
summary 263
TFB-512 operand 73
TIMER operand 73
T2305 operand 73
T2314 operand 73
T2319 operand 73
T3310 operand 73
T3330 operand 73
T3340 operand 73

T3350 operand 73
T3370 operand 73
T3375 operand 73
T3380 operand 73
1403 operand 73
1443 operand 73
2501 operand 73
2540P operand 73
2540R operand 73
3203 operand 73
3211 operand 73
3262 operand 73
3289E operand 73
3505 operand 73
3525 operand 73
3800 operand 73
3800-1 operand 73
3800-3 operand 73
4248 operand 73
DEFINE command
temporary disks 15
use with MSS devices 73
virtual addresses 73
defining virtual devices 73
DETACH
command
described 83
summary 263
vaddr operand 83
DETACH CHANNEL
command
channel c operand 86
described 86
DETACH command, temporary disks 15
detaching
dedicated channels 86
virtual devices 83
DEV operand, of TAG command 234
device-end interrupt pending for a virtual device, specifying 187
devices
virtual
linking to 113
simulating not ready status for 131
DIAL
command
described 87
summary 263
userid operand 87
vaddr operand 87
DIAL command, device restriction 87
directory entry, controlling, virtual machine attributes 13
DISABLE, command, summary 263
DISCONN
command
described 90
HOLD operand 90
summary 263
disconnecting
virtual console 90
virtual machine 122
DISPLAY
command
CAW operand 92
CSW operand 92
described 92
PSW operand 92
summary 263
DISPLAY command, byte alignment 29
displaying
accounting information 196
Channel Address Word (CAW) 92
Channel Status Word (CSW) 92
floating point registers 92
general registers 92
Program Status Word (PSW) 92
storage keys 92
virtual storage 92
virtual storage locations 92
DIST operand
of CHANGE command 59
of CLOSE command 64
distribution code, for an output file 59
DMCP, command, summary 263
DRAIN, command, summary 263
dsname operand, of CLOSE command 64
DSS operand, VMDUMP command 257
DUMP
command
described 98
summary 263

E

ECHO
command
described 102
nn operand 102
summary 263
ECMODE operand, of SET command 196
editing symbols
controlling 238
logical, controlling use of 196
EMSG operand, of SET command 196
ENABLE, command, summary 263
END operand, of TRACE command 246
END sub-command of PER command 134
entering
a CP command from a virtual machine environment 71
CP command 3
EOF operand, of SPOOL command 217
error message handling, controlling 196
ESCAPE operand, of TERMINAL command 238
execution
halting at an instruction address 55

of a virtual machine, resuming 58
extended control mode, setting 196
EXTERNAL
command
code operand 103
described 103
summary 263
external interrupt, simulated 103
EXTERNAL operand, of TRACE command 246

F

FB-512 device information, obtained with QUERY command 170
FCB operand
of CHANGE command 59
of LOADVFCB command 119
of SPOOL command 217
FILE operand, of TAG command 234
files
changing the spool class of 59
changing the spoolid for 59
closing 64
distribution code for 59
holding before output processing 59
holding from further processing 64, 217
holding via CLOSE 217
purging 64, 217
purging via SPOOL 217
reclaiming 253
reordering closed spool files 132
spool, changing filename for 59
terminating processing of 64
transferring to another user 217
transmitting to a virtual reader 253
FILES operand, of QUERY command 161
first-level storage, requesting via DISPLAY command 94
FLASH operand
of CHANGE command 59
of SPOOL command 217
floating-point registers
changing your virtual machine 228
displaying 92
printing 98
FLUSH, command, summary 263
fn ft operand, of CLOSE command 64
FOR operand, of SPOOL command 217
FORCE, command, summary 263
FORM operand
of CHANGE command 59
of CLOSE command 64
of ORDER command 132
of SPOOL command 217
of TRANSFER command 253
PURGE command 157
FORMAT vmtype operand, VMDUMP command 257
forms controls for a virtual printer, specifying 119
FREE, command, summary 263
FROM operand, of TRANSFER command 253

G

G event type of PER command 134
general registers
changing in your virtual machine 228
displaying 92
printing 98
general user, described 9
GET sub-command of PER command 134
GRAF operand
of DEFINE command 73
of QUERY command 183

H

HALT, command, summary 263
halting, execution at an instruction address 55
hexloc operand
of ADSTOP command 55
of BEGIN command 58
hexloc1 operand, VMDUMP command 258
hexloc2 operand, VMDUMP command 258
hiding your password 125
HILIGHT operand, of TERMINAL command 238
HOLD operand
of CHANGE command 59
of CLOSE command 64
of DISCONN command 90
of LOGOFF command 122
of SPOOL command 217
HOLD, command, summary 263
holding
a file before output processing 59
files by user hold status 217
files from further processing 64, 217

I

I/O devices, virtual, defining for your virtual machine 73
I/O operand, of TRACE command 246
IMSG operand, of SET command 196
INDEX operand, for LOADVFCB command 119
INDICATE
command
attached processor load indication 104
described 104
LOAD operand 104
multiprocessor load indication 104

summary 263
USER operand 104
initial program load (IPL)
automatic, suppression of 125
of a virtual machine operating system 108
stopping 108
input/output operations, tracing 246
INSTRUCT event type of PER command 134
INSTRUCT operand, of TRACE command 246
internal print duplication, by 3800 printer 59
interrupt codes
X'1004', clock comparator 24
X'1005', CPU timer 24
interruptions, simulated 24
interrupts
clearing from a virtual machine 189
device-end, specifying virtual 187
external, simulating 103
for a virtual machine 188
range of external codes 103
tracing 246
IPL
command
ATTN operand 108
CLEAR operand 108
described 108
nnnnnn operand 108
NOCLEAR operand 108
PARM operand 108
STOP operand 108
summary 263
SYSTEMNAME operand 108
vaddr operand 108
IPL command, for named systems 108
ISAM operand, of SET command 196

K

keys
program function, controlling 196
storage
displaying 92
printing 98

L

line editing symbols, controlling 196
line length, controlling at your terminal 238
LINE operand, of DEFINE command 73
LINEDEL operand, of TERMINAL command 238
LINEDIT operand, of SET command 196
LINEND operand, of TERMINAL command 238
LINES operand, of QUERY command 161
LINESIZE operand, of TERMINAL command 238
LINK
command
described 113
password operand 113
summary 263
userid operand 113
vaddr operand 113
LINK command
batch job restrictions 113
passwords 113
2305 restrictions 113
linking, a device to your virtual machine 113
LINKS operand, of QUERY command 161
load key, simulation of via IPL command 108
LOADBUF, command, summary 263
loading a system, with a block address, at IPL 108
loading, an operating system 22
LOADVFCB
command
described 119
FBC operand 119
INDEX operand 119
NAME operand 119
summary 263
vaddr operand 119
LOCATE, command, summary 263
LOCK, command, summary 263
logging off your virtual machine 122
logging on your virtual machine 125
logical editing symbols
altering 239
controlling 239
controlling use of 196
logically connected terminal 87
LOGMSG operand, of QUERY command 161
LOGOFF
command
described 122
HOLD operand 122
summary 263
LOGON
command
described 125
NOIPL operand 125
password operand 125
summary 263
userid operand 125
logon procedure, hiding your password during 125

M

MASK event type of PER command 134
masking, password during logon 125
Mass Storage Systems
See MSS, virtual volumes, defined with DEFINE command
merging trace elements 155

MESSAGE
- command
  - * operand 129
  - described 129
  - msgtext operand 129
  - OPERATOR operand 129
  - summary 263
  - userid operand 129

message handling, error, handling 196
messages
- controlling transmission of 196
- sending to other users 129

MODE operand, of TERMINAL command 238
MODIFY operand
- of CHANGE command 59
- of SPOOL command 217

modifying
- spooling control options 217
- storage, using ADSTOP command 55
- virtual device attributes 217

MONITOR, command, summary 263
MSG operand, of SET command 196
MSGNOH, command, summary 263
msgtext operand
- for MESSAGE command 129
- of SMSG command 215

MSS, virtual volumes, defined with DEFINE command 73
multiprocessor, INDICATE command to display system load 104

N

NAME operand
- of CHANGE command 59
- of CLOSE command 64

named systems, IPL command for 108
NAMES operand, of QUERY command 183
nesting, described 49
NETWORK, command, summary 263
nn operand, of ECHO command 102
nnnnnn operand, of IPL command 108
NOCLEAR operand, of IPL command 108
NOCONT operand, of SPOOL command 217
NOEOF operand, of SPOOL command 217
NOHOLD operand
- of CHANGE command 59
- of CLOSE command 64
- of SPOOL command 217

NOIPL operand, of LOGON command 125
NORUN operand, of TRACE command 246
notational conventions 47
NOTERM operand, of SPOOL command 217
NOTRANS operand, of SET command 196
NOTREADY
- command
  - described 131
  - summary 263
  - vaddr operand 131

O

OFF operand
- of ADSTOP command 55
- of SPOOL command 217
- of TRACE command 246

operands, command 4
operating system
- for a virtual machine, passing parameters to 108
- initial program load for 108

operating systems
- loading 22
- passing parms at IPL 22

OPERATOR operand, of MESSAGE command 129
operator, sending messages to 129
options on trace elements 155
ORDER
- command
  - CLASS operand 132
  - described 132
  - FORM operand 132
  - PRINTER operand 132
  - PUNCH operand 132
  - READER operand 132
  - spoolid operand 132
  - summary 263

P

PAGEX operand, of SET command 196
parameters, passing to a virtual machine operating system 108
PARM operand, of IPL command 108
PASS option of the PER command 134
passing parameters to your virtual machine operating system 108
password
- hiding during logon procedure 125
- invalid format for commandline, precaution 125
- with the LINK command 113

password operand
- of LINK command 113
- of LOGON command 125

PER
- command
  - BOTH option 134, 140
  - BRANCH event type 134
  - CMD option 41, 134, 140
  - COUNT sub-command 39, 134
  - DAT option 134, 140
  - DATOFF option 134, 140
  - described 134

END sub-command 134
FROM option 134, 140
G event type 134
GET sub-command 134
GUESTR option 43, 134, 140
GUESTV option 43, 134, 140
INSTRUCT event type 134
MASK event type 134
NORUN option 134, 140
PASS option 134, 140
PRINTER option 134, 140
program debugging, using 38
RANGE option 134, 140
RUN option 134, 140
SAVE sub-command 134
selectivity, with 36
STEP option 134, 140
storage alteration 42
STORE event type 134
summary 263
suspending 37
TABLE sub-command 134
TERMINAL option 134, 140
terminating 37
using 35
PFnn operand
of QUERY command
of SET command 196
power off vs logoff, with 3270 terminal 122
print file requirements, for 3800 printer 61
printer files, virtual, altering the attributes of 59
PRINTER operand
of CHANGE command 59
of CLOSE command 64
of DEFINE command 73
of ORDER command 132
of QUERY command 161
of SPOOL command 217
of TRACE command 246
PURGE command 157
PRINTER option of PER command 134
printer, virtual, specifying forms controls for 119
printing
contents of virtual machine components 98
Program Status Word (PSW) 98
virtual storage keys 98
virtual storage locations 98
printing first-level storage, with DUMP command 98
PRIV operand, of TRACE command 246
privilege classes
for CP commands
defined 7
summarized 8
privileged instructions, tracing 246
PROCESS operand, of QUERY command 161
processor, determining, attached or main 159
program debugging using PER 38
program function keys, controlling 196
PROGRAM operand, of TRACE command 246
Program Status Word (PSW)
changing your virtual machine 228
displaying 92
printing 98
PROMPT operand, of TERMINAL command 238
PSW operand, of DISPLAY command 92
PUNCH
operand
of CHANGE command 59
of PURGE command 157
of QUERY command 161
punch files, virtual, altering the attributes of 59
PUNCH operand
of CLOSE command 64
of DEFINE command 73
of ORDER command 132
of SPOOL command 217
PURGE
command
ALL operand 157
CLASS operand 157
described 157
FORM operand 157
PRINTER operand 157
PUNCH operand 157
READER operand 157
spoolid operand 157
summary 263
PURGE operand
of CLOSE command 64
of SPOOL command 217
purging
closing files 64
spooling files 217
purging closed spool files 157

Q

QUERY
command
ALL operand 161
CHANNELS operand 161
CLASS operand 161
CONSOLE operand 161
CPLEVEL operand 161
CPUID operand 161
DASD operand 161
described 161
FILES operand 161
GRAF operand 161
LINES operand 161
LINKS operand 161
LOGMSG operand 161
NAMES operand 161
PER operand 161
PFnn operand 161
PRINTER operand 161

    - PROCESS operand 161
    - PUNCH operand 161
    - READER operand 161
    - SCREEN operand 161
    - SECUSER operand 161
    - SET operand 161
    - spoolid operand 161
    - STORAGE operand 161
    - summary 263
    - TAPES operand 161
    - TBL operand 161
    - TERMINAL operand 161
    - TIME operand 161
    - UR operand 161
    - USERID operand 161
    - USERS operand 161
    - vaddr operand 161
    - VIRTUAL operand 161
    - VMSAVE operand 161
- QUERY operand, of TAG command 234

R

- read/write access, with LINK command 113
- reader files, virtual, altering the attributes of 59
- READER operand
  - of CHANGE command 59
  - of CLOSE command 64
  - of DEFINE command 73
  - of ORDER command 132
  - of PURGE command 157
  - of QUERY command 161
  - of SPOOL command 217
- reading, an entire file continuously 217
- READY
  - command
    - described 187
    - summary 263
    - vaddr operand 187
- real computer
  - reset button, simulating 232
  - restart button, simulating 232
- reclaiming files, in order to purge 157
- reclaiming transmitted files 255
- reconfiguring, your virtual machine 27
- registers
  - extended control
    - displaying 92
    - printing 98
  - floating-point
    - displaying 92
    - printing 98
  - general
    - displaying 92
    - printing 98
- remote spooling networks
  - receiving files 226
  - transmitting files 225
- remote spooling, TAG command 234
- remote terminal, connecting to a virtual machines 87
- removing a virtual device from your virtual machine 83
- REPEAT, command, summary 263
- REQUEST
  - command
    - described
    - performs same function as ATTN command 57
    - summary 263
- reserved IBM class, described 7
- RESET
  - command
    - described 189
    - summary 263
    - vaddr operand 189
- reset button, simulating 232
- RESET operand, of SYSTEM command 232
- response
  - for #CP command 54
  - for ADSTOP command 56
  - for CHANGE command 62
  - for CLOSE command 67
  - for COMMANDS command 68
  - for COUPLE command 69
  - for DEFINE command 81
  - for DETACH CHANNEL command 86
  - for DETACH command 84
  - for DIAL command 88
  - for DISCONN command 91
  - for DISPLAY command 96
  - for DUMP command 101
  - for ECHO command 102
  - for INDICATE LOAD 104
  - for INDICATE USER 106
  - for LINK command 116
  - for LOGOFF command 123
  - for LOGON command 126
  - for MESSAGE command 130
  - for ORDER command 133
  - for PER command 154
  - for PURGE command 158
  - for QUERY ALL command 166
  - for QUERY CHANNELS command 166
  - for QUERY CONSOLE command 166
  - for QUERY CPUID command 166
  - for QUERY DASD command 166
  - for QUERY FILES command 166
  - for QUERY GRAF command 166
  - for QUERY LINES command 166
  - for QUERY LINKS command 166
  - for QUERY PFnn command 166
  - for QUERY PROCESSOR command 166
  - for QUERY READER, PRINTER, and PUNCH command 166
  - for QUERY SCREEN command 166
  - for QUERY SECUSER command 166
  - for QUERY SET command 166
  - for QUERY SPMODE command 166

for QUERY STORAGE command 166
for QUERY S370E command 166
for QUERY TAPES command 166
for QUERY TERMINAL command 166
for QUERY TIME command 166
for QUERY UR command 166
for QUERY USERID command 166
for QUERY vaddr command 166
for QUERY VMSAVE command 166
for RESET command 189
for REWIND command 190
for SPOOL command 227
for STORE command 231
for SYSTEM command 232
for TAG command 236
for TRACE command 249
for TRANSFER command 255
for VMDUMP command 261
responses to CP commands, types of 51
responses vs system messages 51
restart button, simulating 232
RESTART operand, of SYSTEM command 232
resume
execution of your virtual machine 58
use of your virtual machine after disconnecting it 90
REWIND
command
described 190
summary 263
vaddr operand 190
rewinding a real tape 190
RUN operand
of SET command 196
of TRACE command 246

S

SCREEN
command
'extcolor' operand 191
'exthilight' operand 191
described 191
summary 263
SCREEN operand, of QUERY command 161
SCRNSAVE operand, of TERMINAL command 239
SCROLL operand, of TERMINAL command 239
SECUSER operand, of QUERY command 161
selectivity, using PER 36
SEND
command
CP operand 194
described 194
summary 263
text operand 194
userid operand 194
sending messages
to other users 129
to the operator 129
service representative, described 9
SET
command
ACNT operand 196
AFFINITY operand 196
ASSIST operand 196
AUTOPOLL operand 196
CPUID operand 196
described 196
ECMODE operand 196
EMSG operand 196
IMSG operand 196
ISAM operand 196
LINEDIT operand 196
MSG operand 196
NOTRANS operand 196
PAGEX operand 196
PFnn operand 196
RUN operand 196
SMSG operand 196
STBYPASS operand 196
STMULTI operand 196
summary 263
TIMER operand 196
VMSAVE operand 196
370E operand 196
SET operand, of QUERY command 161
sharing virtual disks 16
SHUTDOWN, command, summary 263
simulating
not ready status for a virtual device 131
the reset button on a real computer 232
the restart button on a real computer 232
SIO operand, of TRACE command 246
SLEEP
command
described 213
summary 263
SMSG
command
described 215
msgtext operand 215
summary 263
userid operand 215
SMSG operand, of SEND command 194
SPACE, command, summary 263
specific trace elements 155
specifying, device-end interrupt for a virtual device 187
SPOOL
command
* operand 217
CHARS operand 217
CLASS operand 217
CLOSE operand 217
CONSOLE operand 217
CONT operand 217
COPY operand 217
COPY* operand 217

described 217
EOF operand 217
FCB operand 217
FLASH operand 217
FOR operand 217
FORM operand 217
HOLD operand 217
MODIFY operand 217
NOCONT operand 217
NOEOF operand 217
NOHOLD operand 217
NOTERM operand 217
OFF operand 217
PRINTER operand 217
PUNCH operand 217
PURGE operand 217
READER operand 217
START operand 217
STOP operand 217
summary 263
SYSTEM operand 217
TERM operand 217
TO operand 217
userid operand 217
vaddr operand 217
spool class
for a file, changing 59
modifying 217
SPOOL command, use with TAG command 225
spool file
altering TAG information 234
changing filename for 59
changing the attributes of 59
characteristics
changing 17
querying 17
setting 17
reordering 132
spoolid 224
TAG command 234
transmitting locally 225
transmitting to remote location 225, 234
spoolid number 224
spoolid operand
of ORDER command 132
of PURGE command 157
of QUERY command 161
of TRANSFER command 253
spooling
considerations during abend 217
considerations during shutdown 217
control, options, modifying 217
operator, described 8
virtual console 19
spooling output to a 3800 printer 223
SPTAPE, command, summary 263
START operand, of SPOOL command 217
START, command, summary 263
status
virtual machine, determining 161
words, tracing 246
STATUS operand
STORE command 228
STORE command, simulating hardware store 33
STOP operand
of IPL command 108
of SPOOL command 217
STOP, command, summary 263
stopping, initial program load (IPL) procedure 108
storage
alteration tracing using PER 42
clearing to zeros 108
keys, printing 98
locations, virtual, displaying
modification of 55
virtual, altering 31
storage keys, displaying 92
storage locations
virtual machine
altering 228
displaying 92
printing 98
STORAGE operand
of DEFINE command 73
of QUERY command 161
STORE
command
described 228
STATUS operand 228
summary 263
STORE command, STATUS operand, simulate hardware store 33
STORE event type of PER command 134
storing
CPU status for virtual machine 228
virtual machine data 228
summary, of CP commands 263
suppressing, automatic IPL of a virtual machine operating system 125
suspending PER 37
SVC operand, of TRACE command 246
SYSTEM
command
CLEAR operand 232
described 232
RESET operand 232
RESTART operand 232
summary 263
system analyst, described 8
system messages vs CP command responses 51
SYSTEM operand
of SPOOL command 217
of VMDUMP command 257
system operator, primary, described 8
system programmer, described 8
system resource operator, described 8
system resources, indication of 104

T

TABCHAR operand, of TERMINAL command 238
TABLE sub-command of PER command 134
TAG
  command
    described 234
    DEV command 234
    FILE operand 234
    QUERY command 234
    summary 263
TAG command
  altering TAG information 234
  transmitting files to remote location 234
  use with SPOOL command 217
TAPES operand, of QUERY command 161
TBL operand, of QUERY command 161
temporary disk space, formatting and clearing 73
temporary, virtual disks 15
TERM operand, of SPOOL command 217
TERMINAL
  command
    APL operand 239
    ATTN operand 239
    BREAKIN operand 239
    BRKKEY operand 239
    CHARDEL operand 239
    CONMODE operand 239
    described 239
    ESCAPE operand 239
    HILIGHT operand 239
    LINEDEL operand 239
    LINEND operand 239
    LINESIZE operand 239
    MODE operand 239
    PROMPT operand 239
    SCRNSAVE operand 239
    SCROLL operand 239
    summary 263
    TABCHAR operand 239
    TEXT operand 239
    TYPE operand 239
TERMINAL operand
  of QUERY command 161
  of TRACE command 246
TERMINAL option of the PER command 134
terminals
  console, disconnecting from your virtual machine 90
  control of attention handling 238
  input/output processing, controlling 238
  line length, controlling 234
  remote, connecting to a virtual machine 87
  session
    altering virtual storage 31
    changing spool file characteristics 17
    commands used to control, summary of 263
    controlling I/O functions 13
    controlling virtual machine 22
    debugging programs 27
    dedicated channels 21
    dedicated devices 20
    displaying virtual storage 28
    logging on 22
    purging spool files 20
    querying spool file characteristics 17
    reordering spool files 20
    setting spool file characteristics 16
    simulating interruptions 24
    spooling 16
    testing programs 27
    tracing virtual machine activity 33
    virtual console spooling 19
    virtual disks 15
    virtual disks, permanent 15
    virtual disks, temporary 15
    virtual unit record devices 16
  testing using the ECHO command 102
terminating
  console input/output 217
  PER 37
  processing of files 64
testing
  programs 27
  terminal 102
text format, of CP commands 51
text operand, of SEND command 194
TEXT operand, of TERMINAL command 239
TEXT, control of translation tables 238
TFB-512 operand, of DEFINE command 73
time, determining length for a terminal session 159
TIMER operand
  of DEFINE command 73
  of SET command 196
TIMER operand, of QUERY command 161
TO * operand, VMDUMP command 257
TO operand
  of SPOOL command 217
  of TRANSFER command 253
TO userid operand, VMDUMP command 257
TRACE
  command
    ALL operand 246
    BOTH operand 246
    BRANCH operand 246
    CCW operand 246
    CSW operand 246
    described 246
    END operand 246
    EXTERNAL operand 246
    I/O operand 246
    INSTRUCT operand 246
    NORUN operand 246
    OFF operand 246
    PRINTER operand 246
    PRIV operand 246
    PROGRAM operand 246
    RUN operand 246

SIO operand 246
summary 263
SVC operand 246
TERMINAL operand 246
TRACE output, closing 34
tracing storage alteration using PER 42
tracing virtual machine activity 33, 246
TRANSFER
command
ALL operand 253
CLASS operand 253
described 253
FORM operand 253
FROM operand 253
spoolid operand 253
summary 263
TO operand 253
userid operand 253
transferring
files
to another user 217
to a virtual reader 253
translating, CCWs, controlling 196
transmitting
spool files
locally 225
to remote locations 225
TYPE operand, of TERMINAL command 238
T2305 operand, of DEFINE command 73
T2314 operand, of DEFINE command 73
T2319 operand, of DEFINE command 73
T3310 operand, of DEFINE command 73
T3330 operand, of DEFINE command 73
T3340 operand, of DEFINE command 73
T3350 operand, of DEFINE command 73
T3370 operand, of DEFINE command 73
T3375 operand, of DEFINE command 73
T3380 operand, of DEFINE command 73

U

underscore, use of 48
UNLOCK, command, summary 263
uppercase (letters and words) and special symbols, use of 48
UR operand, of QUERY command 183
USER operand
LOAD operand, of INDICATE command 104
of INDICATE command 104
of QUERY command 183
userid operand
of COUPLE command 69
of DIAL command 87
of LINK command 113
of LOGON command 125
of MESSAGE command 129
of SEND command 194
of SMSG command 215
of SPOOL command 217
of TRACE command 246
USERID operand, of QUERY command 161
using the PER command 35
using virtual forms control buffer
with the 3262 printer 119
with the 3289E printer 119

V

vaddr operand
of CLOSE command 64
of DETACH command 83
of DIAL command 87
of IPL command 108
of LINK command 113
of LOADVFCB command 119
of NOTREADY command 131
of QUERY command 161
of READY command 187
of RESET command 189
of REWIND command 190
of SPOOL command 217
vaddr1 operand, of COUPLE command 69
vaddr2 operand, of COUPLE command 69
VARY, command, summary 263
virtual console
disconnecting from your virtual machine 90
read environment 26
spooling 19, 217
virtual devices
closing 217
defining 73
detaching from your virtual machine 83
determining number of your virtual machine 183
determining the status of 183
I/O, defining for your virtual machine 73
removing from your virtual machine 83
simulating not ready status for 131
specifying a device-end interrupt for 187
virtual disks, permanent 15
virtual disks, temporary 15
virtual machine
activity, monitoring 134
activity, tracing 246
configuration, altering 73
determining status of 161
device, linking to 113
directory entry 13
disconnecting 122
execution, resuming 58
logging off of 122
logging on 125
operating system, passing parameters to 108
placing in a dormant state 213
reconfiguring 27

sending and processing special messages 215
setting control functions for 196
storing information from 228
tracing activity 33
virtual machine environment, entering a CP command from 71
for CP command 72
VIRTUAL operand, of QUERY command 161
virtual printer, specifying forms controls for 119
virtual storage location
changing 228
printing 98
virtual storage, altering 31
virtual timer, controlling 196
virtual unit record devices 16
VMDUMP
command
*dumpid operand 258
bytecount operand 258
described 258
DSS operand 258
FORMAT vmtype operand 258
hexloc1 operand 258
hexloc2 operand 258
summary 263
SYSTEM operand 258
TO * operand 258
TO userid operand 258
VMSAVE operand
of QUERY command 161
of SET command 196

W

WARNING, command, summary 263
writing, comments in VM/370 4

Z

zeros
clearing storage to via IPL 108
clearing storage to via SYSTEM command 232

Numerics

1403 operand, of DEFINE command 73
1443 operand, of DEFINE command 73
2501 operand, of DEFINE command 73
2540P operand, of DEFINE command 73
2540R operand, of DEFINE command 73
3203 device
as DEFINE operand 73
LOADVFCB restrictions 119
3203 operand, of DEFINE command 73
3203 printer, virtual, specifying forms controls for 73
3211 operand, of DEFINE command 73
3211 printer, virtual, specifying forms controls for 73
3262 operand, of DEFINE command 73
3262 printer, loading a virtual forms control buffer 73
3270 device message length 215
3270 terminal, power off vs logoff, precautions 122
3289E operand, of DEFINE command 73
3289E printer, loading a virtual forms control buffer 73
3505 operand, of DEFINE command 73
3525 operand, of DEFINE command 73
3800 operand, of DEFINE command 73
3800 printer
modifying device attributes 217
modifying spool file attributes 59
of SPOOL command 217
spooling output to one 223
using CHANGE command 59
3800-1 operand, of DEFINE command 73
3800-3 operand, of DEFINE command 73
3850 Mass Storage Systems
See MSS, virtual volumes, defined with DEFINE command
4245 operand, of DEFINE command 73
4248 operand, of DEFINE command 73

CP Command Reference for
General Users
Order No. SC19-6211-3

READER'S COMMENT FORM

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

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

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

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

- What is your occupation? ______________________________

- How do you use this publication:

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

Your comments:

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

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

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

SC19-6211-3

Reader's Comment Form

Fold and Tape   Please Do Not Staple   Fold and Tape

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

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

POSTAGE WILL BE PAID BY ADDRESSEE:

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

Fold   Fold

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

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

IBM®

Cut or Fold Along Line

VM/SP CP Command Reference for General Users (File No. S370/4300-36) Printed in U.S.A. SC19-6211-3

SC19-6211-3

SC19-6211-3

SC19-6211-3