IBM

Virtual Machine/System Product

SC19-6211-05

CP General User Command Reference

Release 6

IBM

Virtual Machine/System Product

SC19-6211-05

CP General User Command Reference

Release 6

Sixth Edition (July 1988)

This edition, SC19-6211-05, is a major revision of SC19-6211-04 and applies to the Virtual Machine/System Product Release 6, program number 5664-167, and to all subsequent releases of this product until 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, 30xx, 4300, and 9370 Processors Bibliography*, GC20-0001, for the editions that are applicable and current.

Summary of Changes

For a list of changes, see "Summary of Changes" on page 265.

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 licensed program in this publication is not intended to state or imply that only IBM's licensed program 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 Corporation, Information Development, 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.

The form for reader's comments provided at the back of this publication may also be used to comment on the VM/SP online HELP facility.

Preface

This book describes the VM/SP Control Program (CP) commands that are available to IBM-defined privilege class G and class Any. Information about class A, B, C, D, E, and F commands, previously included in the *VM/SP CP Command Reference*, is contained in the *VM/SP CP System Command Reference*, Order No. SC24-5402.

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 class G commands described in this publication, general users can reconfigure their virtual machines, control devices attached to their virtual machines, perform input and output spooling functions, and simulate many other functions of a real computer console. CP commands enabling system operators, system programmers and analysts, and service representatives to manage the resources of the system are contained in *VM/SP CP System Command Reference*.

This publication is organized in two parts: "Part 1. Guidance Information" and "Part 2. Reference Information."

Part 1 has three chapters:

"Chapter 1. The CP Command Structure" describes the VM/SP command environments and the general structure of the CP command language. It also describes the notational conventions that are used to display command formats, truncations, and abbreviations.

"Chapter 2. 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.

"Chapter 3. CP Privilege Classes" describes the IBM-defined class structure based on function types.

Part 2 has two chapters:

"Chapter 4. Format of CP Commands" contains reference data for all the class G and class Any CP 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.

"Chapter 5. CP Command Summary" contains all of the class G and class Any CP commands with a brief description of each command's function.

This publication also has one appendix:

"Appendix A. Logical Line Editing Symbols" contains usage information for the VM/SP logical line editing symbols.

Contents

Part 1. Guidance Information 1

Chapter 1. The CP Command Structure 3
VM/SP Command Environments 3
Entering CP Commands 3
Notational Conventions 5

Chapter 2. CP Command Usage 9
Controlling Input and Output Functions 11
Controlling the Virtual Machine 20
Testing and Debugging of Programs 24

Chapter 3. CP Privilege Classes 41

Part 2. Reference Information 43

Chapter 4. Format of CP Commands 45
* 47
#CP 48
ADSTOP 50
ATTN 52
BEGIN 53
CHANGE 54
CLOSE 59
COMMANDS 64
COUPLE 65
CP 67
DEFINE 68
DETACH 77
DETACH CHANNEL 81
DIAL 82
DISCONN 85
DISPLAY 87
DUMP 93
ECHO 97
EXTERNAL 98
INDICATE 99
IPL 103
LINK 109
LOADVFCB 114
LOGOFF 117
LOGON 120
MESSAGE 124
NOTREADY 126
ORDER 127
PER 129
PURGE 149
QUERY (ALL) 151
QUERY (G) 156

READY 182
REQUEST 183
RESET 184
REWIND 185
SCREEN 186
SEND 190
SET 192
SLEEP 209
SMSG 211
SPOOL 213
STORE 224
SYSTEM 227
TAG 229
TERMINAL 233
TRACE 242
TRANSFER 249
VMDUMP 254

Chapter 5. CP Command Summary 259

Appendix A. Logical Line Editing Symbols 263

Summary of Changes 265
Structural Changes 265
Technical Changes 265

Glossary of Terms and Abbreviations 269

Bibliography 281
Prerequisite Publications 281
Corequisite Publications 281

Index 287

Part 1. Guidance Information

This part of the publication provides general guidance information about the Control Program (CP) component of the Virtual Machine/System Product (VM/SP) system. It contains three chapters:

"Chapter 1. The CP Command Structure" describes the VM/SP command environments and the general structure of the command languages.

"Chapter 2. CP Command Usage" provides usage information for some General User (class G) commands.

"Chapter 3. CP Privilege Classes" describes the IBM-defined class structure for CP commands.

Chapter 1. The CP Command Structure

The CP component of VM/SP has its own command language. Much like any other language, the CP command language has a set of rules and structures. This chapter describes the CP command language environment, syntax, and structure, giving general descriptions of command names, operands and their positions on the command line, and character set usage in CP.

Depending on your privilege class, you can enter CP system commands to do various system oriented functions. Privilege class is discussed in "Chapter 3. CP Privilege Classes."

You, as a class G or general user, can use CP commands to simulate the function of the system operator's console. You can enter commands to control the operating system running in your virtual machine much as an operator controls the entire VM/SP operating system using the keys and switches on the system console. A discussion of CP commands other than class G and class Any is included in *VM/SP CP System Command Reference.*

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 enter CP commands.

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

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). If you are in the CP command environment, VM/SP responds to a null line by displaying CP. If you are operating under CMS, VM/SP responds by displaying VM READ in the status area. When you enter a null line again, VM/SP displays CMS and the status area changes to RUNNING.

Entering CP Commands

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.

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.

Command Operands

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 all the class G and class Any commands with all of their operand requirements in "Chapter 4. Format of CP Commands" of this publication.

You must write the operands in the order in which they appear in the command formats in Chapter 4, 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 the following table.

Table 1. Character Sets and Their Contents

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

Notational Conventions

The notation that defines the command syntax for this publication consists of commands, subcommands, variables, and symbols.

Commands, Subcommands, and Variables

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

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

  Example:

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

  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, you must enter the entire word (represented by all capital letters).

- You can also abbreviate the name of some commands 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,

  Reader indicates that the minimum truncation is R.

  In the discussion of the READER operand that follows the format box,

  Reader
  RDR indicates that the abbreviation is RDR.

  Thus, you enter **READER** as **R**, **RE**, **REA**, **READ**, **READE**, **READER**, or **RDR**.

Lowercase letters, words, and symbols that appear in the command format box in *italics* represent variables for which you will substitute specific information. For example,

hexloc indicates that you should enter a hexadecimal storage location address.

Symbols Used in Command Descriptions

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

- Braces { }

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

  Example:

  $$\left\{\begin{matrix}\mathbf{A}\\ \mathbf{B}\\ \mathbf{C}\end{matrix}\right\}$$

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

- Brackets [ ]

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

  Example:

  $$\left[\begin{matrix}\mathbf{A}\\ \mathbf{B}\\ \mathbf{C}\end{matrix}\right]$$

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

- Ellipsis ...

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

  Example:

  {*spoolid1 spoolid2...*}
  means that you can code more than the number of *spoolids* shown.

- Underscore __

  Defaults are underscored.

  Example:

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

  means that $\underline{\mathbf{B}}$ is the default. You can choose **A**, **B**, or **C**. However, if you do not enter any choice, the system will assume **B**.

Use the following symbols as specified in the format box:

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

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

Example:

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

Lets you select a file by:

- File name only
- File name and file type only
- File name, file type, and file mode.

You cannot select file type alone because file type is nested within file name. The same is true if you want to select file mode; you must also select file name and file type.

Chapter 2. CP Command Usage

This chapter 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 the user's virtual machine.

Throughout the rest of this book, the text is interspersed with examples of VM/SP commands and system responses. All user input is shown in lowercase, and all system responses are shown in uppercase and mixed case.

Class G Commands

Before you can activate your virtual machine, you must fill out a request provided by your installation to establish the following:

- User ID
- User classes
- Passwords
- Storage requirements
- Disk space
- I/O devices.

Your installation builds an entry in the VM/SP directory for you from this information.

Commands in the general user class let you control the operation of a virtual machine by using its virtual operator's console device. For example, commands are included in the general user class that let you perform the following operations:

Table 2. General User Tasks and Commands

Task	Command
Activate virtual machine	LOGON
Load an operating system	IPL
Start or restart program execution in the virtual machine	BEGIN
Transfer and store data	LINK, SPOOL, STORE, and TRANSFER
Record and control data	CHANGE, CLOSE, DISPLAY, DUMP, ORDER, PURGE, QUERY, and SPOOL
Reconfigure your virtual machine	DEFINE and DETACH
Communicate with your virtual machine	#CP, ADSTOP, ATTN, BEGIN, EXTERNAL, IPL, REQUEST, SET
Handle errors	TRACE, DISPLAY, DUMP, PER
Stop virtual machine execution at a specified virtual storage address	ADSTOP
Reset and restart the virtual machine	SYSTEM RESET and SYSTEM RESTART
Store and display data contained in virtual storage, registers, and status words	STORE and DISPLAY
Display storage keys, the CSW, and the CAW	DISPLAY
Clear virtual storage to zeros	SYSTEM CLEAR
Simulate an external interruption for the CPU timer or external interruption button on the system console	EXTERNAL
End session	LOGOFF

Class Any Commands

Class Any commands comprise those commands available, with the same format and operands, to all user classes. Some of the tasks and commands are:

Table 3. Class Any Tasks and Commands

Task	Command
Gain initial access to the VM/SP system and set up your virtual machine	LOGON
Establish a logical connection between your terminal and a previously logged-on multiple-access virtual machine.	DIAL
Execute a CP command while remaining in the virtual machine environment	#CP
Disconnect your terminal from your virtual machine	DISCONN
Display the commands and diagnose codes you are authorized to use	COMMANDS
Communicate with the system operator or any other logged-on user.	MESSAGE

Controlling Input and Output Functions

The VM/SP 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. If you run CMS in the virtual machine, the I/O configuration may also include a reference to a *file pool.* A *file pool* is another form of DASD space that CMS manages. More about CMS and file pools is in the *VM/SP CMS User's Guide.* A typical VM/SP directory entry, which happens to be for a CMS user, would look like this:

```
(1)     USER VMUSER 123456 768K 2M G
        ACCOUNT 87321 BIN14           (2)
(3)     OPTION ECMODE   LANG FRANC    (4)
        IPL CMS PARM FILEPOOL VMSYSU: (5)
        CONSOLE 01F 3215              (6)
(7)     SPOOL 00C 2540 R
        SPOOL 00D 2540 P
        SPOOL 00E 1403
(8)     MDISK 130 3380 050 099 VDISK1 WR RWPASSWD
        LINK CMSSYS 440 190 RR
```

where:

(1) The user ID of this virtual machine is VMUSER, and the password is 123456. The virtual machine's storage size is defined as 768K; however, this can be redefined up to 2 MB if, during an operating session, the need for additional storage arises (see "Reconfiguring the Virtual Machine" on page 24). 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 lets the virtual machine use the complete set of virtual System/370 control registers and the dynamic address translation feature of the System/370.

(4) Option LANG FRANC specifies the language to be set for the virtual machine when the user logs on. (The language FRENCH has the language ID (*langid*) FRANC.) The user of this virtual machine receives most system messages in French and can enter CMS commands in French (assuming all the French language files are properly set up).

(5) The IPL CMS control statement causes CMS to be loaded into the virtual machine when it is logged on. The option PARM FILEPOOL VMSYSU: establishes part of the I/O configuration for CMS. It identifies the VMSYSU file pool as the user's default file pool when the user logs on, and the user's top directory is automatically accessed as filemode A. See the *VM/SP CMS User's Guide* for more about SFS directories.

(6) 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.

(7) The virtual unit record devices (reader, punch, and printer) are spooled and their addresses, as far as the virtual machine is concerned, are 00C, 00D, and 00E, respectively.

(8) 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 3380 volume labeled VSDISK1. 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 CMS functions such 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 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 "Directory Control Statements" in the *VM/SP Planning Guide and Reference.*

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

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

Virtual Disks

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 that you may or may not wish to have other users access; or, they can 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 with 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 program.

When you have no further need for temporary disk space, you can release the space to the system by entering 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 lets CP 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 owner's user ID and the virtual address of the disk. If the owner is controlling access to the disk, you need the read or write password. You can then enter 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, see the *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, see the discussion of the LINK command in Chapter 4 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 dedicating 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 let you change some or all of the preceding 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 spool ID 0001, CP will assign the next available spool ID to the new file. One series of spool IDs covers the reader, printer, punch, and console spool files. VM/SP usually assigns the spool ID 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 spool ID at the same time.

An exception also occurs for reader, printer, and punch files. When a converted file (a file with a conversion status of COMPLETE) is changed with the CHANGE command, purged with the PURGE command, or transferred with the TRANSFER command, a new SFBLOK is obtained and a new spool ID is assigned. The new SFBLOK is marked unconverted and has pointers to the print data. The old SFBLOK is marked to be purged. See "Special Processing for Spool Files" on page 17 for additional information.

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 enter 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 enter

```
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 enter 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 spool ID of the spool file, you can enter the command

```
query printer all
```

and one-line descriptions for all printer spool files that have not been processed or selected for processing are displayed at the terminal. The information includes the spool ID number, file name, file type, and other spool file characteristics.

The DEST option lets you select a specific printer or punch to process your print, punch, or console file. For example,

```
spool printer dest dest1
```

prints your file at a printer handling output for a particular destination (*dest1*).

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, but only a few require printing, you can hold all files with the HOLD option. You can then release only the required files with the NOHOLD option of the CHANGE or CLOSE commands.

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

```
close printer nohold
```

This overrides the HOLD status of the SPOOL command for that one file, placing it 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 spool ID numbers of all the closed printer files. Then enter the CHANGE command to alter the HOLD status of each file to be released. For example, if the file with a spool ID 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 enter

```
spool reader nohold
```

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

```
purge reader all
```

Again, you can enter the CHANGE command with the *spoolid* number to control the HOLD status of specific reader files.

The CONT/NOCONT status of a spooling device controls the concatenation of files. Because this is a characteristic of a spooling device, it can be changed only with 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.

Special Processing for Spool Files

Some applications, such as Advanced Function Printing (AFP), use spool files in a special way. Therefore, the files can be converted or unconverted. An unconverted file is one in the usual CP spool file format. A converted file is one on which an application using the SPOOL System Service has done special processing. An example of such an application is the Spool File Conversion Machine (SFCM) part of the Print Services Facility (PSF) for AFP.

A spool file has a conversion status of one of the following:

- NOTCONV - The file is in regular CP spool file format.
- ACTIVE - The file is undergoing conversion now.
- COMPLETE - The file is converted.

A converted file stays in the VM/SP queue until the application completes processing it or a command affecting the file is issued. If the virtual machine that drives the printer is inoperable for a long time, the system operator can use the FORCE option of the PURGE command to purge all files from the queue. The UNCONV option of the CHANGE command marks the file as unconverted.

The CP CHANGE, TRANSFER, and PURGE commands cause the following actions:

- The file gets a new spool file descriptor block with a new spool file ID. The new descriptor block is marked unconverted and points to the spool file data. (The spool file is again eligible for conversion.)

- The descriptor block with the original spool file ID is marked as purged and does not point to the spool file data.

To find out the conversion status of a file, specify the PSF option with the QUERY PRINTER, PUNCH, or READER command.

Reordering and Purging Spool Files

If you want to change the sequence of your closed spool files, enter the ORDER command to place your files in a specific order by device type. Files can be resequenced by spool ID or class or both with 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 your printer, use the PURGE command. Files can be specified individually by spool ID or by class, or both. For example,

```
purge prt 1234 class b
```

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

```
purge punch all
```

removes all punch files.

Entering

```
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 be used only as dedicated devices.

The directory program can dedicate a device through a DEDICATE control statement in the VM/SP directory entry for a virtual machine at system generation, or the system operator's ATTACH command can dynamically dedicate a device.

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 with the DETACH command:

```
detach 495
```

The format of the DEDICATE control statement is described in detail in the *VM/SP Planning Guide and Reference*. More than one user can have the same real device specified as dedicated to his or her 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 does not restrict you to a particular real device. If a malfunction precludes using the volume on one DASD, the disk pack can be transferred to another DASD and 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. Because 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 enters the command

```
attach 281 to smith as 181
```

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

```
TAPE 181 ATTACHED
```

When you have no further use for the device, enter 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 or her virtual machine. CP does not translate device addresses because the virtual addresses must be the same as the real device addresses. Contention for use of the channel is minimized, because all of the channel resources are dedicated to a single virtual machine. The ATTACH CHANNEL command is a class B command that 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 enters

```
attach channel 2 to smith
```

and, if 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 enter 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 dynamically reconfigure your virtual machine 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 with 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 advisable to use automatic loading. If the VM/SP directory entries for your user ID 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 enter the LOGON command as

```
logon userid noipl
```

In this 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 with 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 390 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 entering the IPL command with the STOP operand:

```
ipl 390 clear stop
```

At this point, you can enter CP commands to display or alter data in your nucleus. To restart the virtual machine, enter 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--for 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 with 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, see the IPL command "Usage Notes" on page 106 in Chapter 4.

Simulating Interrupts

You can use the EXTERNAL command to simulate an external interruption to the virtual machine and return control to that virtual machine. If you enter

```
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 2 bytes at location X'86'. An external interruption is then presented to your virtual machine, and your operating system determines subsequent action.

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.

Interrupting the Virtual Machine to Enter CP Commands

There are several ways that you can interrupt the running of your virtual machine 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.

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 you can enter multiple CP command lines.

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
```

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 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 entered — 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 enter 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 define only 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) is issued, and the directory is not updated. Define these devices on different virtual control units and reenter the DIRECT command.

Testing and Debugging of Programs

In addition to the testing and debugging facilities 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. It cannot be directly handled. Second-level storage is storage that is real to the virtual machine; it can be directly handled. It 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. Third-level storage is storage that appears virtual to your virtual machine. The V=V partitions or regions of DOS/VS, VSE systems, and OS/VS constitute third-level storage. Third-level storage generally cannot be directly specified; the user or the virtual operating system must convert any third-level storage addresses to second-level storage addresses before including them in the specific command line. An exception to this is the PER command. (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 entered.
- An IPL or a system reset is performed.
- The ADSTOP OFF command is entered.

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

With the DISPLAY command, you can display on your terminal the contents of virtual storage, storage keys, general registers, floating-point registers, control registers (if in EC mode), PSW, CAW, and CSW. With the DUMP command, you can print all but the CAW and CSW on the spooled virtual printer. 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 Chapter 4. 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:

- 4-byte groups, aligned on fullword boundaries, hexadecimal format, printed 4 fullwords per line

- 16-byte groups, aligned on 16-byte boundaries, hexadecimal format, printed 4 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, enter

```
dump 0
```

To 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 after an asterisk:

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

To print the dump data on the real printer you must first close the virtual printer. Enter 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

```
   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 (that is, storage that is real to your virtual machine), 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 fewer than eight 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, issues error message DMKCDS005E Invalid hexdata - C, and does not store the last character.

You can store data into one or more 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 fewer 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, enter the command

```
set ecmode on
```

Be aware that this command resets your virtual machine and that 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 to 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 book *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 want to record the trace without stopping after each output line, add the RUN operand as the last entry on the command line.

If you have specified multiple activities to be traced and then decide to stop tracing one or more of them, enter

```
trace program branch off
```

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 enter 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 Chapter 4.

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 Chapter 4.

Using the PER Command

The PER command can 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 the events 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 X'500' bytes in length. For example, if you wish to monitor all instructions within your program, you can enter

```
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 determine which 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 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 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 five 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 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 OC    D607C3A2C3AA 0203A8 0204B0 CC=1
```

and then program execution halts, and the CP command environment is 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 clear your screen frequently. The frequency with which events are displayed can be changed with the PASS option. Ordinarily, every successful event is displayed. However, 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 only 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, a number precedes each trace element:

```
query per
```

```
1 INSTRUCT PRINTER RUN
2 G5 DATA 00000023 TERMINAL NORUN
3 G7 TERMINAL NORUN
4 STORE INTO 020628-020630 TERMINAL NORUN
```

This number can selectively end PER tracing; for example, 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 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 as usual 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 trace program interrupts. However, the trace information is displayed after the interrupt has occurred and cannot always 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.

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 five 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 enter 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 six 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. You can now restart the program using PER to trace the execution of instructions in the range beginning with this branch instruction and ending at the program interrupt address.

PER COUNT Subcommand

Another method of finding the failing instruction is to use the PER COUNT subcommand with TRACE. This method, as well as the use of the PER TABLE 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 enter 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, enter 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 enter 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 through 204FF are not displayed. The 2145th instruction is displayed. When the instruction is displayed, enter

```
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 enter

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

and start the program. When the failure occurs, enter 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 described previously to get close to the problem.

PER Command Option

The PER CMD option can 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 an instruction in the range 20000 through 204FF alters any storage within the range 204F0 through 204FF.

Also, the CMD option can 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 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 entering

```
per branch terminal run cmd display g14-15
```

or both commands can be changed to DISPLAY general registers 14 and 15 by entering

```
per cmd display g14-15
```

Storage Alteration

PER can trace the alteration of storage in the user's virtual machine. If you specify PER STORE, then whenever an instruction places a value into storage, that event is traced. This value need *not* differ from the previous value.

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

```
per store into 20100 data 112757
```

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

It is also possible to monitor only cases in which the storage value actually changes. For example,

```
per mask into 20100 data ffffffffffffffff
```

monitors instructions that actually change one or more bytes in the doubleword field starting at location 20100. PER MASK can also 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, you can 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 entered, then both second- and third-level activity is traced (that is, both CP and CMS). To trace only the CP activity (second level) in the range 20000 through 21000, enter

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

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

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

You can 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
```

Chapter 3. CP Privilege Classes

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

CP Privilege Classes

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

- Operations
- Resource
- Programmer
- Spooling
- Analyst
- CE (Customer Engineer - Service)
- General.

The IBM-defined class structure is based on these seven function types. A command keeps its function type even if your installation establishes its own class structure. In most cases, each command class (A-G) has a corresponding function type (O, R, P, S, A, C, or G). Some commands fall into more than one class. Table 4 on page 42 shows the function of each privilege class and function type. See "Chapter 5. CP Command Summary" for a list of all class G and class Any CP commands, their IBM-defined privilege classes, function types, and descriptions of what each command does. If your installation changes the class for a command, record the change in the User-Defined Class column in the summary table.

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

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

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

The CP class Any commands are available to all user classes. These commands perform basic functions required by all virtual machines, such as

- Logging on
- Logging off
- Sending messages.

If you try to issue a command not allowed for your command class, CP does not execute the command and issues an error message. The CP command descriptions for class G and class Any are in Chapter 4 in this book.

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

The following table shows the different privilege classes, the function codes, and the major tasks that can be performed for each privilege class.

Table 4. CP Privilege Classes

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

Part 2. Reference Information

Part 2 contains reference information for all class G and class Any CP commands. It consists of two sections:

"Chapter 4. Format of CP Commands" contains reference information for all the 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.

"Chapter 5. CP Command Summary" alphabetically lists and briefly describes class G and class Any CP commands.

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.

Chapter 4. Format of CP Commands

How to Use this Chapter

This section describes CP commands available to class G and class Any users. Privilege class G commands control the functions of the general user's virtual machine. Privilege classes A, B, C, D, E, and F are assigned to users who manage the VM/SP system and its spooling and hardware resources; commands for these privilege classes are contained in the *VM/SP CP System Command Reference.*

The command names appear at the top of each page for easy reference. The following commands are available to class G or Any and to some classes other than class G:

- CHANGE
- DEFINE
- DETACH
- INDICATE
- MESSAGE
- ORDER
- PURGE
- QUERY
- SET
- TRANSFER.

The formats or options for these commands depend on the privilege class. The versions of these commands available to other than class G users are contained in the *VM/SP CP System Command Reference.*

Note: The Group Control System (GCS) operator commands are described in the *VM/SP Group Control System Command and Macro Reference.* The Transparent Services Access Facility (TSAF) operator commands and the APPC/VM VTAM Support (AVS) operator commands are described in the *VM/SP Connectivity Planning, Administration, and Operation* manual. Operator commands for file pool server machines are described in the *VM/SP CMS Shared File System Administration* manual.

Command Description Formats

The command formats are described in alphabetical order except for special characters such as the asterisk (*) and the pound sign (#), which appear first. The descriptions of each command in this chapter have the following sections:

1. **Privilege Class** — States the IBM-defined privilege class for the command.
2. **Format** — Shows the command format. See "Notational Conventions" on page 5 for information about abbreviations, and so forth.
3. **Operands** — Lists and explains the operands for the command.
4. **Usage Notes** — Lists notes about the command that pertain to more than one subcommand or that pertain to specific hardware, and so forth. Some command descriptions may not require this section.
5. **Responses** — Lists the standard system responses from invoking the various commands and subcommands. Some command descriptions may not include

this section. In such cases, there is no system response, or the system response is self-explanatory.

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

6. **Messages** — Lists error messages issued by the commands. Most of the messages are issued in the language that is set for your virtual machine. However, some of these messages will always be displayed in English, no matter what language is set for your virtual machine. See the *VM/SP System Messages and Codes* for detailed information about the messages. See the *VM/SP System Messages Cross-Reference* for a listing of messages.

The return codes associated with each command directly correspond to the message numbers. For example, if you received a return code of 22 when executing the LINK command, you could look at the description for message number 022:

```
DMKLNM022E Virtual device address missing or invalid
```

*

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).

Format

*	*anycomment*

Response

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.

Format

#CP	[*commandline1* [#*commandline2* #...]]

Operand

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 line-end symbol. If you have redefined the logical line-end symbol, #CP is an invalid command; you must substitute your line-end symbol for the pound sign when using this command.
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 4 bytes of data. If the READ is for fewer than 3 bytes, you must use the attention key to enter the CP environment.
3. You can enter multiple *commandlines* 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) the current logical line-end symbol separates all *commandlines* to be executed.
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 enter 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.

6. The following examples show some of the ways you can use the #CP command. The ¢ is the logical line-delete symbol. For more information on logical line editing symbols, see Appendix A, "Logical Line Editing Symbols" on page 263.

Examples

Table 5. System Action to #CP Command

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	CP interprets "data entered" as an invalid command and issues a message, "Unknown CP command: DATA."
Data entered¢#CP	"Data entered" is ignored. You enter CP environment.
#CP query files¢data entered	QUERY command is ignored; "data entered" passes to the virtual machine.

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.

Format

ADSTOP	*hexloc* **OFF**

Operands

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

OFF
cancels any previous ADSTOP setting.

Usage Notes

1. When execution halts, the CP command mode is entered and a message is displayed. At this point, you may invoke other CP debugging commands. To resume operation of the virtual machine, issue the BEGIN command. Once an ADSTOP location is set, it is 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 protected shared segment. If a protected shared segment is modified by an ADSTOP command, a warning message is issued to the user and the virtual machine is given a private copy of the named system. Other users of the segment continue to operate with an unchanged copy of that segment.

     Also, be aware that using the ADSTOP command within an unprotected shared segment changes the data in storage and can cause problems for other users who have accessed the segment. This problem could arise, for example, when several service machines have write access to a GCS common segment.
   - Use the TRACE command with the INSTRUCT, BRANCH, or ALL operands when a traced instruction is located at the ADSTOP address.

3. Address stop (ADSTOP *hexloc*) is specified as a six-character virtual machine real address. For OS/VS or DOS/VS, the specified address must be in fixed storage. Movement of the page containing the ADSTOP can cause erroneous processing because the resulting SVC X'B3' is presented to the virtual machine for handling.
4. If the SVC handling portion of the virtual machine assist feature is enabled on your virtual machine, CP turns it off when an ADSTOP is set. When the address stop is removed, CP returns the assist feature SVC handling to its previous status.
5. 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 - *operand*
DMKCFD021E	Real device address missing or invalid
DMKCFD022E	Virtual device address missing or invalid
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.

Format

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).

Format

Begin	[*hexloc*]

Operand

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.
3. The execution of the BEGIN command causes the console input buffer to return to the state it was in when the virtual machine stopped.

Response

The virtual machine begins execution.

Message

DMKCFJ004E Invalid hexloc - *operand*

CHANGE

Privilege Class: G

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

Format

```
                                                   ┌                                  ┐1
                                                   │ CHars name1  [[CHars] name2]      │2
                                                   │              [[CHars] name3]      │
                                                   │              [[CHars] name4]      │
                                                   │ CLass c2                          │
                                                   │ COpy [*] nnn                      │
                                                   │ DIst distcode                     │
             { Reader  }   { CLass c1   }          │ FCB name²                         │
  CHange     { Printer }   { FORM form1 }         <  FLash name nnn                     >
             { PUnch   }   { DEST dest1 }          │ FORM form2                        │
                           { spoolid    }          │ DEST dest2                        │
                           { ALL        }          │ [HOLD  ]                          │
                                                   │ [NOHOLD]                          │
                                                   │ MOdify name [n]²                  │
                                                   │ UNCONV                            │
                                                   └                                  ┘

                                                             [NAme {fn [ft] }]
                                                             [     {dsname  }]
```

[1] You *must* specify one of these operands; however, you can specify more than one. They can 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 usually used in this publication.

[2] The CHARS, FCB, and MODIFY options are valid only for the 3800 printer.

Operands

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. See the *VM/SP Operator's Guide* for a detailed description of output 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.

DEST *dest1*
is a one- to eight-character alphameric destination name for printers and punches. Use the destination name to select those files you want to change.

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 entered with the NOHOLD operand specified.

NOHold
releases the specified file from user HOLD status.

DIst *distcode*
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 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 as usual without modification. The *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 is required only if *nameN* conflicts 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*.

DEST *dest2*

changes the spool destination name of the file to *dest2*.

NAme *fn* [*ft*]

assigns identification to the spool file in the CMS format file name and file type. The field *fn* is a one- to eight-character alphameric file name assigned to the file for identification. The field *ft* is a one- to eight-character alphameric file type assigned to the file for identification. If *ft* is not specified, the file type 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).

UNCONV

marks the spool file as unconverted. An unconverted file is one in the usual CP spool file format. A converted file is one on which an application using the SPOOL System Service has done special processing.

Usage Notes

1. Enter 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 print certain classification titles automatically (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 or she 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. 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 as usual. 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 with the GENIMAGE command. The MVS Character Conversion Aid can also convert existing customized character sets to the 3800 Model 3 pel density.

Responses

```
{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.

Note: When the target spool file has the CONVERSION status of complete, the changed spool file has a new *spoolid* associated with it. A message is issued indicating that the *spoolid* has been changed. For example:

```
SPOOLID 0003 HAS BEEN CHANGED TO 0008
SPOOLID 0004 HAS BEEN CHANGED TO 0009
```

Messages

DMKCSU003E	Invalid option - *option*
DMKCSU006E	Invalid device type - *devtype*
DMKCSU008E	Invalid spoolid - *spoolid*
DMKCSU013E	Conflicting option - *option*
DMKCSU024E	{CYLS|BLKS|Form number|DEST} missing or invalid
DMKCSU026E	Operand missing or invalid
DMKCSU027E	Spoolid missing or invalid
DMKCSU028E	Class missing or invalid
DMKCSU029E	Filename and/or filetype missing or invalid
DMKCSU030E	Copies missing or invalid
DMKCSU032E	Distribution 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
DMKCSU424E	Reader file spool error; file unchanged

CLOSE

Privilege Class: G

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

Format

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

Operands

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 can 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 1 on page 62). 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 1 on page 62). 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 the receiving virtual machine user changes it to NOHOLD.

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 usually purged from the virtual machine.

PUrge

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

FORM *form*

specifies the form name for a file. The *form* operand is a one- to eight-character alphanumeric form name that your installation defines during system generation. If FORM is not specified for the CLOSE command, the file created has the form name associated with the last SPOOL command applied to it. The form name is changed for this file only and does not affect other files.

DEST *dest*

specifies the destination name for a file. The *dest* operand is a one- to eight-character alphameric destination name for printers and punches.

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 file name and file type. The file name, *fn*, is a one- to eight-character alphameric name assigned to the file for identification. The file type, *ft*, is a one- to eight-character alphameric type assigned to the file for identification. If *ft* is not specified, the file type is set to blanks. See Usage Note 6.

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). QUERY displays only 18 characters of the 24-character *dsname*, even though a name of up to 24 characters is valid. See Usage Note 6.

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 enter a CLOSE READER command 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 (with 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 you did not SPOOL continuous (CONT), any attributes that were set with the SPOOL or TAG command are preserved for use with the next print file. In general, attributes such as CHAR and FCB, set with 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 file name and file type and an optional distribution code to aid in later identification of the file and its contents.

6. If you are spooling continuously (CONT), the CLOSE command does not close the file, but can change the hold status, distribution code, form, destination name, and name associated with the file. Additional CLOSE commands can override these options, and additional print commands can override the name assigned in the preceding CLOSE command.

7. Figure 1 on page 62 shows what happens to an input file in your virtual reader when the CLOSE command is entered with the HOLD or NOHOLD operand in effect. The CLOSE command can result in an input file being purged or saved.

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

Figure 1. Action Taken on CLOSE for a Virtual Input Device.

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

8. Figure 2 shows what happens to a file in your virtual output device when the CLOSE command is entered 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	No action
CLOSE HOLD	File held	File held	No action	No action
CLOSE NOHOLD	File released for processing	File released for processing	No action	No action

Figure 2. Action Taken on CLOSE for a Virtual Output Device

Note: No action is taken until you enter a SPOOL {PRT|PUN} NOCONT command. Then you can enter a CLOSE command to release the file for processing.

Response

```
┌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 IMSG option is OFF, this message is not issued even if the preceding conditions are met.

Messages

DMKCSQ003E	Invalid option - *option*
DMKCSQ006E	Invalid device type - *devtype*
DMKCSQ013E	Conflicting option - *option*
DMKCSQ022E	Virtual device address missing or invalid
DMKCSQ024E	{FORM number|DEST} missing or invalid
DMKCSQ029E	Filename and/or filetype missing or invalid
DMKCSQ032E	Distribution code missing or invalid
DMKCSQ040E	Device *vdev* does not exist

COMMANDS

Privilege Class: Any

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

Format

COMMands	

Usage Note

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.

Response

```
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.

Format

COUPLE	*vaddr1* [To] *userid* *vaddr2*

Operands

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* can 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*.

Usage Note

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

Responses

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

This is the response you receive when you enter 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 enter 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 entered 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 entered the COUPLE command receives if the virtual CTC `vaddr1` was already coupled when the COUPLE command was entered. The previous connection is terminated. This response is always followed by the response:

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

Messages

DMKDIB006E	Invalid device type - *devtype*
DMKDIB011E	Invalid device type - *userid vdev*
DMKDIB020E	Userid missing or invalid
DMKDIB022E	Virtual device address missing or invalid
DMKDIB040E	Device *vdev* does not exist
DMKDIB045E	*userid* not logged on
DMKDIB047E	*userid vdev* does not exist
DMKDIB058E	CTC *vdev* busy on *userid*

CP

Privilege Class: Any

Use the CP command as an optional header on commands to CP in a CP environment.

Format

CP	[*commandline1* [#*commandline2* #...]]

Operand

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

Usage Notes

1. The pound sign (#) shown in the format block represents the logical line-end symbol currently in effect for your virtual machine. If you have redefined the logical line-end symbol, you must substitute your line-end symbol for the pound sign when using this command.
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 them with the logical line-end symbol (#).
4. An attention key that is pressed while in full-screen mode (user mode) is not interrogated by CP, but is passed to the full-screen application for processing. Exceptions to this are the PA1 key and the break key (SCRNSAVE). Therefore, if you enter #CP *commandline* while in full-screen mode, the input line is passed to the application, and CP does not process the command. To execute a CP command while in full-screen mode, use PA1 or the break key first 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.

Format

```
DEFine  { { CONsole }
        { { CTCa    }
        { { Printer }
        { { PUnch   }
        { { Reader  }
        { { TIMer   }
        { { 1403    }
        { { 1443    }
        { { 2501    }
        { { 2540P   }  [As]  vaddr
        { { 2540R   }
        { { 3088    }
        { { 3203    }
        { { 3211    }
        { { 3262    }
        { { 3289E   }
        { { 3505    }
        { { 3525    }
        { { 4245    }
        { { 4248    }
        {
        { { 3800   }                            [ 2Wcgm ] [ BTS ] [ DAtck   ]
        { { 3800-1 }[As]  vaddr  [Size ww 11]   [ 4Wcgm ] [ CFS ] [ NODatck ]
        { { 3800-3 }
        {
        {   CHANnels  [As]  { SEL }
        {                   { BMX }
        {
        {   GRAF    cuu    [ 3036 ]
        {                  [ 3066 ]
        {                  [ 3138 ]
        {                  [ 3148 ]
        {                  [ 3158 ]
        {                  [ 3270 ]
        {
        {   LIne  [As]   vaddr  [ IBM 1  ]
        {                       [ TELE 2 ]
        {
        {   Printer  [As]  vaddr  [ 1403   ]
        {   PRT                   [ 1443   ]
        {                         [ 3203   ]
        {                         [ 3211   ]
        {                         [ 3262   ]
        {                         [ 3289E  ]
        {                         [ 3800   ]
        {                         [ 3800-1 ]
        {                         [ 3800-3 ]
        {                         [ 4245   ]
        {                         [ 4248   ]
```

```
 ┌                                                        ┐
 │ ┌TFB-512┐                                              │
 │ │T3310  │                                              │
 │ │T3370  │                                              │
 │{│T9313  │} [As]  vaddr  [BLK]  [nnnnnn]                │
 │ │T9332  │                                              │
 │ │T9335  │                                              │
 │ └       ┘                                              │
 │ ┌      ┐                                               │
 │ │T2305 │                                               │
 │ │T2314 │                                               │
{  │T2319 │  [As]  vaddr  [CYL]   [nnnn]                   }
 │{│T3330 │}                                              │
 │ │T3340 │                                               │
 │ │T3350 │                                               │
 │ │T3375 │                                               │
 │ │T3380 │                                               │
 │ └      ┘                                               │
 │  STORage [As]   {nnnnnK}                               │
 │                 {nnM   }                               │
 │  vaddr 1   [As]   vaddr 2                              │
 └                                                        ┘
```

Operands

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.

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.

Reader [As] *vaddr*
RDR
adds a spooling 2540 card reader 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 not the INDEX feature. 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.

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.

$$\left\{\begin{matrix}\textbf{3800}\\ \textbf{3800-1}\\ \textbf{3800-3}\end{matrix}\right\}[\textbf{As}]\ \textit{vaddr}\quad [\textbf{Size}\ \textit{ww}\ \textbf{11}]\ \begin{bmatrix}\textbf{2Wcgm}\\ \textbf{4Wcgm}\end{bmatrix}\begin{bmatrix}\textbf{BTS}\\ \textbf{CFS}\end{bmatrix}\ \begin{bmatrix}\textbf{Datck}\\ \textbf{NODatck}\end{bmatrix}$$

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-inch 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 by 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, there is enough storage for CP to load 128 characters in the 3800; if the user chooses the default, 4WCGM, the user 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 CP reflects data checks for the 3800 to the virtual machine. If the user 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 also depresses the reflection of some load checks that are the result of invalid translation tables.

Note: The user should specify the DATCK keyword only when it is absolutely necessary because 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.

$$\textbf{CHANnels}\quad [\textbf{As}]\left\{\begin{matrix}\textbf{SEL}\\ \textbf{BMX}\end{matrix}\right\}$$

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 can 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.

```
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.

```
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.

```
┌TFB-512┐
│T3310  │  [As] vaddr [BLK] [nnnnnn]
┤T3370  ├
│T9313  │
│T9332  │
└T9335  ┘

┌T2305┐
│T2314│
│T2319│  [As] vaddr [CYL] [nnnn]
┤T3330├
│T3340│
│T3350│
│T3375│
└T3380┘
```

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

CYL *nnnn* 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, 3370, 9313, 9332, or 9335).

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 $\left\{ \text{[As]} \quad \begin{matrix} nnnnn\mathbf{K} \\ nn\mathbf{M} \end{matrix} \right\}$

redefines the size of the virtual storage for the virtual machine as *nnnnn*K (where K represents 1024 bytes) or *nn*M (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.

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 results in any open output files being closed and available for further processing as determined by previously entered 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 addresses are the same as those specified in the mass table create control statements and in the input to the OS/VS system generation process. This is because VM/SP does not perform device mapping for MSC messages or orders.

If desired, an installation can 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 can define 3330V addresses 120 through 12F for CPUID A associated with one virtual machine user. The same set of addresses can be defined for CPUID B. VM/SP, on the other hand, can have defined addresses 120 through 12F and 220 through 22F as separate devices in DMKRIO.

All virtual machines must address the 3330V devices as 120 through 12F because these addresses are predefined in the mass table create program and by the OS/VS system generation process. With the DEDICATE statement in USER1's directory, the installation can DEDICATE real addresses 120 through 12F as virtual addresses 120 through 12F. In a similar manner, the installation can 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 with the DEFINE command.

Usage Notes

1. When you alter the configuration or channel operating mode of your virtual machine with 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 with 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. If you define a shared and a nonshared device on the same virtual control unit, an error message is issued (DMKDEF331E). Define shared and nonshared devices on separate subchannels.
6. The figure that follows 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 3. 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 can 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. After defining a virtual CTCA or virtual 3088 (VCTCA/V3088), if the VCTC has not yet been coupled, each side is considered to have the other side in the 'disabled state'. In this case, all SIOs (except a SENSEID CCW to a V3088) receive a condition code 3 in response.
10. When defining a temporary virtual disk on any volume, you can get only space that is contiguous.
11. To find out the status of the virtual devices that have been defined, enter QUERY VIRTUAL ALL.
12. If you have initiated an SIO (SIOF) to a virtual console on a TTY terminal, do not redefine the virtual console address while there is still an outstanding I/O operation to the real device; otherwise, unpredictable results can occur.

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 entered the CP SET IMSG OFF command.

```
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 can specify is 8K.

Messages

DMKDEF003E	Invalid option - *option*
DMKDEF022E	Virtual device address missing or invalid
DMKDEF024E	{CYLS|BLKS|Form number} missing or invalid
DMKDEF025E	Storage missing or invalid
DMKDEF026E	Operand missing or invalid
DMKDEF040E	Device *vdev* does not exist
DMKDEF091E	DASD *vdev* not defined; temp space not available
DMKDEF092E	Device *vdev* not defined; *devtype vdev* already defined
DMKDEF095E	Device *vdev* not defined; insufficient free storage
DMKDEF097E	Device *vdev* not defined; device *vdev* has active I/O
DMKDEF136E	Device *vdev* not defined; channel *x* dedicated
DMKDEF153E	Device {*vdev*|range|channel *x*} exceeds maximum number of virtual devices
DMKDEF331E	*vdev* not defined - use a {shared|nonshared} VCU instead
DMKDEG025E	Storage missing or invalid
DMKDEG026E	Operand missing or invalid
DMKDEG094E	Storage exceeds allowed maximum
DMKDEI046E	{Device *rid*|*devtype rdev*} offline
DMKDEI075E	Device *volid* is not available
DMKDEI274E	*rdev* has a mounted volume; MSS task not available to demount it
DMKDEI275E	*rdev* already SYSVIRT
DMKDEI276E	*rdev* not a 3330V
DMKDEI277E	Error demounting volume *volid*
DMKDEI278E	Device *rdev* not a DASD
DMKDEI279E	DASD *rdev* not a VUA
DMKDEI280E	*rdev* already virtual

DETACH

Privilege Class: G

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

Format

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

Operands

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 the operator previously attached to your virtual machine, a message is sent to inform the operator 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 can 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 or her that the device is successfully detached
- Another B privilege class operator (not the system operator) who may have initiated the detach operation.

DETACH for class G documents only those responses that the general user would receive in the course of a virtual machine terminal session. For responses the system operator receives as a result of a general user's detach operation, see the DETACH command in *VM/SP CP System Command Reference.*

The following responses do not appear on your terminal if you have entered the CP SET IMSG OFF command.

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 class B user initiated the detach operation.

```
{type vaddr  }  DETACHED BY operator, {TAPE VOLUME (S)  UNLOADED     }
{vaddr...    }                        {TAPE VOLUME (S)  NOT UNLOADED }
{vaddr-vaddr }
```

For a tape device, the user receives one of these responses upon completion of the detach operation if the system operator or another class B user initiated the detach operation.

Note: Tape volume(s) unloaded means the tape device has been detached, rewound, and unloaded. Tape volume(s) not unloaded means the tape device is detached, but the tape is not rewound and unloaded. The tape remains positioned as it was before the DETACH was entered by the operator or another class B user.

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

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

```
CTC vaddr DETACHED
```

In the preceding 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
DMKVDD003E	Invalid option - *option*
DMKVDD006E	Invalid device type - *devtype*
DMKVDD020E	Userid missing or invalid
DMKVDD021E	Real device address missing or invalid
DMKVDD022E	Virtual device address missing or invalid
DMKVDD040E	Device *rdev* does not exist
DMKVDD045E	*userid* not logged on
DMKVDD046E	{Device *rid*|*devtype rdev*} offline
DMKVDD121E	[Printer *rid*|*devtype rdev*] not attached to *userid*

DMKVDD123E	DASD *rdev* CP-owned
DMKVDD124E	DASD *rdev* in use by *nnn* users
DMKVDD135E	*devtype vdev* not detached; channel *x* dedicated
DMKVDD140E	*devtype rdev* attached to *userid*
DMKVDD142E	*devtype rdev* not drained
DMKVDD143E	*devtype rdev* in use by system
DMKVDD273E	DETACH rejected; 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.

Format

DETach	**CHANnel** *c*

Operand

CHANnel *c*
is the real address of the channel that is to be detached.

Usage Note

If DETACH CHANNEL is invoked before the 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
DMKVCH040E	Device *rdev* does not exist
DMKVCH045E	*userid* not logged on
DMKVCH048E	Channel *x* [processor *nn*] does not exist
DMKVCH130E	Channel *x* [processor *xx*] not attached to *userid*
DMKVCH138E	Channel *x* [processor *xx*] not attached; insufficient free storage
DMKVCH153E	Device {*vdev*|range|channel *x*} exceeds maximum number of virtual devices
DMKVCH188E	Specified processor unavailable
DMKVDD020E	Userid 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.

Format

Dial	*userid* [*vaddr*]

Operands

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 use NCP lines in a 3704/3705/3725 control unit; it is supported for 3138, 3148, and 3158 display consoles.

   Note: VTAM terminals using NCP lines *can* use the DIAL command.

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. When you do not specify a *vaddr*, you will get the first address available. If you specify a *vaddr*, you can dial that address as long as that address defines a terminal device.

5. 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 enter 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 is issued and the command terminates. You should then clear the screen and reissue the command.

6. 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 the DIAL command.

7. The maximum amount of data that can be transferred from remote 3270 that you DIAL 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 re-dial
DMKDIA020E	Userid missing or invalid
DMKDIA022E	Virtual device address 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	Device *rid* mode switch not possible
DMKDIA199E	Command rejected; device *rdev* pending offline
DMKDIA206E	Cannot connect to host virtual machine
DMKDIA743A	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 user ID (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.

Format

DISConn	**[HOld]**

Operand

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. If you specify the HOLD option from a logical device, you disconnect from the logical device without losing the connection between the logical device and the system.

Usage Notes

1. The terminal remains disconnected until it is reconnected with a LOGON command. The virtual machine is logged off 15 minutes after an attempt 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 a "read" to the console, which can cause the logoff time-out situation.
2. If your terminal connection is broken because of terminal, line, or teleprocessing control unit errors, CP places the virtual machine in disconnect mode for up to 15 minutes and your virtual machine does not continue to run. If you log on within 15 minutes, your virtual machine can continue operating. Because 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 entered to spool the virtual console output, all "writes" or output messages to the virtual console are ignored. When the terminal is reconnected with the usual logon procedure, the terminal is placed in CP console function mode. To resume execution of the virtual machine, enter the BEGIN command.
4. A user does not receive a message if not logged on, disconnected, or suppressing messages (SET MSG OFF), even if the user issued CP SPOOL CONSOLE START. The sender receives a diagnostic message to this effect. A message that is not received by a user is not saved and must be sent at a later time when the user is receiving messages.

   Disconnected users get messages only when they have a valid connection to the Message System Service.

5. If you have a secondary user ID 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.

6. Local console options– APL, ATTN, CONMODE 3270, HILIGHT, LINESIZE, SCRNSAVE, SCROLL, TEXT, and TYPE 3101– for the TERMINAL command are valid for the session 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 entered, the disconnect time message is issued.

```
DISCONNECT AT hh:mm:ss zone weekday mm/dd/yy
```

is the response to the user who entered the command.

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

is the response to the primary system operator informing him or her 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.

Format

```
Display  ┌                                           ┐
         │  [ hexloc 1]     {-} [hexloc 2]           │
         │  [Khexloc 1]     {:} [END     ]           │
         │  [L hexloc 1]                             │
         │  [T hexloc 1]                             │
         │                  {.} [bytecount]          │
         │                      [END      ]          │
         │                                           │
        <  Greg 1     [ {-} [reg 2]   ]               >
         │  Yreg 1     [ {:} [END  ]   ]             │
         │  Xreg 1     [               ]             │
         │             [ {.} [regcount]]             │
         │             [     [END     ]]             │
         │                                           │
         │  Psw                                      │
         │  CAW                                      │
         │  CSW                                      │
         └                                           ┘
```

Operands

hexloc1
L*hexloc1*
T*hexloc1*
K*hexloc1*

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; EBCDIC translation for uppercase and some special characters begins 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*, can be one to six hexadecimal digits; leading zeros are optional.

$$\left\{\begin{matrix}-\\:\end{matrix}\right\}\left[\begin{matrix}\textit{hexloc2}\\ \underline{\textbf{END}}\end{matrix}\right]$$

is the last of the range of hexadecimal storage locations whose contents are to be displayed at the terminal. Either a - or a : must be specified to display the contents of more than one location by storage address. If - or : is the first character of the operand, then *hexloc1* defaults to zero. If *hexloc2* is not specified, the contents of all storage locations from *hexloc1* to the end of virtual storage are displayed. If specified, *hexloc2* must be equal to or greater than *hexloc1* and within the virtual storage size. (See "Usage Notes" for a discussion on discontiguous saved segments.) The address, *hexloc2*, can 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 "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.

G*reg1*

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.

Y*reg1*

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.

X*reg1*

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 X*reg1* is specified for a virtual machine without extended mode operations available, only control register 0 is displayed.

$$\left\{\begin{matrix}-\\:\end{matrix}\right\}\begin{matrix}\textit{reg2}\\ \underline{\textbf{END}}\end{matrix}$$

is a number representing the last register whose contents are to be displayed at the terminal. The dash (-) or colon (:) following *reg1* specifies 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 G*reg1* or X*reg1* is specified, *reg2* may be a decimal number from 0 to 15 or a hexadecimal number from 0 to F. If Y*reg1* 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 through 15 or a hexadecimal number from 1 through 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 through 15 or a hexadecimal number from 1 through 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 entering 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 displays only 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 can 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 can then enter 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 (with 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 nontranslatable characters. Hexadecimal values for lowercase letters are not translatable. Multiple lines are used (if required) for a range of locations. If translation to EBCDIC is requested (T*hexloc*), 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-numbered 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 - *operand*
DMKCDB009E	Invalid range - *range*
DMKCDB010E	Invalid register - *register*
DMKCDB026E	Operand missing or invalid
DMKCDB160E	Hexloc *hexloc* exceeds storage
DMKCDB923I	Failing storage frame at location *rstor*

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 (see Usage Note 8)
- 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.

Format

```
DUmp  [Lhexloc 1]   ⎧ {-} [hexloc 2] ⎫     [*dumpid]
      [Thexloc 1]   ⎪ {:} [END     ] ⎪
      [ hexloc 1]   ⎨                ⎬
                    ⎪ {.} [bytecount]⎪
                    ⎩     [END      ]⎭
```

Operands

L*hexloc1*
T*hexloc1*
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*, can 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.

$\begin{Bmatrix}-\\ :\end{Bmatrix}\begin{bmatrix}hexloc2\\ \underline{\text{END}}\end{bmatrix}$

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. You can specify up to 100 characters, with or without blanks, 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 press 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, you must enter a CLOSE command for the spooled virtual printer.
6. 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 end of virtual storage. Blanks 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 dumps only 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.

8. You can enter DUMP to print the contents of control registers: When ECMODE is ON, the contents of registers 0 through 16 are printed; when ECMODE is OFF, only the contents of register 0 are printed. You can SET ECMODE with an entry in your VM/SP directory or by entering the SET ECMODE command.

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 successful completion of the dump function.

Message

DMKCDM003E	Invalid option - *option*
DMKCDM004E	Invalid hexloc - *operand*
DMKCDM009E	Invalid range - *range*
DMKCDM026E	Operand missing or invalid
DMKCDM033E	Hexloc missing or invalid
DMKCDM060E	Dump failed; virtual printer unavailable
DMKCDM061E	Dump failed; virtual printer error
DMKCDM160E	Hexloc *hexloc* exceeds storage
DMKCDN003E	Invalid option - *option*
DMKCDN004E	Invalid hexloc - *operand*
DMKCDN009E	Invalid range - *range*
DMKCDN026E	Operand missing or invalid
DMKCDN033E	Hexloc missing or invalid
DMKCDN060E	Dump failed; virtual printer unavailable
DMKCDN061E	Dump failed; virtual printer error
DMKCDN160E	Hexloc *hexloc* exceeds storage
DMKCDN923I	Failing storage frame at location *raddr*

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.

Format

ECho	[*nn* 1]

Operands

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

Usage Notes

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

Format

EXTernal	[*code* 40]

Operand

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 that cause an external interrupt. Control is given to the virtual machine immediately.

Responses

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

Message

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.

Format

```
INDicate     [LOAD]
             [USER]
```

Operands

LOAD
: provides an indication of the operating load of VM/SP by displaying values for 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 your virtual machine is using in the current terminal session.

Usage Notes

1. Enter INDICATE LOAD to display system load conditions.
2. Enter INDICATE USER to display the total amount of certain resources your virtual machine is using 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 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 scheduler maintains the values.

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% of the user's working set size must fit in the available page frames, the value of STORAGE can be more than 100%.

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 Administration* book.

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 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

DMKTHI003E Invalid option - *option*
DMKTHI045E *userid* not logged on

IPL

Privilege Class: G

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

Format

```
Ipl   ┌ vaddr  ┌cylno  ┐ ┌CLear  ┐                    ┐
      │        └nnnnnn ┘ └NOCLear┘  [STOP]  [ATTN]    │ [PARM p1[ p2 ...]]
      │                                               │
      └ systemname                                    ┘
```

Operands

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 with the SAVESYS command. It is loaded into virtual storage and given control. For information about saved systems, see the *VM/SP Administration* book.

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

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

PARM *p1 [p2...]*
passes up to 64 bytes of data (excluding all leading blank characters after the keyword, PARM, but including all other embedded and trailing blanks) to your virtual machine's general registers (4 bytes per register).

Note: For an IPL of a virtual device, the movement begins with the high order byte of general register zero. One byte of binary zeros is added at the end of the IPL data, unless 64 bytes of data already exist. For a named system, the place and length of movement is determined by the PARMRGS option specified in the NAMESYS entry for the named system.

For further details on the PARMRGS option see the *VM/SP Planning Guide and Reference.*

For parameters that CMS recognizes, see "Optional CMS Initialization Parameters."

Optional CMS Initialization Parameters

The following CMS parameters are passed to the System Profile EXEC (SYSPROF EXEC). Any unrecognized parameters are ignored by CMS initialization, but are still passed to the SYSPROF EXEC.

CMS users can use the PARM operand to specify any of the following CMS parameters:

AUTOCR
is the automatic carriage return parameter that simulates the pressing of the ENTER key as input to the virtual machine at the initial VM READ. Your PROFILE EXEC is automatically executed if it exists on your filemode A.

BATCH
indicates that the CMS IPL is being performed in a batch instead of an interactive virtual machine.

Use of this parameter does not affect execution of the SYSPROF EXEC. Execution of the SYSPROF EXEC is suppressed only if the NOSPROF parameter was specified.

At the beginning of each job, the batch facility work disk is accessed and then immediately erased, preventing the current user job from accessing files that might remain from the previous job. Because of this, execution of the PROFILE EXEC is disabled for the CMS Batch Facility machine. However, if the BATPROF EXEC exists on an accessed system disk, it is invoked instead of the PROFILE EXEC.

FILEPOOL *filepoolid*
specifies the default file pool that is established as the user's filemode A. If FILEPOOL is not specified, 191 is established as the filemode A.

The following parameters are passed to the SYSPROF EXEC, but the SYSPROF EXEC does not process them. For additional information about the SYSPROF EXEC, see the *VM/SP Administration* book.

NOSPROF
indicates that the SYSPROF EXEC is bypassed. You must specify NOSPROF if you intend to enter the CMSBATCH command when the initial VM READ is issued. When the IPL is of a non-DASD device or the SYSPROF EXEC does not exist, you do not have to specify NOSPROF to enter the CMSBATCH command.

INSTSEG YES
links the default CMS installation saved segment for this CMS session. The CMS installation saved segment is an optional shared segment that contains execs and editor macros that your installation provides.

INSTSEG NO
indicates that you do not want to use the CMS installation saved segment during this CMS session.

INSTSEG *name*
links the named CMS installation saved segment for this CMS session. The segment may be a physical or a logical saved segment.

SAVESYS *sysname*
is used in the process of creating named systems. The SAVESYS parameter is available only to class E users. It saves a virtual machine storage space with registers and PSW as they currently exist. The *sysname* must be a predefined name representing a definition of installation requirements of the named system. The definition indicates the number of pages to be saved, the DASD volume on which the system is to be saved, and the shared segments, if any.

After the system is saved, initialization continues, all CMS parameters entered on the IPL command are ignored, and SYSPROF EXEC is executed. The parameters entered on the original IPL command are not passed to SYSPROF EXEC.

No other parameters may be specified with this parameter. If any other parameters are specified, all parameters are ignored and a CONFLICT parameter is passed to SYSPROF EXEC, as well as the parameters entered on the IPL command.

Usage Notes

1. If you want to use the NOSPROF, INSTSEG, or SAVESYS parameters of the IPL command, you must have:

   ```
   PARMRGS=(0,15)
   ```

   in the NAMESYS macro for the CMS named system.

2. IPL simulates the IPL sequence on real hardware. The specified virtual address is accessed and the required I/O operations are performed to retrieve the IPL data.

3. Optionally, the IPL procedure can be stopped just before loading the virtual PSW except when loading a named system. Also, parameters can be passed to the virtual machine's general registers. When the simulated load function is complete, CP initiates execution of the virtual machine by loading the IPL PSW that was stored during the simulation process.

4. Be careful when passing parameters to a named system (*systemname*). Named systems expect certain registers to be initialized when they are given control. Indiscriminate use of the PARM operand could overlay a previously initialized register causing unpredictable results.

5. For IPL by device, the IPL simulator program occupies one page (4096 bytes) of storage in your virtual machine. Therefore, if, during the processing of the IPL CCWs, the system being loaded utilizes data remaining in storage by the previous system, be careful about where that data is located. Also, be careful where the system is being loaded to avoid conflict with the location of the IPL Simulator Page. The starting address of the overlaid virtual storage is:

Virtual Storage Size		**Location of IPL Simulator Page**
256K and above	=	20,000
Less than 256K	=	1/2 the storage size

   For example:
   Storage size=200K then the IPL simulator is loaded at 100K.

6. To load a VMSAVE system, a number of conditions must be met:

   - The user must be specified in module DMKSNT as either the owner (USERID=) or the receiver (RCVRID=).
   - The area must contain a valid saved system.
   - The system must have been saved under the same *systemname* in the IPL command.
   - The area cannot be the target area for other users currently enabled for VMSAVE.

   When these conditions are not met, an appropriate message is issued, and the IPL is terminated.

7. Any logical lines following the IPL command are ignored because the execution of the IPL command transfers control from the CP environment to the virtual machine environment.

8. To indicate that the CMS initial program load is being performed in a batch rather than an interactive virtual machine, you must either specify the BATCH parameter on the IPL command, or specify the NOSPROF parameter and then enter the CMSBATCH command at the VM READ. Then, a BATPROF EXEC is executed in the batch machine rather than a PROFILE EXEC. 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. See the *VM/SP CMS Command Reference* for information on the CMSBATCH facility.

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

   Note: For a disconnected CMS service virtual machine, this mechanism can be very useful. The name of a CMS restart routine can be specified in the CMS nucleus. In case 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 are 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 you IPL the loader from your virtual machine to create a CP nucleus, CONMODE should be set to 3215 mode. Otherwise, console messages generated by the loading process are not displayed at the terminal.

13. The SYSPROF EXEC contains some of the CMS initialization function. When you IPL CMS, it is executed by default before any filemodes are accessed. The user PROFILE EXEC is invoked after the SYSPROF EXEC. You can bypass the SYSPROF EXEC by specifying the NOSPROF parameter on the IPL command. For additional information, see the *VM/SP Application Development Guide for CMS* book and the *VM/SP Installation Guide.*

14. If you have defined disk addresses 190, 192, and 19E in the VM/SP directory, or if they are defined before you IPL CMS, these disks are accessed as the S-, D-, and Y-disks, respectively. After you IPL CMS, you must issue explicit ACCESS commands to access other disks. Ordinarily, you have access only to files with a file mode number of 2 on the system disk.

    If you have defined disk address 191 and the FILEPOOL parameter is omitted, the 191 disk is accessed as filemode A. However, if the FILEPOOL parameter is specified, your top directory in the identified file pool is accessed as filemode A, even if you also have a 191 minidisk defined. If the access of the top directory fails (perhaps because the file pool is unavailable), the 191 minidisk is accessed. If there is no 191 minidisk, nothing is accessed as filemode A.

When ACCESS is the first command issued after an IPL of the CMS system, filemode A is not automatically defined. Another ACCESS command must be issued to define filemode A.

For more information on the CMS ACCESS command, see the *VM/SP CMS Command Reference*.

15. When you IPL a virtual machine with a virtual CTCA or virtual 3088 (VCTCA/V3088) a system reset is simulated for those devices. When you IPL a virtual machine with a dedicated CTCA or D/T3088, a HDV (halt device) is issued to the hardware. This causes an interface disconnect.

Responses

During IPL, CP responses are not displayed if you have entered SET IMSG OFF.

After a successful IPL, any responses you receive are those from the operating system that was loaded and initialized.

Messages

DMKCFF083E	VMSAVE area does not contain valid saved system
DMKCFG002E	Invalid operand - *operand*
DMKCFG003E	Invalid option - *option*
DMKCFG013E	Conflicting option - *option*
DMKCFG026E	Operand missing or invalid
DMKCFG040E	Device *vdev* does not exist
DMKCFG044E	System *sysname* does not exist
DMKCFG170E	System *sysname* exceeds storage
DMKCFG171E	System *sysname* volid *volid* not mounted
DMKCFG172E	System *sysname* DASD *vdev* incompatible SYSRES
DMKCFG173E	System *sysname* requires DASD *vdev*
DMKCFG174E	Paging error - I/O error: IPL failed
DMKCFG177E	Parameter exceeds 64 characters
DMKCFG203E	Virtual = Real area requires IPL by device address
DMKCFG437E	System paging space not available, IPL terminated
DMKCFQ546I	Interruption pending; device *rdev*, CSW *csw*, userid *userid*
DMKCLK970W	TOD CLOCK SYNC CHECK RECEIVED
DMKSAV972E	NUCLEUS AREA IS TOO SMALL; IT IS NOT UPDATED. THIS NUCLEUS WOULD REQUIRE *storage*.
DMKVMI022E	Virtual device address 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.

Format

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.

Operands

[TO] *userid*
is the name of the user whose VM/SP directory is to be searched for device *vaddr1*. An asterisk (*) specifies 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 a write link to the disk exists, 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 ensures 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. Write access is established if no other user has a link to the disk; 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 this case, a read link is given because 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).

For example, in CMS, when a file is updated, it is moved on the disk and the directory for the disk is updated. The user with a read-only link will be using an old copy of the file until the disk is reaccessed. As a precaution, the user with a read-only link should reaccess the disk every time a user with a write link updates the disk.

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

Be aware that 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 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), because 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 is forced read-only (R/O). You can obtain write access by entering 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 be requested only by 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 4. Illustration of the Type of Link Established Under Every Condition

6. When entering 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* causes a conflict in the user's virtual machine; if so, the command is not executed and an error message is sent. See 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.

7. If the LINK command is issued from a disconnected virtual machine, the link is not performed until the directory block of the user being linked to is free.

Responses

```
Enter read password:
XXXXXXXX
```

Type the read password over the mask to request read-only access to the desired disk. The mask appears only 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 appears only 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 appears only 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 that 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 user ID 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 preceding 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

DMKLNM020E	Userid missing or invalid
DMKLNM022E	Virtual device address missing or invalid
DMKLNM052E	Error in CP directory
DMKLNM053E	*userid* not in CP directory
DMKLNM101W	DASD *vdev* forced R/O; R/O by {*nnn* users|*userid*}
DMKLNM102W	DASD *vdev* forced R/O; R/W by {*nnn* users|*userid*}
DMKLNM103W	DASD *vdev* forced R/O; R/W by {*nnn* users|*userid*}
DMKLNM104E	*userid vdev* not linked; R/O by {*nnn* users|*userid*}
DMKLNM105E	*userid vdev* not linked; R/W by {*nnn* users|*userid*}
DMKLNM107E	*userid vdev* not linked; not in CP directory
DMKLNM108E	*userid vdev* not linked; volid *volid* not mounted
DMKLNM109E	*userid vdev* not linked; invalid link device
DMKLNM110E	*userid vdev* not linked; *devtype vdev* already defined
DMKLNM111E	*userid vdev* not linked; no read password
DMKLNM112E	*userid vdev* not linked; no write password
DMKLNM113E	*userid vdev* not linked; no multiple password
DMKLNM114E	*userid vdev* not linked; mode or password incorrect
DMKLNM115E	LINK invalid; excessive incorrect passwords
DMKLNM116E	*userid vdev* not linked; CP directory in use
DMKLNM117E	Userid *userid* not linked; volid *volid* conflict
DMKLNM118E	*userid vdev* not linked; command format not valid
DMKLNM119E	*userid vdev* not linked; insufficient free storage
DMKLNM137E	DASD *vdev* not linked; channel *x* dedicated
DMKLNM153E	Device {*vdev*|range|channel *x*} exceeds maximum number of virtual devices
DMKLNM199E	Command rejected; device *rdev* pending offline
DMKLNM281E	MSS volume *volid* not linked; demount in progress, retry
DMKLNM331E	*vdev* not linked - use a {shared|nonshared} VCU instead
DMKSSS071E	Device not available - MSS cannot be accessed
DMKSSS078I	MSS mount proceeding for volume *volid*
DMKSST070E	Device not available - MSS devices allocated
DMKSSV073E	Device not available - MSS error

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.

Format

LOADVFCB	*vaddr* **Fcb** *name* [**Index** [*nn*]]

Operands

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 Administration* book 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 specify a virtual FCB image for 1403 printers so that programs that use printer overflow sensing can 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 can provide you with additional FCB images.

3. The virtual machine operating system can 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 usual 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 then prints 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. When sent to the real hardware, the Load FCB CCW resets the FCB line position to 1. Therefore, the application must 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. An attempt to send the extended FCB image to a printer that does not support the extended FCB format generates an error message.

Responses

None.

Messages

DMKCSB006E	Invalid device type - *devtype*
DMKCSB022E	Virtual device address missing or invalid
DMKCSB026E	Operand missing or invalid
DMKCSB031E	Buffer missing or invalid [*;modifier*]
DMKCSB036E	Index missing or invalid
DMKCSB040E	Device *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.

Format

LOGoff **LOGout**	[**HOld**]

Operand

HOld
retains the connection for a switched communication line to enable you to log on without redialing the VM/SP system. If you specify the HOLD option from a logical device, you log off from the logical device without losing the connection between the logical device and the 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 off terminal power. Turning 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. This occurs if no attempt is made to turn terminal power on and communicate with the still logged-on virtual machine during the 15 minutes.
   - *Local 3270 Display Terminal* — Logoff takes place 15 minutes after VM/SP discovers that the terminal has been turned off 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 can 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:

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 previously listed, control automatically returns to CP and the "VM/370 ONLINE" message is printed.

$$\left\{\begin{array}{l}\text{GRAF raddr}\\ \text{LINE raddr}\\ \text{DEV rid}\end{array}\right\}\text{ LOGOFF AS userid USERS = nnn}$$

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

$$\left\{\begin{array}{l}\text{GRAF raddr}\\ \text{LINE raddr}\\ \text{DEV rid}\end{array}\right\}\text{ LOGOFF AS userid USERS = nnn FORCED}$$

is the response to the primary system operator if the logoff is forced by a line timeout or by turning off the terminal. `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

DMKCFQ546I	Interruption pending; device *rdev*, CSW *csw*, userid *userid*
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.

Format

Logon **Login**	*userid* [*password*] [**Noipl**]

Operands

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 turned 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 sure 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 entered 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 the DISCONN command. For example, after a reconnect, CP resets the line delete character specified in the directory to the system default value. Enter 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 (by the user directory's OPTION statement) that the AFFINITY setting of the 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 you exceed system-defined values by attempting to LOGON with trial passwords, a message is issued to an installation user ID and the VM logo is displayed in response to any further LOGON attempt.

7. CP automatically masks your password on typewriter terminals to ensure 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 your password should you forget to include the MASK operand on the LOGON command.

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 enter 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*|*LDEV*|*device*} *rdev*|line disconnected}
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. A `luname` response specifies the logical unit name of a user who logs on to VM/SP through a SNA network.

Messages

DMKCFS080E	No preallocated VMSAVE area
DMKCFS081E	VMSAVE area already in use, must use SET command
DMKCFS082E	More than one VMSAVE area allocated, must specify area name
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*|*LDEV*|*device*} *rdev*|line disconnected}
DMKLOG067E	Command format not valid
DMKLOG131E	Insufficient free storage available for LOGON request

DMKLOG206E	Cannot connect to host virtual machine
DMKLOG361E	LOGOFF/FORCE pending for user *userid*
DMKLOH188E	Specified processor unavailable
DMKLOH189E	AFFINITY not set; system in uniprocessor mode
DMKLOH250E	370E is not available [on processor *xx*]
DMKLOH251E	370E feature is not active
DMKLOH365E	Requested language *langid1* is unavailable. Language *langid2* set.
DMKLOJ090E	Device *vdev* not defined; device {*rdev*|*volid*} not available
DMKLOJ091E	DASD *vdev* not defined; temp space not available
DMKLOJ092E	Device *vdev* not defined; *devtype vdev* already defined
DMKLOJ093E	Device *vdev* not defined; error in CP directory
DMKLOJ095E	Device *vdev* not defined; insufficient free storage
DMKLOJ331E	*vdev* not created - use a {shared|nonshared} VCU instead
DMKLOJ769W	Incomplete machine - tell your system administrator
DMKLOM196E	Secondary user not in CP directory
DMKLOM331E	*vdev* not attached - use a {shared | nonshared} VCU instead
DMKSSS071E	Device not available - MSS cannot be accessed
DMKSSS072E	Device not available - *rdev* not proper type
DMKSSS078I	MSS mount proceeding for volume *volid*
DMKSST070E	Device not available - MSS devices allocated
DMKSSV073E	Device not available - MSS error

MESSAGE

Privilege Class: Any

Use the MESSAGE command to transmit message text to a specified *userid* or to the primary system operator.

Format

```
Message    { userid   }     msgtext
Msg        { *        }
           { OPerator }
```

Operands

userid
identifies 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.

msgtext
is the text of the message to be transmitted. The length of the message is limited to the number of characters remaining on the input line after you enter the command and the appropriate operand.

Usage Notes

1. A user does not receive a message if not logged on, disconnected, or running with messages suppressed (SET MSG OFF), even if the user entered "CP SPOOL CONSOLE START." The sender receives a diagnostic message to this effect. A message that is not received by a user is not saved and must be sent at a later time when the user is receiving messages.

   Disconnected users get messages only when they have a valid connection to the Message System Service.

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 user ID of an abandoned terminal, or it can be used in a program intended to be executed under several user IDs.

6. If the system operator logs off and another class A user logs on, messages to the operator are 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' is 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 recognized as the system operator.

Responses

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

is the message that the user receives 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|message too long|SMSG off| not authorized|WNG off|IUCV error *nnnn*|VMCF error *nnnn*|quiesced}

NOTREADY

Privilege Class: G

Use the NOTREADY command to change a virtual device to not ready status. You cannot specify a Start I/O (SIO) instruction for a virtual device in NOTREADY status.

Format

NOTReady	*vaddr*

Operand

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.

Responses

None.

Messages

DMKCPB006E	Invalid device type - *devtype*
DMKCPB022E	Virtual device address missing or invalid
DMKCPB040E	Device *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.

Format

```
ORDer    ⎧ Reader  ⎫ ⎧ CLass c1  CLass c2 ...          ⎫ [1]
         ⎨ Printer ⎬ ⎨ FORM form1  FORM form2 ...      ⎬
         ⎩ PUnch   ⎭ ⎨ DEST dest1  DEST dest2 ...      ⎬
                     ⎩ spoolid1  spoolid2 ...          ⎭
```

[1] You can enter a combination of CLASS, FORM, DEST and *spoolid* specifications.

Operands

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 through Z and from 0 through 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*, *form2*, ... specifications are one- to eight-character form names defined during system generation.

DEST *dest1* **DEST** *dest2...*
are the destination names for the files you want rearranged, and the sequence in which you want CP to process them. The *dest1*, *dest2*, ... specifications are one- to eight-character alphameric destination names your installation assigns to printers and punches during system generation.

spoolid1 spoolid2...
are the spool IDs of the files to be rearranged and the sequence in which they are to be processed.

Usage Notes

1. You can order only your own files.
2. Use the QUERY command to determine spool file attributes such as form, class, and spool ID for files you want to rearrange.
3. You can use a combination of CLASS, FORM, DEST, and *spoolid* specifications to rearrange files. For example,

   ```
   order printer class a 1963 class c form catalog4 dest bldgl
   ```

   specifies that CP queues printer files in the following order: all class A files, the file with spool ID 1963, all class C files, all files with form name CATALOG4, all files with a destination of BLDG1, followed by any other printer spool files not specified in the ORDER command line.

   As files are selected, they are removed from the list of files to be processed by the ORDER command. If the spool ID 1963 file in the preceding example were a class A file, it would be selected with other class A files and removed from the list of files to be processed. When the spool ID 1963 criteria was processed, the message DMKCSV042E SPOOLID *nnnn* DOES NOT EXIST would be displayed.
4. When duplicate spoolids are present, the ORDER command resequences only the first duplicate spool file found.
5. The order of your spool files can 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*
DMKCSV013E	Conflicting option - *option*
DMKCSV024E	{FORM number|DEST} 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 *nnnn* does not exist

PER

Privilege Class: G and all IBM-defined 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 you can monitor with the PER command are

- The fetching and execution of an instruction
- The execution of a successful branch instruction
- The execution of an instruction that alters a specific general-purpose register
- The execution of an instruction in the virtual machine that alters storage.

For examples on how to use the PER command for debugging, see "Testing and Debugging of Programs" on page 24.

Format

```
PER   EVENT TYPES:
     ┌                                                                ┌         ┐┐
     │Instruct [[ DATA ]  hex-data ]                                  │         ││
     │BRanch   [[ INTO ]  into-addr-range ]                           │ options ││
     │                                                                │         ││
     │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   instruction-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    ┘
```

Operands

EVENT-TYPES

The PER command supports five different EVENT keywords. More than one EVENT keyword can be placed on a single line, and they can be in any order. If 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. The *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 is produced only when an instruction alters storage in the indicated range.

If *addr* and *hex-data* are specified, then *addr* must be a single address. It indicates the address where the data comparison begins. The *hex-data* that follows must consist of complete bytes. The length of the *hex-data* determines the number of bytes to be tested. In this case, trace output is 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. This value need *not* 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 clc2c3
```

traces instructions that alter storage in the range 200 to 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 to test. The *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 is 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 results.

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 2-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 through 15 or a hexadecimal integer from 0 through F representing the first, or only, register to be monitored. If *reg2* is specified, then all the registers from *reg1* to *reg2* are monitored. The *reg2* must be greater than or equal to *reg1*. If *regcount* is specified, then it is a decimal number from 0 through 15 or a hexadecimal integer from 0 through 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 are monitored. G-*reg2* and G:*reg2* specify the registers from 0 to *reg2* and G.*regcount* specifies the registers from 0 to *regcount* minus 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. The *hexword* consists of one to eight hexadecimal characters, padded on the left with zeros as necessary, to complete a fullword. Trace information is 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 to 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. All except CMD (see CMD option) can be placed anywhere on the command line. These options are:

Range
FRom
PAss
CMd
Printer
TErminal
BOth
RUN
NORun
STEp
GUESTR
GUESTV
DATOFF
DAT

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* applies to *all* event types on the same command line. If multiple ranges are specified in this case, then each range applies to all the event types on that command line. A maximum of eight ranges can be specified on a single command line.

Examples are

```
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 and replaces all the *instruction-addr-ranges* currently in effect. See "Changing Options on Trace Elements" on page 139.

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. Note that address stops set using the PER command remain in effect until you end the trace element.

To set up multiple address stops with PER, use 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 entered 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 is always 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 invalid combinations 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 $\begin{bmatrix}\underline{1}\\ n\end{bmatrix}$

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. The *n* must be an integer between 1 and 32,767. If *n* is not specified, then 1 is the default value for *n*. Specify either the RUN or the NORUN option to turn off STEP. Do not specify STEP more than once on a command line.

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

"PRINTER and NORUN" or "PRINTER and STEP" are invalid combinations 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 and PRINTER options are turned on.

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 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 are 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 are included in the QUERY output.

SUBCOMMANDS

The PER command supports five subcommand keywords. Only one subcommand can be specified on a command line. A subcommand 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 subcommand is valid only when a current traceset exists.

When the PER count reaches 16 million, this count is displayed, and the count continues 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 you END it or log off. A *traceset-name* must be one to eight 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 "Merging Trace Elements" on page 139.)

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 entered the PER command

```
per save set1
```

the saved traceset (SET1) now contains:

```
query per set1
```

```
INSTRUCT RANGE 020000-030000 TERMINAL NORUN
BRANCH TERMINAL RUN
```

GET *traceset-name* **[APpend]**

causes the current traceset to be replaced by a copy of the saved traceset. The saved traceset is unchanged.

If APPEND is specified, then the named traceset is merged with the current traceset. (See "Merging Trace Elements" on page 139.)

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.

The *traceset-name* eliminates all trace elements in the saved traceset.

The *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.

The *event-type* eliminates any trace elements in the current traceset that match the event-type specified. The *event-type* can be any of the possible EVENT keywords or their abbreviations (for example, BR, I, G, ST, M).

Note: The elements in the current traceset are renumbered following the execution of the PER END *element-number* or PER END *event-type* subcommand.

Merging Trace Elements

Trace elements within a traceset are merged whenever the following conditions are satisfied:

- The event types are the same.
- The options PASS, STEP, RUN, NORUN, PRINTER, TERMINAL, CMD, GUESTR and GUESTV match.
- The DATA (if any) for both trace elements are equal.
- The INTO and FROM ranges, or the registers and FROM ranges are such that the trace elements can be merged to form a trace element that is equivalent to the original two.

For example, if the current traceset contains

```
1 STORE INTO 023048 DATA FF88 RANGE 020000-021000
                                              TERMINAL NORUN
2 BRANCH INTO 021000-021008 RANGE 020000-021000 TERMINAL NORUN
```

and you enter

```
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 you enter

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

then the range 24000 to 25000 replaces the from ranges 20000 to 21000 and 20000 to 24000 (however, the storage address range 23034 to 23100 is not changed). The PASS 3 option is applied, the PRINTER option replaces the TERMINAL option, and the RUN option replaces the NORUN option. The resulting current traceset is

```
1 G2 G4 RANGE 024000-025000 PRINTER RUN PASS 3
2 INSTRUCT 41 RANGE 024000-025000 PRINTER RUN PASS 3
3 STORE INTO 023034-023100 RANGE 024000-025000 PRINTER
  RUN PASS 3
```

Changing Specific Trace Elements

The options in effect for a specific trace element in a traceset are replaced by the options specified on a new trace element whenever the following conditions are satisfied:

- The event types are the same.
- The DATA (if any) for both trace elements are equal.
- The FROM ranges, INTO ranges (if any), and registers specified (if any) are equal.

For example, if the current traceset contains

```
1 G2 G4 RANGE 020000-024000 TERMINAL NORUN
2 INSTRUCT 41 RANGE 020000-021000 PRINTER TERMINAL RUN
3 STORE INTO 023034-023100 RANGE 020000-024000 PRINTER RUN
```

and you enter

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

then the resulting current traceset is

```
1 G2 G4 RANGE 020000-024000 TERMINAL NORUN
2 INSTRUCT 41 RANGE 020000-021000 PRINTER TERMINAL RUN
3 STORE INTO 023034-023100 RANGE 020000-024000 TERMINAL
                                                 NORUN PASS 4
```

Note that, although not explicitly specified, the default options TERMINAL and NORUN have also been applied to the third trace element.

For the PER GET subcommand with the APPEND option, whenever a trace element in the saved traceset matches a trace element in the current traceset as described previously, 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 previously, 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 enter

```
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 entering 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 is 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 (for example, SVC, PROG, I/O, External) is not traced. These interrupts can be traced using the TRACE command.
   - The storing of the CSW for I/O operations is not traced.
   - The updating by CP of the interval timer is not traced.
   - The alteration of storage caused by the following CP simulated instructions 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 is 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 to 20 is the same as the single address 20.

11. For MASK, the storage value is saved when the original command is entered 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. Certain tracing can take longer than others depending on the events and ranges chosen for the trace. This is related to the way that the hardware recognizes a successful PER event. The PER command allows greater selectivity than the hardware for the PER events that are displayed. Therefore, some PER interrupts generated by the hardware are not displayed due to the selectivity of the PER command. 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 goes to the printer with the lower virtual address. However, there is no way to prevent output from PER and TRACE from being mixed.

14. To use the line-end character for multiple commands, LINEDIT must be ON before issuing the PER command.

    When multiple commands have been properly entered, 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 two separate command buffers. If a branch occurs, then both the instruction and branch trace elements are successful. 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 successful at the same time, then the commands are executed in the order specified by the QUERY PER command.

16. If you enter 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 enter

    ```
    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 is counted only once, regardless of how many trace elements are successful. For example,

    ```
    per i br r 20000.1000
    per count
    ```

    The execution of a successful branch instruction in the range 20000 to 20FFF adds 1 to the count even though both the INSTRUCT and BRANCH trace elements are 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 you enter

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

    then the resulting current traceset would be

    ```
    1 INSTRUCT RANGE 020000-024000 TERMINAL NORUN
    ```

    However if the current traceset contains

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

    and the command

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

is entered, 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 that appears real to your virtual machine. Third-level storage is storage that 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 fewer than 255 elements.

23. The SLEEP command should not be used with the PER CMD option. If the SLEEP command is entered from the CMD option while PER is active, then the SLEEP command is ignored. If the SLEEP command is entered 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 is 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 in the preceding)
```

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.

The 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
DMKPEI312E	More than {8 ranges|1 PASS|1 STEP} specified on a command line
DMKPEL013E	Conflicting option - {*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	{*userid* {*vdev*|*segname*} |The current traceset|The traceset *traceset* |A saved 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	{*userid* {*vdev*|*segname*} |The current traceset|The traceset *traceset* |A saved traceset} does not exist
DMKPEN180E	PER trace not in effect
DMKPEN311E	Maximum number of trace elements exceeded
DMKPER174E	Paging error - {I/O|address exception}: {command terminated, IPL failed}
DMKPER182I	PER trace is active
DMKPER313E	PER failure - code *n*
DMKPER314W	PER failure - code *n*, userid = *userid*
DMKPET003E	Invalid option - *option*
DMKPET061E	Virtual printer error
DMKPET180E	PER trace not in effect
DMKPET313E	PER failure - code *n*
DMKPET314W	PER failure - code *n*, userid = *userid*

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.

Format

```
PURge  {                                                  }
       {  Reader    [ CLass c1  CLass c2 ...         ]1   }
       {  Printer   [ FORM form1  FORM form2 ...     ]    }
       {  PUnch     [ DEST dest1  DEST dest2 ...     ]    }
       {  ALL       [ spoolid1    spoolid2 ...       ]    }
       {            [ ALL                            ]    }
```

[1]You can enter a combination of CLASS, FORM, DEST and *spoolid* specifications.

Operands

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 through Z and 0 through 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*, *form2*, ... options are one- to eight-character form numbers.

DEST *dest1* **DEST** *dest2...*
purges the files with the specified destination values. The *dest1*, *dest2*, ... options are one- to eight-character alphameric names your installation assigns to printers and punches during system generation.

spoolid1 spoolid2...
are the spool IDs 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, DEST, and *spoolid* specifications. For example,

   ```
   purge printer class a 1932 class d 619 form invoice dest bldgl
   ```

   specifies that CP should purge all class A and class D printer files, printer files with spool IDs of 1932 and 619, all printer files with form name INVOICE, and all printer files with a destination of BLDG1.
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*
DMKCSV013E	Conflicting option - *option*
DMKCSV024E	{FORM number|DEST} 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	Printer *rdev*, FCB-forms mismatch; fileid *nnnn* {held|purged}
DMKTCS241E	Printer *rdev*, invalid load module *mmmm* specified, file *nnnn* {held|purged}

QUERY (ALL)

Privilege Classes: All IBM-defined Classes of Users Except Class Any

Use the QUERY command for all classes of users except those in the Any category to:

- Display the identifier of the language that is set to issue CP messages
- Display the log messages
- List all the users that are logged on
- Display PER command tracesets and trace elements
- Display the number of users that are logged on or dialed to VM/SP.

Format

```
Query  { CPLANG                         }
       { LOGmsg                         }
       { Names                          }
       {        [ ALL            ]      }
       { PER    [ Names          ]      }
       {        [ traceset-name  ]      }
       {        [ CUrrent        ]      }
       { Users    [userid]              }
```

QUERY Operands and Responses

QUERY CPLANG

CPLANG
: displays the identifier of the language in your virtual machine which is set to issue CP messages.

Response:

```
langid
```

where:

`langid` is the identifier of the language that is set in your virtual machine to issue CP messages.

For example, your virtual machine is running and issuing CP messages in American English, which has a language ID of AMENG. When you enter query cplang, the following response is displayed:

```
AMENG
```

QUERY LOGMSG

LOGmsg
displays the log messages of the day.

Response:

```
*  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

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.

Response:

```
userid - ┌DSC  ┐ ,...
   .     ┤raddr├
   .     └resid┘
   .
userid - ┌DSC  ┐ ,...
         ┤raddr├
         └resid┘

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

`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 the user is connected is displayed; if the user is not connected to the system, DSC is displayed. The userid - luname response displays the user IDs 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 user IDs 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 PER

PER

displays PER command tracesets and trace elements.

QUERY PER CURRENT

PER <u>CUrrent</u>

displays the current traceset elements and any active PER count.

Response:

```
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

PER Names

displays the names of all saved tracesets.

Response:

```
saved-traceset1 saved-traceset2...saved-tracesetn
```

where:

saved-tracesetn

is the name of the nth saved traceset.

QUERY PER traceset-name

PER *traceset-name*

displays the traceset elements for the specified traceset.

Response:

```
   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

PER ALL

displays all the current traceset elements, the value of an active PER count, and the name and contents of all the saved tracesets.

Response:

```
  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 USERS or QUERY USERS userid

Users
displays the number of logged on users and the number of users logically connected to other virtual machines.

Response:

```
nnnn USERS, mmmmm DIALED, xxxx NET
```

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

where:

nnnn
: is the total number of logged-on users.

mmmmm
: is the total number of users attached by DIAL to virtual machines.

xxxx
: 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.

Users *userid*
displays the user identification and the terminal device address of the specified user if the user 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).

Response:

```
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/Pass-Through program product, Lnnn displays the logical address of that user.

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 (G)

Privilege Class: G

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 user ID and system identifier.

There are other operands you can use with the QUERY command if you have the privilege class required to use them; information about class A, B, C, D, E, and F QUERY formats is contained in *VM/SP CP System Command Reference*. 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.

Format (Class G)

```
Query   { CPLEVEL
        
          CPUid
                                                         [ HOld     ]
          Files  [CLass c ] [FORM form ] [DEST dest] [*] [ NOHold   ]
                                                      -  [ USERhold ]
                                                         [ SYShold  ]
          Links vaddr
        
          PF [nn ]
        
          Printer [                                           [ HOld     ] ] [ALL ]
          PUnch   [ [ CLass  c] [FORM form ] [DEST dest]      [ NOHold   ] ] [TBL ]
          Reader  [          spoolid                          [ USERhold ] ] [PSF ]
                  [                                           [ SYShold  ] ]
        
          PROCessr
          SCREen
          SECuser
          Set
          SPMODE
          S370E
          TERMinal
          Time
          USERID
        
          {                   [ ALL              ] }
          {                   [ CHannels         ] }
          {                   [ CONsole          ] }
          {                   [ DAsd             ] }
          {                   [ GRaf             ] }
          { [Virtual]         [ LINES            ] }
          {                   [ STORage          ] }
          {                   [ TApes            ] }
          {                   [ UR               ] }
          {                   [ vaddr [ -vaddr ] ] }
        
          VMSAVE                                                                      }
```

Operands and Responses (Class G)

QUERY CPLEVEL

CPLEVEL
displays the system software product name, release number, service level number, nucleus generation time and date, and IPL time and date.

Response:

```
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 into CP.

`GENERATED AT mm/dd/yy hh:mm:ss timezone`
indicates the date and time the CP system software was generated.

`IPL AT mm/dd/yy hh:mm:ss timezone`
indicates the date and time that CP was last started.

QUERY CPUID

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.

Response:

```
CPUID= aabbbbbbccccdddd
```

where:

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

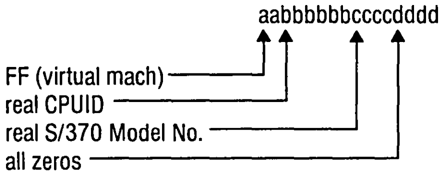
`aa` is the version code; these two digits are forced to X'FF' to identify that the virtual machine is running under VM/SP.

`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 zeros 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
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.

Response:

```
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, DEST, 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.

CLass *c*
selects spool files of class *c* for counting.

FORM *form*
selects spool files with a particular form name for counting.

DEST *dest*
selects spool files with a particular destination for counting.

NOHold
selects spool files that are not held (either by system hold or user hold) for counting.

HOld
selects spool files that are held (either by system hold or user hold) for counting.

USERhold
selects spool files that are in user hold for counting.

SYShold
selects spool files that are in system hold for counting.

*
displays only your spool files.

Note: This is the only option available to class G users.

QUERY LINKS VADDR

Links *vaddr*
displays the user ID, 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.

Response:

```
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 PFnn

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.

Response:

```
                 [tab...    ]
      {IMMED}    [copy...   ]
PFnn  {DELAY}    [pfdatal...]

      RETRIEVE   [BACKWARD]
                 [FORWARD ]
```

The program function defined for a program function key is displayed. If there is no function defined for the program function key, the following message is issued:

```
PFnn UNDEFINED
```

Notes:

1. A logical line-end character imbedded in pfdata1 appears as a carriage return in the response to a QUERY PF*nn* command.
2. In the case of PF*nn* COPY or PF*nn* TAB, the existence of unnecessary or lack of trailing blanks can negate the function and force the PF key to be recognized as pfdata.

QUERY PRINTER, QUERY PUNCH, QUERY READER

Printer
PRT
PUnch
PCH
Reader
RDR

displays the following information, pertaining to your closed virtual reader, virtual printer, and virtual punch spool files:

- User ID (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
- File hold status.

One line of information is displayed for each spool file identification (spoolid).

ALL

displays additional information for spool files. The ALL operand must follow the READER, PRINTER, or PUNCH operand. In addition to the information usually displayed for the reader, printer, or punch files, the following is also displayed:

- Date and time the file was created
- File name and file type of file (if any); if your file was assigned a data set name (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 usually 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
- Number of 4K disk records in the 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.

PSF

besides information usually given, displays additional information about spool files, including:

- Conversion status of a file
  - NOTCONV - The file is in regular CP spool file format.
  - COMPLETE - Special processing is finished, and the file is marked as converted.
  - ACTIVE - The file is being converted now.
- Special status of a file. A spool file is SPECIAL if it contains at least one record with X'5A' CCWs.

Copies and hold status are not displayed when you specify PSF. The PSF option must follow the READER, PRINTER, or PUNCH operand.

spoolid

selects the specified spool file for display.

CLass *c*

selects spool files of class *c* for display.

FORM *form*

selects spool files with a particular form name for display.

DEST *dest*

selects spool files with a particular destination name for display.

NOHold

selects spool files that are not held (either by system hold or user hold) for display.

HOld

selects spool files that are held (either by system hold or user hold) for display.

USERhold

selects spool files that are in user hold for display.

SYShold

selects spool files that are in system hold for display.

Response:

```
           Basic Information

ORIGINID   FILE       CLASS     RECORDS
userid     spoolid    a typ     norecs
```

Additional Information

```
        Query

CPY    HOLD     FORM     DEST
nnn    stat     form     dest

                         Query-ALL

CPY    HOLD     DATE     TIME        NAME     TYPE     DIST
nnn    stat     mm/dd    hh:mm:ss    name     type     distcode

                         Query TBL

FLASH-CNT       FCB    MDFY-C      LOAD     CHARS                       SIZE
ovly    ccc     fcb    mod   n    ⎧NO ⎫     cat   [cat  [cat  [cat]]]   size
                                  ⎨BEG⎬
                                  ⎩ANY⎭

            Query PSF

FORM    DEST    CONVERSION       SPECIAL
form    dest    convstat         ⎧YES⎫
                                 ⎩NO ⎭
```

where:

userid — is the user who originally created the file.

spoolid — is a unique, system-assigned spool file identification number that is used by VM/SP to identify the file.

a — is the spool file class.

typ — is the originating device type (PRT, PUN, CON, DMP, or RDR).

norecs — is the number of logical records contained in the file.

[*]nnn — is the number of copies assigned to the file (it has no effect for virtual reader files). Appearance of the asterisk indicates that spool file duplication is handled by the 3800 printer. The device copies a spool file internally one page at a time (collated output produced).

stat — is the file hold status: NONE (no hold), USER (user hold), SYS (system hold), or USYS (system and user hold).

form is the form name of the spool file.

mm/dd is the date the file was created in month/day.

hh:mm:ss is the time the file was opened for creation in hours:minutes:seconds. For some files, such as spooled console files, this time can be a span of hours before the time at which the file is closed.

name is the file name 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 file type assigned to the file (if any).

distcode is the distribution code assigned to the file.

dest is the destination name assigned to the file.

ovly is the name of the forms overlay frame superimposed on the output.

ccc is the number of copies that are printed while the forms overlay frame is in place.

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.

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.

size is the number of 4K disk records in the file.

convstat is the conversion status of the file, which can be one of the following:

NOTCONV - the file is in regular CP spool file format.
COMPLETE - special processing is finished, and the file is marked as converted.
ACTIVE - the file is being converted now.

YES indicates that the file contains at least one record with X'5A' CCWs.

NO indicates that the file does not contain X'5A' CCWs.

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, file name, file type and file distribution code) is displayed only when you specify the ALL operand. However, if you specify the TBL operand, the information listing FLASH-CNT, FCB, MDFY-C, 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 CCWs within a 3800 spool file), CHARS, and SIZE 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, see the CHANGE command in this section.

QUERY PROCESSR

PROCessr
displays for the user which processors are currently online to VM/SP.

Response:

```
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 SCREEN

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).

Response:

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

SECuser
displays the user ID of the secondary user that is specified in the CONSOLE directory statement.

Response:

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 SET

Set
displays the status of the SET command functions.

Response:

```
                               ┌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 }            {ON }            {ON }
 MIH {OFF} , VMCONIO  {IUCV} , CPCONIO  {IUCV} , SVCACCL  {OFF} , CONCEAL  {OFF}
```

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 SPMODE

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

Response:

```
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 S370E

S370E
indicates that MVS/System Extensions support is active or inactive (not available) on the system.

Response:

```
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 can 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 can 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 TERMINAL

TERMinal
displays the current options in effect for your virtual console environment.

Response:

$$\text{LINEND}\begin{Bmatrix}\text{n}\\ \text{OFF}\end{Bmatrix},\ \text{LINEDEL}\begin{Bmatrix}\text{n}\\ \text{OFF}\end{Bmatrix},\ \text{CHARDEL}\begin{Bmatrix}\text{n}\\ \text{OFF}\end{Bmatrix},\ \text{ESCAPE}\begin{Bmatrix}\text{n}\\ \text{OFF}\end{Bmatrix},\ \text{TABCHAR}\begin{Bmatrix}\text{ON}\\ \text{OFF}\\ \text{CHAR}\end{Bmatrix}$$

$$\text{LINESIZE}\begin{Bmatrix}\text{nnn}\\ \text{OFF}\end{Bmatrix},\ \text{ATTN}\begin{Bmatrix}\text{ON}\\ \text{OFF}\end{Bmatrix},\ \text{APL}\begin{Bmatrix}\text{ON}\\ \text{OFF}\end{Bmatrix},\ \text{TEXT}\begin{Bmatrix}\text{ON}\\ \text{OFF}\end{Bmatrix},\ \text{MODE}\begin{Bmatrix}\text{CP}\\ \text{VM}\end{Bmatrix},\ \text{HILIGHT}\begin{Bmatrix}\text{ON}\\ \text{OFF}\end{Bmatrix}$$

$$\text{CONMODE}\begin{Bmatrix}\text{3215}\\ \text{3270}\end{Bmatrix},\ \text{BREAKIN}\begin{Bmatrix}\text{IMMED}\\ \text{GUESTCTL}\end{Bmatrix},\ \text{BRKKEY}\begin{Bmatrix}\text{PA1}\\ \text{PF1}\\ .\\ .\\ \text{PF24}\end{Bmatrix},\ \text{SCRNSAVE}\begin{Bmatrix}\text{ON}\\ \text{OFF}\end{Bmatrix}$$

$$\text{TYPE}\begin{Bmatrix}\text{3101}\\ \text{TTY}\end{Bmatrix},\ \text{PROMPT}\begin{Bmatrix}\text{VM}\\ \text{TTY}\end{Bmatrix},\ \text{SCROLL}\begin{Bmatrix}\text{nnn}\\ \text{CONT}\end{Bmatrix},\ \text{CNTL}\begin{Bmatrix}\text{USR}\\ \text{SYS}\end{Bmatrix},\ \text{ASCIITBL}\begin{Bmatrix}\text{VM2}\\ \text{VM1}\end{Bmatrix}$$

The settings of all functions 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 you do not have a 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. The last line of the response is only displayed if you are using a TTY device.

QUERY TIME

Time
displays the current time, time zone, weekday, date, connect and processor time for the current terminal session.

Response:

```
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 USERID

USERID
displays the user ID and system identifier (if applicable).

Response:

```
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.

userid
displays the user identification and the terminal device address of the specified user if the user 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).

Response:

```
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/Pass-Through program product, Lnnn displays the logical address of that user.

QUERY VIRTUAL

Virtual

displays the status of all your virtual devices. It is the same as entering QUERY VIRTUAL ALL.

QUERY VIRTUAL ALL

Virtual ALL

displays the status of all your virtual devices.

The response is in the same form 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 CHANNELS

Virtual CHannels

displays the channel mode of operation for the virtual machine.

Response:

$$\text{CHANNELS} = \begin{Bmatrix} \text{SEL} \\ \text{BMX} \end{Bmatrix}$$

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 CONSOLE

Virtual CONsole
displays the status of your virtual consoles. It also displays the SNA logical unit name (luname) and the VSM user ID for SNA user's console devices.

Response:

```
            {    [ luname              ] }
            {    [{GRAF}               ] } {  TERM} {STOP }
CONS vaddr  {ON  [{LINE}         raddr ] } {NOTERM} {START} [VSM vvvvvvvv]
            {DISCONNECTED                }

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

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

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.
dest	is the destination name assigned to each spool file produced from the device.

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 used 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 user ID 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 [VSM vvvvvvvv]
      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

Virtual DAsd
displays the status of all your virtual direct access storage devices.

Response:

$$\text{DASD vaddr type volser} \begin{Bmatrix} \text{R/W} \\ \text{R/O} \end{Bmatrix} \begin{Bmatrix} \text{nnnnnn BLK} \\ \text{nnnn CY} \end{Bmatrix}$$

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.

`nnnn` 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 GRAF

Virtual 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 logical unit name (luname) and VTAM service machine (VSM) user ID for each SNA device accessing your virtual machine using the CP DIAL command.

Response:

```
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 logical unit name (luname) and VTAM service machine (VSM) user ID 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 by the DIAL command.

luname is the logical unit name of the SNA device accessing your virtual machine.

vvvvvvvv is the user ID of the VSM controlling the SNA device.

If there are no locally and remotely attached display devices and graphics devices dedicated to your virtual machine, the response is:

```
GRAF DO NOT EXIST
```

QUERY VIRTUAL LINES

Virtual LINES
displays the status of all your virtual communication lines.

Response:

```
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.

LINE vaddr $\left\{\begin{matrix}\text{ENABLED}\\\text{DISABLED}\end{matrix}\right\}$

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

QUERY VIRTUAL STORAGE

Virtual STORage
displays the size of your virtual storage.

Response:

```
STORAGE = nnnnnK
```

The size of the virtual machine in multiples of 1024 bytes is displayed.

QUERY VIRTUAL TAPES

Virtual TApes
displays the status of all your virtual magnetic tape devices.

Response:

```
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.

If there are no virtual magnetic tape devices attached to your virtual machine, the response is:

```
TAPES DO NOT EXIST
```

QUERY VIRTUAL UR

Virtual UR
displays the status of all your unit record devices.

Response:

```
               {  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 EOF READY

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

      {TO }
vaddr {FOR} userid DIST distcode DEST dest

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

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

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

                  {2WCGM} {BTS} {DATCK  } {3800-1}
vaddr SIZE ww 11  {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 as usual on the printer assigned in the spool file class.

Note: Further information can be obtained under the SPOOL command "Usage Notes."

`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.

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.

dest — is the destination name 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.

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  DEST OFF
PRT vaddr CL A NOCONT NOHOLD COPY 001 READY FORM form
    vaddr  FOR  userid  DIST distcode  FLASHC 000  DEST OFF
    vaddr FLASH     CHAR     MDFY     FCB
    vaddr  SIZE 0F 11  4WCGM  CFS  NODATCK  3800-1
```

where:

userid and distcode are assigned for the virtual machine, and form is the installation defined default value.

QUERY VIRTUAL vaddr

Virtual *vaddr*
displays the status of the virtual device at address *vaddr*.

If you QUERY VIRTUAL *vaddr* and the device does not exist, the response

```
DEVS DO NOT EXIST
```

is displayed.

QUERY VIRTUAL vaddr-vaddr

Virtual *vaddr-vaddr*

displays the status of the devices within the range of addresses specified.

The response is in the same form as QUERY VIRTUAL DASD, TAPES, LINES, GRAF, CONSOLE, 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 VMSAVE

VMSAVE

displays information about all of the target VMSAVE areas specified in module DMKSNT that the user either owns or receives.

Response:

```
   USER    RECEIVER SYSNAME PRTY IPLD  ENBLD IN USE DATE/TIME VOLID
   userid  userid   sysname nnnn {YES} {YES} {YES}   {NONE}   volid
 0001-0002 0006                  {NO } {NO } {NO }   {N/A }
```

where:

`USER` is the user ID of the user who can store into the area or IPL the area (the owner of the area).

`RECEIVER` is the user ID 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 Administration* book for more information regarding the specification and use of the VMSAVE option.

Messages

DMKCQG022E Virtual device address missing or invalid
DMKCQG040E Device *vdev* does not exist
DMKCQG045E *userid* not logged on
DMKCQG361E LOGOFF/FORCE pending for user *userid*
DMKCQH024E {FORM number|DEST} missing or invalid
DMKCQH028E Class missing or invalid
DMKCQH042E Spoolid *nnnn* does not exist
DMKCQP003E Invalid option - *option*
DMKCQP023E Volid missing or invalid
DMKCQP040E Device *rdev* does not exist
DMKCQP466I CTLR *rdev* IN BUFFER SLOWDOWN MODE
DMKCQQ006E Invalid device type - *devtype*
DMKCQQ021E Real device address missing or invalid
DMKCQQ022E Virtual device address missing or invalid
DMKCQQ040E Device *rdev* does not exist
DMKCQQ045E *userid* not logged on
DMKCQQ361E LOGOFF/FORCE pending for user *userid*
DMKCQR020E Userid missing or invalid
DMKCQR026E Operand missing or invalid
DMKCQR045E *userid* not logged on
DMKCQS003E Invalid option - *option*
DMKCQT002E Invalid operand - *operand*
DMKCQT006E Invalid device type - *devtype*
DMKCQT021E Real device address missing or invalid
DMKCQT747A Device *rdev* VARY in progress, please re-enter this command
DMKCQT747A CSW=(*csw*)[, 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
DMKCQY045E *userid* not logged on
DMKCQY361E LOGOFF/FORCE pending for user *userid*
DMKCQZ026E Operand missing or invalid
DMKPEI047E {|The current traceset |The traceset *traceset*|A 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. If the virtual device is a 4245 or 4248 printer, the READY command sets a device-end, unit exception and an attention interrupt.

Format

READY	*vaddr*

Operand

vaddr
is a virtual device address (cuu).

Usage Note

If the virtual device is in a NOTREADY status, it is changed to a READY status.

Responses

None.

Messages

DMKCPB006E	Invalid device type - *devtype*
DMKCPB022E	Virtual device address missing or invalid
DMKCPB040E	Device *vdev* does not exist
DMKCPB087I	Device *vdev* is in the process of being reset
DMKCPB089I	Device *vdev* is in the process of being detached

REQUEST

Privilege Class: G

Use the REQUEST command to make an attention interrupt pending at your virtual console.

Format

REQuest	

Usage Notes

1. The REQUEST command performs the same functions as ATTN, and the two commands can be interchangeably used.
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 can result if the user enters 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.

Format

RESET	*vaddr*

Operand

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. The 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 entered, it is not reset to the defaults.
4. When you issue RESET to a virtual CTCA or virtual 3088 (VCTCA/V3088), a selective reset is simulated. When you issue RESET to a dedicated CTCA or D/T3088, a HDV (halt device) is issued to the hardware. This causes an interface disconnect.

Response

```
DEVICE RESET
```

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

Messages

DMKCFQ546I	Interruption pending; device *rdev*, CSW *csw*, userid *userid*
DMKCPB022E	Virtual device address missing or invalid
DMKCPB040E	Device *vdev* does not exist
DMKCPB087I	Device *vdev* is in the process of being reset
DMKCPB089I	Device *vdev* is in the process of being detached

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.

Format

REWind	*vaddr*

Operand

vaddr
is the virtual device address (cuu) of the tape unit to rewind.

Usage Note

The REWIND command accomplishes the manual operation of rewinding and making the tape ready at the tape unit.

Response

```
REWIND COMPLETE
```

is the response you receive upon the successful initiation of the command. Receiving this message does not imply that the tape is physically rewound yet.

Messages

DMKCPB006E	Invalid device type - *devtype*
DMKCPB022E	Virtual device address missing or invalid
DMKCPB040E	Device *vdev* does not exist
DMKCPB899I	REWIND NOT PERFORMED

SCREEN

Privilege Class: G

Use the SCREEN command to change any extended color or extended highlight definitions for your virtual machine console. You can enter 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 valid only 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 see the *VM/SP Terminal Reference*.

Format

SCREen	$\left\{ \mathit{area} \left\{ \left[\begin{Bmatrix} \mathit{extcolor} \\ \underline{\textbf{DEFault}} \end{Bmatrix} \quad \begin{Bmatrix} \mathit{exthilight} \\ \underline{\textbf{NONe}} \end{Bmatrix} \right] \right\}^{1} \right\}$

[1] Each time you enter the command, you must specify at least one screen *area* operand with at least one *extcolor* or *exthilight* value. You can specify more than one *area* operand on the same command line.

Operands

area
specifies the area of the screen. The *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 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
<u>DEFault</u> — 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
<u>NONe</u> — no extended highlighting.

Usage Notes

1. If you issue the SCREEN command from a terminal that does not support extended color and 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 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:

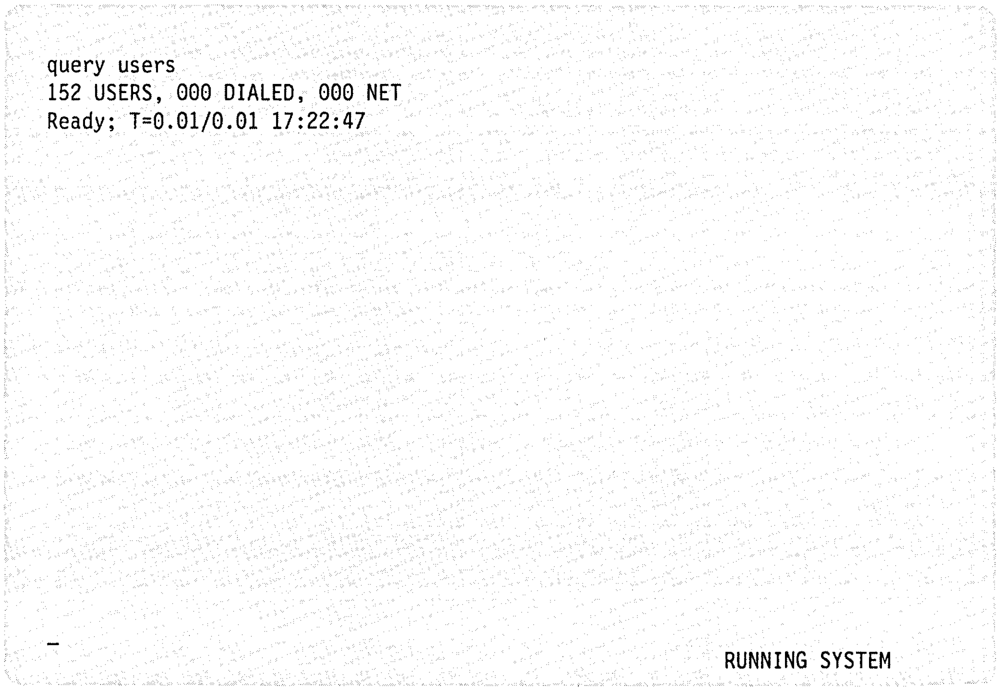
```
screen inredisp yellow none cpout blue none vmout red none

screen inarea green blink statarea pink blink
```

CP sets the following values for your terminal:

Area	Color	Extended Highlight
cpout	blue	none
vmout	red	none
inredisp	yellow	none
inarea	green	blinking
statarea	pink	blinking

Using the previous settings, the display areas on your screen would look similar to the following after issuing the QUERY USERS command:



```
query users
152 USERS, 000 DIALED, 000 NET
Ready; T=0.01/0.01 17:22:47










_
                                                  RUNNING SYSTEM
```

where:

`query users`
is the INREDISP area (yellow), redisplaying the command that was entered.

`152 USERS, 000 DIALED, 000 NET`
is the CPOUT area (blue), displaying the CP response to the QUERY USERS command.

`Ready; T=0.01/0.01 17:22:47`
is the VMOUT area (red). In this example, the VMOUT area displays the output from CMS after the command has executed.

_ (Cursor)
is the INAREA (green and blinking)

RUNNING SYSTEM
is the STATAREA (pink and blinking)

Response

```
COMMAND COMPLETE
```

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 Facilities for Programming* book.

Format

SEND	[CP] {*userid* *lprt*} [*text*]

Operands

CP

indicates that CP executes the text for the receiving virtual machine. If the first 3 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.

Note: CP does not have meaning when a logical printer name is specified instead of a *userid.*

userid

represents the disconnected virtual machine that has a specified *userid* of the secondary user defined in its directory entry.

lprt

is the name of a logical printer. When you specify *lprt*, the text is sent to the logical printer that you specify.

text

when specified with *userid*, 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 you want to send multiple lines in the text, use the logical line end (#) immediately preceded by the logical escape character ("); in addition, you must have line editing enabled in your virtual machine. Note that this restriction also applies when '#CP command' is the command text because the logical line-end character is in fact a carriage return, implying multiple lines.

When text is specified with *lprt*, it can be any data recognized by the receiving logical printer. The unedited text is passed to the logical printer to process.

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 line-editing character is sent in the SEND command *text*, the character must be that of the sender, not the disconnected machine.
4. The secondary user must not be disconnected when the SEND command is entered.
5. To send information to a specific logical printer, enter a SEND command. For example, you can specify

   ```
   send lprt query lprt
   ```

   where *lprt* is the logical printer name and query lprt is the *text*.

Responses

None.

Messages

DMKSND020E	Userid missing or invalid
DMKSND045E	*userid* not logged on
DMKSND068E	SEND command failed - receiver {is not disconnected | has not authorized sender | has no virtual console | console input waiting | is executing a console function}

SET

Privilege Class: G

Use the SET command to control various functions within your virtual system.

Format

```
SET  ┌ ACNT       { ON  }                                          ┐
     │            { OFF }                                          │
     │ AFFInity   [ ON  ]                                          │
     │            [ OFF ]                                          │
     │ ASsist     { [ON]   [ SVC   ]  [ TMR   ] }                  │
     │            {        [ NOSVC ]  [ NOTMR ] }                  │
     │            {  OFF                        }                  │
     │ AUTOPoll   { ON  }                                          │
     │            { OFF }                                          │
     │ CONceal    { ON  }                                          │
     │            { OFF }                                          │
     │ CPCONIO    { OFF  }                                         │
     │            { IUCV }                                         │
     │ CPUid        bbbbbb                                         │
     │ ECmode     { ON  }                                          │
     │            { OFF }                                          │
    {  EMSG       { ON   }                                          }
     │            { OFF  }                                         │
     │            { CODE }                                         │
     │            { TEXT }                                         │
     │            { IUCV }                                         │
     │ IMSG       { ON   }                                         │
     │            { OFF  }                                         │
     │            { IUCV }                                         │
     │ ISAM       { ON  }                                          │
     │            { OFF }                                          │
     │ LINEDit    { ON  }                                          │
     │            { OFF }                                          │
     │ MIH        { ON  }                                          │
     │            { OFF }                                          │
     │ MSG        { ON   }                                         │
     │            { OFF  }                                         │
     └            { IUCV }                                         ┘
```

```
 ┌ NOTRans     { ON  }                                               ┐
 │             { OFF }                                               │
 │ PAGEX       { ON  }                                               │
 │             { OFF }                                               │
 │ PFnn   [ COPY (resid)  ]                                          │
 │ PFnn   [ COPY (cuu)    ]                                          │
 │ PFnn   [ COPY (luname) ]                                          │
 │ PFnn   [ COPY (Laddr)  ]                                          │
 │ PFnn   [ IMMed   ] [pfdata1 #pfdata2 #...pfdatan ]                │
 │        [ DELayed ]                                                │
 │ PFnn  RETrieve  [ backward ]                                      │
 │                 [ FORward  ]                                      │
 │ PFnn  [TAB n1  n2  ... ]                                          │
 │ RUN         { ON  }                                               │
 │             { OFF }                                               │
 │ SMsg        { ON   }                                              │
 │             { OFF  }                                              │
 │             { IUCV }                                              │
 │ STBypass    { [nnnnnK  [ NOVERIFY ]] }                            │
 │             { [nnM     [          ]] }                            │
 {             {                        }                            }
 │             {  VR                    }                            │
 │             {  OFF                   }                            │
 │ STMulti     { [n  [ [ USEG xx ]  CSEG yyy ]] }                    │
 │             { [ON [ [         ]           ]] }                    │
 │             {  OFF                           }                    │
 │ SVCAccl     { ON  }                                               │
 │             { OFF }                                               │
 │ TIMER       { ON   }                                              │
 │             { OFF  }                                              │
 │             { REAL }                                              │
 │ VMCONIO     { OFF  }                                              │
 │             { IUCV }                                              │
 │ VMSAVE      { ON   }                                              │
 │             { OFF  }                                              │
 │             { name }                                              │
 │ WNG         { ON   }                                              │
 │             { OFF  }                                              │
 │             { IUCV }                                              │
 │ 370E        { ON  }                                               │
 └             { OFF }                                               ┘
```

Operands (Class G)

ACNT $\left\{\begin{matrix}\text{ON}\\\text{OFF}\end{matrix}\right\}$

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.

AFFInity $\begin{bmatrix}\text{ON}\\\text{OFF}\end{bmatrix}$

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 non-IPL processor of a multiprocessor system. Usual 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 is no 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 for a virtual machine are not restricted to a processor by an affinity setting. See also SET ASSIST.

ASsist $\left\{\begin{array}{ll}{[}\text{ON}{]} & \begin{bmatrix}\underline{\text{SVC}}\\\text{NOSVC}\end{bmatrix}\begin{bmatrix}\text{TMR}\\\underline{\text{NOTMR}}\end{bmatrix}\\ {[}\text{OFF }{]} & \end{array}\right\}$

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 usually 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 non-IPL processor that the virtual machine is logged on. (For the extent of VM/SP hardware assist available on supported VM/SP processors, see 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 CP System Command Reference* book 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 Administration* book for additional information on how to use VM/SP hardware assist.

Note:

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 the user's processor. In attached processor or multiprocessor setups where the user has no processor preference (AFFINITY=NONE) upon entering SET ASSIST the user 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 the affinity setting, then initiate the SET ASSIST command.

AUTOPoll $\left\{\begin{matrix}\textbf{ON}\\\textbf{OFF}\end{matrix}\right\}$

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.

CONCeal $\left\{\begin{matrix}\textbf{ON}\\\textbf{OFF}\end{matrix}\right\}$

controls whether or not the user is placed in a protected application. SET CONCEAL ON interactively places the user in a protected application and defines the break key as NONE on a 3270-type terminal. SET CONCEAL OFF removes the user from the protected application and returns the break key to its default setting for 3270-type terminals (PA1). When you log on VM/SP, CONCEAL is set according to the OPTION control statement in your directory entry: ON if CONCEAL was specified and OFF if not.

When you operate in a protected application:

- Multiple attentions will not cause you to enter CP mode.
- The terminal break key is set to NONE.
- CP initiates an automatic re-IPL when it encounters errors, such as virtual machine disabled wait, paging error, invalid PSW, external interrupt loop, program interrupt loop, and translation exception.
- If a shared page is altered, CP attempts to resume execution in the virtual machine before initiating an automatic re-IPL.
- CP limits re-IPL attempts to avoid keeping the user in a re-IPL loop. If an error causing a re-IPL attempt occurs within one minute of the last re-IPL attempt or occurs when ten re-IPLs have been attempted since entering a protected application environment, the user is placed in CP mode and an error message is displayed.

Note: For a system operator, a virtual machine disabled wait does not initiate an automatic re-IPL.

CPCONIO $\left\{\begin{matrix}\textbf{IUCV}\\\textbf{OFF}\end{matrix}\right\}$

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 through 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.

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 machines 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 to 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.

ECmode $\left\{\begin{matrix}\textbf{ON}\\ \textbf{OFF}\end{matrix}\right\}$

controls whether the virtual machine operating system can use System/370 extended control mode and control registers 1 through 15. Control register 0 can 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.

$$\textbf{EMSG}\ \left\{\begin{array}{l}\textbf{ON}\\ \textbf{OFF}\\ \textbf{CODE}\\ \textbf{TEXT}\\ \textbf{IUCV}\end{array}\right\}$$

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 displayed. CODE specifies that only the error code is to 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.

$$\textbf{IMSG}\ \left\{\begin{array}{l}\textbf{ON}\\ \textbf{OFF}\\ \textbf{IUCV}\end{array}\right\}$$

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. SET IMSG also controls the displaying of informational messages generated by CP when the SPOOL command closes an output device. 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. Spooling messages generated by users sending punch, printer, or reader files to another virtual machine are also suppressed if OFF is specified. 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 through 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 entered.

$$\textbf{ISAM}\ \left\{\begin{array}{l}\textbf{ON}\\ \textbf{OFF}\end{array}\right\}$$

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.

LINEDit $\left\{\begin{array}{l}\textbf{ON}\\\textbf{OFF}\end{array}\right\}$

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 usually 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.

MIH $\left\{\begin{array}{l}\textbf{ON}\\\textbf{OFF}\end{array}\right\}$

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.

MSG $\left\{\begin{array}{l}\textbf{ON}\\\textbf{OFF}\\\textbf{IUCV}\end{array}\right\}$

controls whether messages sent with the MSG command from other users are received at the terminal. If ON is specified, the messages are displayed. If OFF is specified, no messages are received. When you log on VM/SP, MSG is set ON. If IUCV is specified, the message is passed to the virtual machine through 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 entered.

NOTRans $\left\{\begin{array}{l}\textbf{ON}\\\textbf{OFF}\end{array}\right\}$

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 is active
- The first CCW is 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 nondedicated device (spooled unit record console virtual CTCA (channel to channel adapter) or minidisks that are less than a full volume)
- I/O device has an alternate path
- Device status is pending.

Any of the preceding conditions forces CCW translation. For additional information, see the *VM/SP Administration* book.

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 $\left\{\begin{array}{l}\textbf{ON}\\\textbf{OFF}\end{array}\right\}$

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.

PF*nn* [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 pressed. The output is printed on an IBM printer connected to the same control unit as your display terminal; if no printer is connected, the copy function is not available. The PF key COPY function is not available for dedicated 3284, 3286, 3287, 3288, and 3289 printers. 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 resource identification number. The resource identification 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.

PF*nn* [COPY (*cuu*)]
specifies that the designated program function key, PF*nn*, 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.

PF*nn* [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 PF*nn* COPY function can be used for any printer managed by VM/VTAM, that is, 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).

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:
   a. The printer is already busy, or all printers are busy.
   b. The printer is turned off.
   c. The printer is operational but has not been varied online by the system operator.
   d. The printer is out of paper or is in any other intervention-required condition.
   e. The designated printer does not support the copy functions.
   f. 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.

PF*nn* [COPY (L*addr*)]
specifies that the program function key numbered *nn* performs a COPY function for the logical device L*addr*. The number *nn* must be a value between 1 (or 01) to 24. The character L must precede the device address.

PF*nn* $\begin{bmatrix}\textbf{IMMed}\\ \underline{\textbf{DELayed}}\end{bmatrix}$ [*pfdata1#pfdata2#...pfdatan*]

defines a program function for a program function key on a 3270 Information Display station and indicates when that function is to be executed. If PF*nn* 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 PF*nn*, *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 nonimmediate 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 ("). If no command lines are entered following the PF*nn* designation, then the program function is marked undefined. Program functions cannot be embedded within one another.

```
PFnn    RETrieve    [backward]
                    [FORward ]
```

specifies that the program function key numbered *nn* carries out the RETRIEVE BACKWARD or FORWARD function on a 3101, 3270, or VTAM display terminal.

On terminals with 12 PF keys, *nn* must be a number from 01 to 12. On terminals with 24 PF keys, *nn* must be a number from 01 to 24. (Leading zeros are optional.)

Once you have defined a RETRIEVE program function key, CP begins saving your input lines. When you press the designated PF key, VM/SP redisplays either the most recently saved or the oldest saved input line (depending on which RETRIEVE option you choose), so that you can modify and reenter the data.

VM/SP can actually keep track of several input lines. The buffer is a fixed size so the number of lines saved depends on the length of the lines. As the buffer fills up, it discards older input lines as it stores newer ones. If you enter a duplicate line (one which is exactly the same as the line immediately preceding it) the buffer does not store it, since it is not useful to retrieve the same line twice. For security reasons, the system does not store input lincs not displayed on the terminal, such as passwords.

BACKWARD specifies that the PF key will retrieve input in a backward direction.

When you first press the RETRIEVE BACKWARD program function key, VM/SP displays the most current input line. If you press the key again, VM/SP displays the previous input line. Each time you press the key, VM/SP scrolls through the input lines, retrieving older input lines, and displaying them one at a time. After VM/SP reaches the oldest line stored, it returns to the most current line once again.

VM/SP resets the RETRIEVE function pointer each time you enter a new input line.

BACKWARD is the default of the SET PF RETRIEVE command. In fact, the function only tests for valid abbreviations of the FORWARD parameter. An abbreviation for the FORWARD parameter must contain at least the first three letters of the word; thus, FOR, FORW, etc., are valid abbreviations. This means that *anything* you enter, other than a valid form of the option FORWARD, defaults to BACKWARD.

FORWARD specifies that the PF key will retrieve input in a forward direction.

When you first press the designated RETRIEVE FORWARD PF key, VM/SP displays the oldest saved input line. If you press that PF key repeatedly, VM/SP scrolls forward from the oldest saved input line, displaying lines one at a time, until it displays the most recently saved input line. It then returns to the oldest saved line of input once again.

VM/SP only saves input lines when you have a program function key defined for the RETRIEVE function, and are logged on at a 3270 data screen compatible terminal or at a 3101 using block mode. The system begins recording input *immediately after* you issue the first SET PF RETRIEVE command of your terminal session.

While PF keys retain their function, the buffer that contains your "input history" does not retain its contents during or after a terminal disconnect.

PF*nn* [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. TAB is a keyword identifying the tab function. The tab settings (*n1 n2*...) may be entered in any sequence. For example,

```
set pf09 tab 20 30 40
```

defines tab settings on columns 20, 30, and 40 of your command line.

If the TAB keyword and tab settings are not entered, that program function key is considered "undefined." When defining the TAB function, the existence of more than one blank preceding the keyword TAB or the lack of trailing tab settings forces the PF key to be treated as pfdata.

Notes:

1. Both SET PF*nn* TAB and SET PF*nn* COPY are executed immediately upon pressing the appropriate program function key. If you insert the keywords DELAYED or IMMED after the PF*nn* operand, the command is accepted, but the keywords are ignored. The program function still executes immediately.

2. If you use the SET PF*nn* command to set up a series of concatenated commands, you should be careful of line-editing characters. The following examples demonstrate two methods for entering a series of commands:

   Example 1

   Enter the following command while in CMS mode:

   ```
   set linedit on
   ```

   and press the enter key.

   Now enter one of the following commands while in CMS mode:

   ```
   #cp set pf02 immed q rdr#q prt#q pun
                -- or --
   cp set pf02 immed q rdr"#q prt"#q pun
                -- or --
   set pf02 immed q rdr"#q prt"#q pun
   ```

   and press the enter key.

   CP assigns the three QUERY commands as functions of the PF02 key. Pressing the PF02 key executes the three QUERY commands.

   Example 2

   Enter the following command while in CMS mode:

   ```
   set linedit off
   ```

   and press the enter key.

Then enter:

```
set pf02 immed q rdr#q prt#q pun
             -- or --
cp set pf02 immed q rdr#q prt#q pun
```

and press the enter key.

CP assigns the three QUERY commands as functions of the PF02 key.

Then enter:

```
set linedit on
```

and press the enter key.

Pressing the PF02 key executes the three QUERY commands.

3. If a terminal has an inhibited (nondisplay) 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.

RUN $\left\{\begin{matrix}\textbf{ON}\\\textbf{OFF}\end{matrix}\right\}$

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.

Notes:

1. 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.
2. If you use the SET RUN ON, it is possible that command output may be interrupted by a guest virtual machine read or write.

SMsg $\left\{\begin{matrix}\textbf{ON}\\\textbf{OFF}\\\textbf{IUCV}\end{matrix}\right\}$

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. When you log on VM/SP, SMSG is set OFF.

```
STBypass [[nnnnn  K [NOVERIFY]]]
          [nn  M              ]
          VR
          OFF                  ]
```

allows V=R users to eliminate and reëstablish shadow tables. Allows V=V users to reduce shadow-table processing.

The *nnnnn*K 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.

The *nn*M 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.

The *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.

The USEG *xx* defines the size of the contiguous preallocated pool of shadow page tables. The *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.

The 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 (for example, MVS, SVS, and so forth) is running in a virtual machine.

SVCAccl $\left\{\begin{matrix}\textbf{ON}\\\textbf{OFF}\end{matrix}\right\}$

setting SVCACCL ON activates the acceleration of SVC 9, 10, and 11. To use this support, the SVCACCL parameter of the OPTION control statement must be included in your VM/SP directory entry. Setting SVCACCL ON reduces the overhead incurred when IX/370 is running in a virtual machine. If it is available, the IX/370 assist is also available.

To determine whether this option is in effect, issue the QUERY SET command.

TIMER $\left\{\begin{matrix}\textbf{ON}\\\textbf{OFF}\\\textbf{REAL}\end{matrix}\right\}$

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 runtime 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.

Note:

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.

VMCONIO $\left\{\begin{matrix}\textbf{IUCV}\\\textbf{OFF}\end{matrix}\right\}$

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 (for example, 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.

VMSAVE $\left\{\begin{matrix}\textbf{ON}\\\textbf{OFF}\\\textit{name}\end{matrix}\right\}$

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 Administration* and the *VM/SP Planning Guide and Reference* for more information.

$$\textbf{WNG}\quad \left\{\begin{array}{l}\textbf{ON}\\ \textbf{OFF}\\ \textbf{IUCV}\end{array}\right\}$$

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.

$$\textbf{370E}\quad \left\{\begin{array}{l}\textbf{ON}\\ \textbf{OFF}\end{array}\right\}$$

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.

Responses

None.

Messages

DMKCFJ002E	Invalid operand - *operand*
DMKCFJ004E	Invalid hexloc - *operand*
DMKCFO003E	Invalid option - *option*
DMKCFO020E	Userid missing or invalid
DMKCFO026E	Operand missing or invalid
DMKCFO045E	*userid* not logged on
DMKCFO175E	{RESERVE|*name*} 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 processor *xx*]
DMKCFO255W	370E feature now disabled
DMKCFS003E	Invalid option - *option*
DMKCFS006E	Invalid device type - *devtype*
DMKCFS026E	Operand missing or invalid
DMKCFS044E	System *sysname* does not exist
DMKCFS080E	No preallocated VMSAVE area
DMKCFS081E	VMSAVE area already in use, must use SET command
DMKCFS082E	More than one VMSAVE area allocated, must specify area name

DMKCFS086E	I/O error on DASD
DMKCFS170E	System *sysname* exceeds storage
DMKCFS171E	System *sysname* volid *volid* not mounted
DMKCFS175E	{RESERVE|*name*} already in use by *userid*
DMKCFS179E	System *sysname* volid *volid* not CP-owned
DMKCFS250E	370E is not available [on processor *xx*]
DMKCFS251E	370E feature is not active
DMKCFS420E	No SVCACCL specified in directory
DMKCFV003E	Invalid option - *option*
DMKCFV026E	Operand missing or invalid
DMKCFV062E	ECMODE not set on
DMKCFV063E	Shadow table bypass; not set
DMKCFV065E	Highest allowable high-water mark value is x*xxxx*k
DMKCFW002E	Invalid operand - *operand*
DMKCFW026E	Operand missing or invalid
DMKCFY003E	Invalid option - *option*
DMKCFY026E	Operand missing or invalid
DMKCFY045E	*userid* not logged on
DMKCFY052E	Error in CP directory
DMKCFY053E	*userid* not in CP directory
DMKCFY183E	VM assist not active
DMKCFY184E	VM assist not available on processor *xx*
DMKCFY187E	Timer assist not available
DMKCFY188E	Specified processor unavailable
DMKCFY189E	AFFINITY not set; system in uniprocessor mode
DMKCFY190E	No AFFINITY specified in directory
DMKJRL145I	USERID. userid AT rdev WITH INVALID {LINK|LOGON} PASSWORD NO. nnn HAS VIOLATED THE LIMIT OF limit
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 automatically awakened when the specified interval has elapsed.

Format

SLeep	[*nn* [SEC MIN HRs]]

Operands

nn [SEC
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 enter 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 enter 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 is reset if the LOGON or DISCONN command is chained to it.
8. If the SLEEP command is entered while in a full-screen application, it results in a hung terminal condition. Any write to the terminal unlocks the keyboard.

Responses

None.

Message

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.

Format

SMsg	*userid msgtext*

Operands

userid
is the receiving virtual machine's logon ID; if you specify *userid* as an asterisk (*), this indicates the message is to be sent to yourself.

msgtext
is the text of the message 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, if you are using a display device that has a 139 character input buffer, such as a 3279 Mod 2A, and you enter:

$$\underbrace{\text{smsg pd58ghlc}}_{\text{14 characters used}} \qquad \underbrace{\text{special message}}_{\text{125 characters left}}$$

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

Usage Notes

1. When the SMSG command is issued by DIAGNOSE X'8', CP reads only the first 240 characters.
2. 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.
3. 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 programmed virtual machine processes the instructions. If SET SMSG IUCV is in effect, the message is sent to the receiving virtual machine from IUCV instead of VMCF.
4. Virtual machines can receive and process special messages only under these conditions:

   - The virtual machine has issued the SET SMSG ON command, and the user wishing to receive special messages has entered 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 entered an IUCV CONNECT to the CP Message System Service of VM/SP.

Note: For more information on VMCF and IUCV, see the *VM/SP System Facilities for Programming* book. For more information on the ALLOW option of IUCV, see the *VM/SP Planning Guide and Reference.*

Responses

None.

Messages

DMKMSG045E *userid* not logged on
DMKMSG057W *userid* not receiving; {disconnected|MSG off|message too long|SMSG off| not authorized|WNG off|IUCV error *nnnn*|VMCF error *nnnn*|quiesced}

SPOOL

Privilege Class: G

Use the SPOOL command to change 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. See ″Transmitting Files to Remote Locations,″ page 220. The default values for the SPOOL command appear under “Usage Notes.”

Format

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

        { Printer } ( [ [To ] { userid } ] [HOld  ] [CONt  ] [CLass c] [COpy[*]nnn] 1 )
        { PUnch   } ( [ [For] {   *    } ] [NOHold] [NOCont]                         )
        { CONsole } ( [       { SYSTEM } ]                                           )
        { vaddr   } ( [ OFF              ]                                           )
                    (                                                                )
                    ( [CLOSE] [STArt] 2  [FORM { form }] [DEST { dest }] [TErm  ] 2 )
                    ( [PURGE] [STop ]    [     { OFF  }] [     { OFF  }] [NOTErm]   )
                    (                                                                )
                    ( [ 3                                           ]  [Dist { distcode }] )
                    ( [ FLash name { nnn }                          ]  [     { OFF      }] )
                    ( [            {  0  }                          ]                      )
                    ( [ MOdify { name [n] }                         ]                      )
                    ( [        { NULL     }                         ]                      )
                    ( [ CHars { name1 [name2 [name3 [name4 ]]] }    ]                      )
                    ( [       { NULL                           }    ]                      )
                    ( [ CHars name1                                 ]                      )
                    ( [   [CHars name2                              ]                      )
                    ( [   [CHars name3                              ]                      )
                    ( [   [CHars name4]]]                           ]                      )
                    ( [ FCB { name }                                ]                      )
                    ( [     { NULL }                                ]                      )
```

[1]You must select at least one of the options within braces. However, you can specify more than one and enter them in any order on the command line.
[2]These options apply only to a virtual spooled console.
[3]Use these options only to modify a virtual spooling printer. These options apply only to a 3800 printer as a virtual spooling device.

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; starts or stops the spooling of virtual console input and output, including CP I/O.

vaddr
is the virtual device address (cuu) of the virtual unit record device or console whose options are to be modified.

Class $\left\{\begin{matrix} c \\ * \end{matrix}\right\}$

specifies the spool class of the device. The *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. CLOSE requests for a punch or printer for which the CONT operand has been specified do not close the punch or printer. The file name and file type assigned to the spool file is that of the last file printed. If CONT is specified, NOCONT may not be specified.

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 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 may not 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. Change the status of files with the CHANGE command. Change the status of output devices with the SPOOL command. If HOLD is specified, NOHOLD may not 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 *) 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 are 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, who receives only one copy. However, if you specify OFF or FOR on a subsequent command, the receiver of your spool file receives the number of copies you specify with COPY. For example, if you enter the command

```
spool pun to usera copy 3 class b
```

the COPY operand has no effect on the file going to USERA. However, if you enter

```
spool pun off
  -- or --
spool pun for usera
```

the COPY 3 specified in the first statement affects subsequent punch files.

TO SYSTEM is equivalent to specifying OFF and resets the transferred spool option.

For $\left\{\begin{array}{l}\textit{userid}\\ *\\ \textbf{SYSTEM}\end{array}\right\}$

indicates the *userid* under which printed or punched output is produced. The *userid* becomes the owner of the output spool file and the distcode (distribution code) on the file is the distcode for the user 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.

When you specify the DIST option with FOR *userid*, the files created have the *distcode* from the VM/SP directory entry for that user. However, your virtual device is updated to the *distcode* specified on the command. You can change the *distcode* only for your own virtual device. For example, if you enter

```
spool pun usera dist 338
```

USERA becomes the owner of the output punch files and your virtual punch *distcode* is updated to 338.

When you specify the DIST option with FOR * or FOR SYSTEM, your virtual device is updated to the *distcode* specified on the command.

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 (you need not specify leading zeros). When you specify the optional asterisk (*), 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 as usual.

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. For example, the 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 may not 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* $\left\{\begin{matrix} nnn \\ \underline{0} \end{matrix}\right\}$

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 through 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 $\left\{\begin{matrix} name\ [n] \\ \textbf{NULL} \end{matrix}\right\}$

assigns a copy modification module to use when printing a file on the 3800. This function makes it possible to change text either by 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 without modification. The *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.

NULL clears the MODIFY value previously entered.

CHars $\left\{\begin{matrix} name1\ [name2\ [name3\ [name4]]] \\ \textbf{NULL} \end{matrix}\right\}$

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.

NULL clears the CHARS value previously entered.

The multiple form of CHARS must be used if the *name* value conflicts with an option name.

$\textbf{FCB}\begin{Bmatrix}\textit{name}\\ \textbf{NULL}\end{Bmatrix}$

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.

NULL clears the FCB value previously entered.

STArt
places all console input and output in a spool file. Until a CLOSE is issued for the console, you can change characteristics of the console spool file with the SPOOL CONSOLE command. After the console is closed, the file becomes a printer spool file whose characteristics you can change with the CHANGE PRINTER command.

STop
stops 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 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. The *form* is a one- to eight-character 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.

DEST *dest*
specifies the destination name for spool files created on this virtual device. The *dest* is a one- to eight- character alphameric name your installation assigns to printers and punches at system generation.

DEST OFF
resets the destination name for the virtual device to the default value of OFF.

Dist *distcode*
specifies the distribution code for spool files created on this virtual device. The *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 print certain classification titles automatically (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 or she 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* the operator specifies 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 as usual. 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 by using the CHANGE command or can 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 can recode customized 3800 Model 1 character sets and build new modules with the GENIMAGE command. The MVS Character Conversion Aid can 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 with the SPOOL and TRANSFER commands. To transfer printer or punch files of your virtual machine to the virtual reader of some other user, enter 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.

To print or punch a file for someone else, you can enter 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, enter 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.

To reclaim a transferred file, enter the same command with the FROM option:

```
transfer option from userid
```

where option and *userid* are the same as in 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 that 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.

The SPOOL command indicates that output to the device specified in the command is to be sent to the RSCS virtual machine, which transmits 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 user ID of the RSCS virtual machine at your installation. You can find out the user ID of your installation's RSCS virtual machine and the location identifiers for the various remote stations from your installation's system programmer.

After you enter 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 enter 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. The NET is the user ID 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 that 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, enter 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.

Response

If the SPOOL command closes an output device, the informational message

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

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 IMSG OFF is in effect, this message is not issued, even if the preceding conditions are met.

Messages

DMKCQH013E	Conflicting option - *option*
DMKCQH024E	{CYLS|BLKS|Form number} missing or invalid
DMKCSO006E	Invalid device type - *devtype*
DMKCSO024E	{CYLS|BLKS|Form number} missing or invalid
DMKCSO040E	Device *rdev* does not exist
DMKCSP003E	Invalid option - *option*
DMKCSP007E	Invalid userid - *userid*
DMKCSP013E	Conflicting option - *option*
DMKCSP020E	Userid missing or invalid
DMKCSP024E	{CYLS|BLKS|Form number|DEST} missing or invalid
DMKCSP028E	Class missing or invalid
DMKCSP030E	Copies missing or invalid
DMKCSP032E	Distribution code missing or invalid
DMKCSP052E	Error in CP directory
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
DMKCSQ022E	Virtual device address missing or invalid
DMKCSQ024E	{CYLS|BLKS|Form number} missing or invalid
DMKCSQ028E	Class missing or invalid
DMKCSQ029E	Filename and/or filetype missing or invalid
DMKCSQ030E	Copies missing or invalid
DMKCSV013E	Conflicting option - *option*
DMKCSV024E	{CYLS|BLKS|Form number} missing or invalid
DMKCSV026E	Operand 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.

Format

```
STore   ┌ hexloc                                        ┐
        │ Lhexloc      hexword1     [hexword2 ... ]     │
        │                                               │
        │ Shexloc      hexdata...                       │
        │                                               │
        │ {Greg}                                        │
        < {Xreg}       hexword1     [hexword2 ... ]     >
        │                                               │
        │  {Yreg}      hexword1     [hexword2 ... ]     │
        │                                               │
        │ Psw          [hexword1 ]  hexword2            │
        │                                               │
        └ STATUS                                        ┘
```

Operands

hexloc
L*hexloc 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 L*hexloc*) 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 more than one *hexword* is specified, they must be separated by one or more blanks.

S*hexloc hexdata...*

stores the data specified (*hexdata...*) in the address specified by *hexloc*, without word alignment. The shortest string that can be stored is 1 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.

G*reg 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. In the *hexword1* [*hexword2...*] operand, each entry 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 you specify more than one *hexword*, they must be separated by one or more blanks.

X*reg 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. In the *hexword1* [*hexword2...*] operand, each entry 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 you specify more than one *hexword*, they must be separated by one or more blanks.

Y*reg hexword1* [*hexword2...*]

stores the hexadecimal data (*hexword1* [*hexword2...*]) 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. In the *hexword1* [*hexword2...*] operand, each entry 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 you specify more than one *hexword*, 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. In the [*hexword1*] *hexword2* operand, each entry 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 entered 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	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

Usage Notes

1. First-level storage is real main storage. You can store into only second-level storage (storage that is real to the virtual machine). 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 can 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 properly processed.
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 your virtual machine uses the STORE command 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 1 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 can then be displayed at the terminal with 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 - *operand*
DMKCDS005E	Invalid hexdata - *data*
DMKCDS010E	Invalid register - *register*
DMKCDS012E	Invalid PSW - *psw*
DMKCDS026E	Operand missing or invalid
DMKCDS033E	Hexloc missing or invalid
DMKCDS160E	Hexloc *hexloc* exceeds storage
DMKCDS162W	Invalid ECR *x* - *xxxxxxxx*
DMKCDS163E	Store exceeds maximum register
DMKCDS164E	Hexloc *hexloc* non-addressable storage
DMKCDS166E	Changed shared page not written to auxiliary storage
DMKCDS168E	Status operand requires ecmode
DMKCDS923I	Failing storage frame at location *rstor*

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.

Format

```
SYStem      ┌CLEAR  ┐
            │RESET  │
            └RESTART┘
```

Operands

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 CLEAR, RESET, and RESTART functions leave the virtual machine in a disabled wait state. Both CLEAR and RESET clear pending interrupts and load a PSW of all zeros with only the WAIT bit set to ON (X'0002000000000000').
2. When the virtual machine enters a disabled wait state, the RUN option of the CP SET command is automatically set to OFF. The virtual machine continues with RUN set to OFF until a valid program is loaded into storage and a CP SET RUN ON is processed.
3. Depending on the virtual machine's storage size, if you previously IPLed a named system, the contents of virtual storage are unpredictable after a CP SYSTEM RESET.

Responses

```
STORAGE CLEARED - SYSTEM RESET
```

is the response to the command SYSTEM CLEAR.

```
SYSTEM RESET
```

is the response to the command SYSTEM RESET.

If you enter the command SYSTEM RESTART, 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 descriptive information with a VM/SP spool file or output unit record device.

Format

```
┌─────────┬──────────────────────────────────────────────────┐
│         │                                                  │
│ TAg     │          ┌ Printer ┐                             │
│         │          │ PUnch   │                             │
│         │   DEv    │ CONsole │  [tagtext]                  │
│         │          └ vaddr   ┘                             │
│         │                                                  │
│         │   FIle     spoolid    [tagtext]                  │
│         │                                                  │
│         │                ┌          ┌ Printer ┐ ┐          │
│         │                │          │ PUnch   │ │          │
│         │                │  DEv     │ CONsole │ │          │
│         │   QUery        │          └ vaddr   ┘ │          │
│         │                │                      │          │
│         │                └  FIle      spoolid   ┘          │
│         │                                                  │
└─────────┴──────────────────────────────────────────────────┘
```

Operands

```
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, see "Usage Notes."

```
        ┌      ┌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 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 location identifiers of remote stations accessible to your virtual machine.

   userid

   is the user ID 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 with 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

```
               ┌PRINTER┐
TAG QUERY DEV  ┤PUNCH  ├
               └CONSOLE┘
```

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

```
┌PRT ┐
┤PUN ├ vaddr TAG:
└CONS┘
 tagtext..........
```

repeated for each device in the specified class

```
TAG QUERY DEV vaddr
```

Displays the contents of the tag associated with the specified device, as follows:

```
┌PRT ┐
┤PUN ├ vaddr TAG:
└CONS┘
 tagtext..........
```

if tag data exists

```
TAG NOT SET
```

if the TAG command was never issued to that device.

```
TAG QUERY FILE spoolid
```

Displays the contents of the tag associated with the specified file, as follows:

```
tagtext..........
```

if tag data exists

```
(TAG BLANK)
```

if the tag is all blanks

```
(TAG MISSING)
```

if the file did not contain a tag because it was either an input file from the real card reader or was an output file generated prior to VM/370 Release 2 PLC 11.

Messages

DMKCST003E	Invalid option - *option*
DMKCST006E	Invalid device type - *devtype*
DMKCST016E	TAG invalid for specified file type
DMKCST022E	Virtual device address missing or invalid
DMKCST026E	Operand missing or invalid
DMKCST027E	Spoolid missing or invalid
DMKCST040E	Device *vdev* does not exist
DMKCST042E	Spoolid *nnnn* does not exist
DMKCST424E	Reader 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
- 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
- Specifying ASCII translation tables used
- Specifying the insertion of line control characters.

Format

```
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     }
                         { NONE     }

           TYpe          { 3101     }
                         { TTY      }

           PROMpt        { VM       }
                         { TTY      }

           SCROll        { nnn      }
                         { CONT     }

           CNTL          { USR      }
                         { SYS      }

           ASCiitbl      { VM2      }
                         { VM1      }
```

[1] You can specify more than one function with a single entry of the TERMINAL command. For example,

```
terminal chardel off linesize 20 mode vm
```

Operands

CHardel $\left\{\begin{array}{l}\textbf{ON}\\ \textbf{OFf}\\ \textit{char}\end{array}\right\}$

defines the logical character delete symbol. The logical character delete symbol allows you to delete one or more of the previous characters entered. The default symbol is usually @, but this depends on what is specified in your VM/SP directory entry.

If ON is specified, the default symbol becomes the logical character delete symbol. 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 the system operator logs on your virtual machine with the AUTOLOG command, CHARDEL OFF is the default.

LINEDel $\left\{\begin{array}{l}\textbf{ON}\\ \textbf{OFf}\\ \textit{char}\end{array}\right\}$

defines the logical line delete symbol. The logical line delete symbol deletes the entire previous physical line, or the last logical line back to (and including) the previous line end. The default symbol is usually ¢, but this depends on what is specified in your VM/SP directory entry.

If ON is specified, the default symbol becomes the logical line delete symbol. 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 the system operator logs on your virtual machine with the AUTOLOG command, LINEDEL OFF is the default.

LINENd $\left\{\begin{array}{l}\textbf{ON}\\ \textbf{OFf}\\ \textit{char}\end{array}\right\}$

defines the logical line end symbol. The logical line end symbol allows you to key in more than one command on the same line, and thus minimizes the amount of time you have to wait between entering commands. The default symbol is usually #, but this depends on what is specified in your VM/SP directory entry.

If ON is specified, the default symbol becomes the logical line end symbol. 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 the system operator logs on your virtual machine with the AUTOLOG command, LINEND OFF is the default.

EScape $\left\{\begin{array}{l}\textbf{ON}\\ \textbf{OFf}\\ \textit{char}\end{array}\right\}$

defines the logical escape character. The logical escape character causes VM/SP to consider the next character entered to be a data character, even if it is usually one of the logical line editing symbols. The default symbol is generally ", but this depends on what is specified in your VM/SP directory entry.

If ON is specified, the default symbol becomes the logical escape character. 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 the system operator logs on your virtual machine with the AUTOLOG command, ESCAPE OFF is the default.

TABchar $\left\{\begin{array}{l}\textbf{ON}\\ \textbf{OFf}\\ \textit{char}\end{array}\right\}$

allows a user to define a variety of logical tab characters on graphics devices. QUERY TERMINAL command displays ON, OFF, or the current logical tab character.

ON sets the logical tab character to the one used by the system, X'6A' (¦).

OFF sets the logical tab character to X'00'.

The *char* defines the logical tab character the system uses internally. It can 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 $\left\{\begin{array}{l}\textbf{ON}\\ \textbf{OFf}\end{array}\right\}$

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 usual translation tables (that is, BCD or correspondence code). Unless otherwise specified, APL OFF is in effect. If APL is ON, TEXT is forced OFF.

Note: If the TERMINAL APL ON command is entered 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 entered, the results can be unpredictable.

If APL ON is specified, the LINESIZE value is overridden (see the explanation for the LINESIZE operand). However, the setting of the SET LINEDIT command is independent of the TERMINAL APL setting.

TEXT $\left\{\begin{matrix}\textbf{ON}\\\textbf{OFf}\end{matrix}\right\}$

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 usual translation tables. Unless otherwise specified, TEXT OFF is in effect. When TEXT ON is specified, APL OFF is forced.

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

ATtn $\left\{\begin{matrix}\textbf{ON}\\\textbf{OFf}\end{matrix}\right\}$

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 $\left\{\begin{matrix}\textbf{ON}\\\textbf{OFf}\end{matrix}\right\}$

highlights the command line that a user enters on a terminal. When the feature is deactivated, user input is displayed at regular intensity on the screen. When you log on VM/SP, HILIGHT is set OFF.

SCRNsave $\left\{\begin{matrix}\textbf{ON}\\\textbf{OFf}\end{matrix}\right\}$

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 valid only if you have a local 3270 Information Display device and have entered the CONMODE 3270 operand of the TERMINAL command. If you specify ON with this option, CP saves the screen and it is 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 $\left\{\begin{matrix}\textbf{CP}\\\textbf{VM}\end{matrix}\right\}$

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 Chapter 2.

LINESize $\left\{\begin{matrix} \textit{nnn} \\ \textbf{OFF} \end{matrix}\right\}$

The *nnn* specifies the maximum allowable line length for terminal output. It 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 $\left\{\begin{matrix} \textbf{3215} \\ \textbf{3270} \end{matrix}\right\}$

allows you to choose between 3215 and 3270 support at an SIO level. The initial definition is determined by the device type specified in the CONSOLE statement in your directory entry. You must have a local, remote, or VM/VTAM display device that has a 3270 compatible command set to use CONMODE 3270 support. 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 $\left\{\begin{matrix} \textbf{IMMED} \\ \textbf{GUESTctl} \end{matrix}\right\}$

allows you to control breakin by CP. This option is available to users with local or VM/VTAM 3270 Information Display devices. VTAM users need ACF/VTAM Version 3.1.2 (Product Number 5664-280) to use BREAKIN GUESTCTL; those with earlier releases who attempt to use BREAKIN GUESTCTL receive the error message 'INVALID DEVICE TYPE.'

When you use the GUESTCTL option while in full-screen mode, messages from CP are displayed when you press the BRKKEY key or when you enter a command requiring line mode I/O. If you specify GUESTCTL, CP breakin 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, breakin by CP takes place on any full-screen READ or full-screen WRITE that has an inactive keyboard; this also occurs when you hit the BRKKEY.

BRKkey $\left\{\begin{matrix} \textbf{PA1} \\ \textbf{PF1} \\ . \\ . \\ \textbf{PF24} \\ \textbf{NONE} \end{matrix}\right\}$

allows the PA1 key or a PF key you specify to control breakin by CP for full-screen guests. Only local graphics terminal users can specify a PF key. Users of remote and VM/VTAM 3270-type terminals can set the break key to either the PA1 key or to NONE. PA1 is the default for all graphics terminals. Use of the NONE option disables the break key and prevents you from entering CP mode. 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 breakin controller that does not exist on the guest virtual machine console hardware can lock out CP mode. Pressing PA1 still causes the CP environment to be entered when operating in line mode unless NONE is specified. Some applications (for example, the System Product Editor) also interpret PA1 as a request for CP mode. Once the break key has been pressed, the BEGIN command must be entered to return to the guest virtual machine console.

TYpe $\left\{\begin{matrix}\textbf{3101}\\ \textbf{TTY}\end{matrix}\right\}$

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 1 through 24
- Causes output to begin at column 1 of line 24 on the display.

PROMpt $\left\{\begin{matrix}\textbf{VM}\\ \textbf{TTY}\end{matrix}\right\}$

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 $\left\{\begin{matrix}nnn\\ \textbf{CONT}\end{matrix}\right\}$

The *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 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.

CNTL $\left\{\begin{matrix}\underline{\textbf{SYS}}\\ \textbf{USR}\end{matrix}\right\}$

allows you to specify whether you or CP handles line control characters in the data stream. The USR option prevents CP from inserting these characters. The SYS option indicates that CP inserts line control characters in the data stream. If you specify CNTL as USR and you are logged off, CNTL returns to SYS.

ASCiitbl $\left\{\begin{matrix}\underline{\textbf{VM1}}\\ \textbf{VM2}\end{matrix}\right\}$

allows you to specify ASCII to EBCDIC and EBCDIC to ASCII translate tables. If you select the VM1 option, translation tables based on STD TTY ANSI X3.4 1977 are used. If you select the VM2 option, translation tables based on STD TTY ANSI X3.26 1980 are used. If you specify ASCIITBL as VM2 and you are logged off, ASCIITBL returns to VM1.

Usage Notes

1. To determine the settings for your terminal, enter the QUERY TERMINAL command.

2. 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. TEXT, APL, ATTN, and LINESIZE are reset if you log on after disconnecting.

3. Although you can define line-editing symbols and status with the TERMINAL command, the LINEDIT operand of the SET command determines whether the VM/SP line-editing functions are ON or OFF.

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

5. Do not use any of the letters A through Z or numbers 0 through 9 as a symbol (*char*) in association with any of the CHARDEL, LINEDEL, LINEND, ESCAPE, and TABCHAR operands.

6. If you assign LINEDEL, CHARDEL, or ESCAPE as the same logical character, you lose the function of the logical character. If you assign LINEDEL, CHARDEL, or ESCAPE as the same logical character as LINEND, only LINEND is processed when you use that character.

7. The pound sign (#) in the #CP command is the logical line end symbol and is the default supplied by VM/SP. If you or your installation have redefined the logical line end symbol, #CP is an invalid command; you must substitute the redefined line end symbol for the pound sign when using this command.

8. The following table, 3270 High Intensity Features, describes the other features available for highlighting because the TERMINAL command cannot activate or deactivate these features.

Table 6. 3270 High Intensity Features

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

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

10. For the user to insert line control characters, the CMS WRTERM must have the EDIT option set to NO or LONG; otherwise, control characters are inserted. In

addition, the TERMINAL command options TYPE and PROMPT must be set to TTY. If TYPE and PROMPT are not both set to TTY, both are forced to TTY.

11. If CNTL is set to USR and you try to set PROMPT to VM or TYPE to 3101, the CONFLICTING OPTIONS message is displayed, and PROMPT and TYPE remain set to TTY. To change these settings, first set CNTL to SYS.
12. 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 accepts the MASK operand on the TERMINAL command as valid. However, there is no need to specify this operand because CP automatically masks your password.
13. 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.
14. 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.
15. If CONMODE 3270 is turned off and back on (by using the TERMINAL command, or by disconnecting and reconnecting and using the TERMINAL command), the virtual machine is presented an X'8E' status in the CSW for the console. An X'8E' indicates that the 3270 screen may have been reformatted. User console channel programs should allow for this if the CONMODE is switched. Otherwise, a RESET to the console is needed to clear all status after the CONMODE 3270 is turned on again.

Responses

None.

Messages

DMKCFT002E	Invalid operand - *operand*
DMKCFT006E	Invalid device type - *devtype*
DMKCFT013E	Conflicting option - *option*
DMKCFT026E	Operand missing or invalid
DMKCFT052E	Error in CP directory
DMKCFT053E	*userid* not in CP directory

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. The TRACE command can use operands in a cumulative manner; 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.

Format

```
TRace   ┌ ┌          ┐ 1  ┌ PRINter                     ┐┐
        │ │ SVC      │    │                             ││
        │ │ I/O      │    │  ┌ TERMinal ┐ ┌ NORun ┐     ││
        │ │ PROgram  │    │  └ BOTH     ┘ └ RUN   ┘     ││
        │ │ EXTernal │    │                             ││
        │ │ PRIV     │    │                             ││
        │{│ SIO      }    │                             ││
        { │ CCW      │    │   OFf                       │}
        │ │ SNS      │    └                             ┘│
        │ │ BRanch   │                                   │
        │ │ INSTruct │                                   │
        │ │ ALL      │                                   │
        │ └ CSW      ┘                                   │
        │   END                                          │
        └                                                ┘
```

[1] You can trace more than one of these activities with a single TRACE command. For example,

```
trace svc program sio printer
```

Operands

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 nonspooled. SIO and SIOF instructions are not 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.

SNS
traces sense data resulting from I/O errors. The data returns in the form:

```
SNS V 0vdev xxx...xxx
```

Where:

vdev	is the virtual address of the device from which the sense data was received, and
xxx...xxx	represents the sense data displayed as hex digits.

Up to 32 bytes (64 digits) can be displayed in groups of 4 bytes.

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
stops 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. This simulates the instruction step function.

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 finished 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 stopped 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 stopped with the TRACE END command, CP turns hardware assist on again.

3. The point at which execution stops depends on the type of instruction and the type of trace.

   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.

4. The NORUN option stops execution after each output message. To resume operation of the virtual machine, enter the BEGIN command.

5. RUN and NORUN can be specified in different TRACE commands with no conflict as long as the functions being traced are different.

6. 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.

7. When tracing I/O, you cannot detach devices with the DETACH command. Attempts to detach a device are rejected and the following message is issued:

```
DMKVDD273E  DETACH rejected; virtual machine tracing I/O
```

8. TRACE does not detect successful branches to the next sequential instruction and branch-to-self instructions.

9. Instructions that change or examine the first 2 bytes of the next sequential instruction cause erroneous processing for BRANCH and INSTRUCT tracing.

10. 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.

11. If a virtual machine using a protected shared segment is tracing BRANCH, INSTRUCT, or ALL activities, the user receives a warning message and the virtual machine is given a private copy of the named system. Other users of the segment continue to operate with an unchanged copy of that segment.

Also, be aware that using the BRANCH, INSTRUCT, or ALL operands within an unprotected shared segment changes the data in storage and can cause problems for other users who have accessed the segment. This problem could arise, for example, when several service machines have write access to a GCS common segment.

12. I/O operations for virtual channel-to-channel devices, with both ends connected to the same virtual machine, cannot be traced.
13. 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 lets you trace only the information that pertains to a particular problem.
14. 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.
15. The PA1 key on a 3270 display terminal should not be pressed with TRACE ALL RUN active while the display screen indicates RUNNING status because any subsequent CP command entered from the console can result in interleaved output.
16. Some modules in CMS do not issue SIOs. The I/O operation is initiated through 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, SEEK, REAL EXTENT, VIRT EXTENT, and 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 because it is already at the correct cylinder; there is no 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     tttttttt  tttttttt   (WAIT bit on)
::: vvvvvv LPSW  xxxxxxxx ==> tttttttt  tttttttt   (WAIT bit not on)
::: vvvvvv mnem  xxxxxxxx                          (all others)
```

Executed Instructions:

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

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 option - *option*
DMKTRA026E	Operand missing or invalid
DMKTRA180W	TRACE not in effect
DMKVDD273E	DETACH rejected; virtual machine tracing I/O

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.

Format

```
TRANsfer  ┌Printer┐  ┌spoolid    ┐  ┌┌[To]    ┌*     ┐┐  ┌Printer┐
          │PUnch  │  │CLass c    │  ││        └userid┘│  │PUnch  │
          │Reader │  ┤FORM form  ├  ┤                 ├  │Reader │
          └       ┘  │DEST dest  │  │ From    ┌ALL   ┐│  └       ┘
                     └ALL        ┘  └         └userid┘┘
```

Operands

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 through Z and from 0 through 9.

FORM *form*

specifies that you want to transfer all files that have the form name you specify. The *form* is a one- to eight-character form name defined at system generation.

DEST *dest*

specifies the destination name for the files that you want to transfer. The *dest* is a one- to eight-character alphameric name your installation assigns to printers and punches 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." The * denotes your own *userid.*

From $\left\{\begin{array}{l}\textit{userid}\\ \textbf{ALL}\end{array}\right\}$

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 Notes

1. The TRANSFER command does not transfer any active spool files.
2. The command transfers only 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:

	Spool Files	Queue		
		Reader	Printer	Punch
Origin	Real Reader	*		
	Printer	*	*	
	Console	*	*	
	Punch	*		*
	Dump/Monitor	*		

   CP flags, as an error, any attempt to transfer spool files from a print queue to a punch queue and any attempt to transfer spool files from a punch queue to a print queue. You can, 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.

3. Transferring a file to a device which is in system hold status places the file in system hold. In addition, when a file that is in system hold status is transferred, it remains in system hold status even if the device to which it is transferred is not in system hold.
4. When the target spool file has a conversion status of COMPLETE, a new spoolid is assigned to this file when it is transferred to another queue (other than the print queue).

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 did not 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.

`transfer pun dest bldgl to * prt`

transfers all the punch files with a destination of BLDG1 to your own print queue.

Responses

```
⎧RDR⎫ FILE  spoolid TRANSFERRED TO  userid ⎧RDR⎫
⎨PRT⎬                                       ⎨PRT⎬
⎩PUN⎭                                       ⎩PUN⎭
```

is the response when you enter 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 entering 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 enter the TRANSFER command. It is not displayed if you entered the CP SET IMSG OFF command line.

When the target spool file has CONVERSION status of COMPLETE, then a new spoolid is assigned to this file when it is transferred to another chain (other than the print chain).

```
SPOOLID spoolid1 HAS BEEN CHANGED TO spoolid2
```

is the response you receive when you enter the TRANSFER command and the file has the conversion status of COMPLETE.

For example, you can receive the response

```
SPOOLID 0003 HAS BEEN CHANGED TO 0008
SPOOLID 0004 HAS BEEN CHANGED TO 0009
```

CP does not display any of these responses if you entered the CP SET IMSG OFF command.

Messages

DMKCSX003E	Invalid option - *option*
DMKCSX006E	Invalid device type - *devtype*
DMKCSX007E	Invalid userid - *userid*
DMKCSX008E	Invalid spoolid -*spoolid*
DMKCSX013E	Conflicting option - *option*
DMKCSX020E	Userid missing or invalid
DMKCSX024E	{FORM number|DEST} missing or invalid
DMKCSX026E	Operand missing or invalid
DMKCSX027E	Spoolid missing or invalid
DMKCSX028E	Class missing or invalid
DMKCSX035E	Device type missing or invalid
DMKCSX042E	Spoolid *nnnn* does not exist
DMKCSX053E	*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 and 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 VM/SP IPCS, see the *VM/SP Interactive Problem Control System Guide and Reference.*

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.

Format

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

         ┌TO *      ┐
         │TO userid │
         └SYSTEM    ┘

         [FORMAT vmtype]

         [DSS]

         [*dumpid]
```

Operands

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 dumps 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 dumps 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* that is greater than *hexloc1* value. The *hexloc2* value must not exceed the limit of virtual storage.

{.} *bytecount*
<u>END</u>
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 dumps 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 byte count. When you enter

```
vmdump 1000.20
```

CP dumps 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*
<u>TO *</u>
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*) that VM/IPCS uses to format the dump. The *vmtype* keyword is a one- to eight-byte name. CP also uses the specified *vmtype* as the virtual card reader file type. CP does not check the *vmtype* for validity. 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 not specify FORMAT, the default *vmtype* is FILE. Be careful to specify the *vmtype* correctly; if a *vmtype* is not specified or is incorrect or invalid, VM/IPCS may not be able to perform advanced functions such as appending load maps, locating dump analysis routines, or locating component-related IPCSSCAN subcommands. See the *VM/SP Interactive Problem Control System Guide and Reference* for further information on IPCS. See Usage Note 4.

DSS
specifies that CP dump all discontiguous saved segments in use by your virtual machine. When DSS is specified, only the discontiguous save segments are dumped unless you explicitly specify other locations. An error message is 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 that 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 is not 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 usual 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.

Response

```
COMMAND COMPLETE
```

is the response indicating successful 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 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: {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.

Chapter 5. CP Command Summary

This section lists the following:

- The CP class G and class Any commands in alphabetical order
- The corresponding function type
- The IBM-defined privilege classes that can execute the command
- A partially blank column in which you can record changes your installation makes to the class of a command.
- 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.
ADSTOP	<G>	G		Halt execution at a specific virtual machine instruction address.
ATTN	<G>	G		Make an attention interruption pending for the virtual machine console.
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	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.
DEFINE	G	G		Reconfigure your virtual machine.
DETACH	G	G		Detach a virtual device from a virtual machine. Detach a channel from your virtual machine.

Command	Function Type	IBM-Defined Privilege Class	User-Defined Class	Usage
DIAL	N/A	Any	None	Connect a terminal or display device to the virtual machine's virtual communication line.
DISCONN	N/A	Any	None	Disconnect your terminal from your virtual machine.
DISPLAY	<G>	G		Display virtual storage on your terminal.
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.
EXTERNAL	<G>	G		Simulate an external interruption for a virtual machine and return control to that machine.
INDICATE	G	G		Indicate resource utilization and contention.
IPL	<G>	G		Simulate IPL for a virtual machine.
LINK	<G>	G		Provide access to a specific DASD by a virtual machine.
LOADVFCB	<G>	G		Load virtual forms control buffer for a virtual 3203, 3262, 3289E, 3211, 4245, or 4248 printer.
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.
NOTREADY	<G>	G		Simulate NOT READY for a device to a virtual machine.
ORDER	G	G		Rearrange closed spool files in a specific order.
PER	<G>	A, B, C, D, E, F, G		Monitor certain events in the user's virtual machine as they occur during program execution.
PURGE	G	G		Remove closed spool file from the system.
QUERY	G	G		Request information about machine configuration and system status.

Command	Function Type	IBM-Defined Privilege Class	User-Defined Class	Usage
READY	<G>	G		Simulate device end interruption for a virtual device.
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.
SCREEN	<G>	G		Change the color and extended highlighting values of the virtual machine.
SEND	<G>	G		Pass commands and message replies to disconnected virtual machine for processing. Pass message text to logical printer.
SET	G	G		Control various functions within the virtual machine.
SLEEP	N/A	Any		Place the virtual machine in a dormant state but allow messages to be displayed.
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 through the RSCS virtual machine.
STORE	<G>	G		Alter specified virtual storage locations and registers.
SYSTEM	<G>	G		Simulate RESET, CLEAR STORAGE, and RESTART buttons on a real system console.
TAG	<G>	G		Specify variable information to be associated with a spool file or output unit record device. Interrogate the current TAG text setting of a given spool file or output unit record device.
TERMINAL	<G>	G		Define or redefine the input and attention handling characteristics of your virtual console.

Command	Function Type	IBM-Defined Privilege Class	User-Defined Class	Usage
TRACE	<G>	G		Trace specified virtual machine activity at your terminal, spooled printer, or both.
TRANSFER	G	G		Transfer input files or reclaim input files from a specified user's virtual card reader.
VMDUMP	<G>	G		Dump virtual machine (use VM/SP IPCS to view dump.)

Appendix A. Logical Line Editing Symbols

To aid you in entering command or data lines from your line mode or display terminal, VM/SP provides a set of logical line editing symbols, which you can use to correct mistakes as you enter lines. Each symbol has been assigned a default character value. These normally are:

Symbol	Character
Logical character delete	@
Logical line end	#
Logical line delete	¢
Logical escape	"

Logical Character Delete

The logical character delete symbol (@) allows you to delete one or more of the previous characters entered. The @ deletes one character per @ entered, including the ¢ and # logical editing characters. For example,

ABC#@@ results in AB

ABC@D results in ABD

¢@DEF results in DEF

ABC@@@ deletes the entire string

Logical Line End

The logical line end symbol (#) allows you to key in more than one command on the same line, and thus minimizes the amount of time you have to wait between entering commands. You type the # at the end of each logical command line, and follow it with the next logical command line. VM/SP stacks the commands and executes them in sequence. For example, the entry

```
query users#query time#query dasd
```

is executed in the same way as the entries

```
query users
query time
query dasd
```

The logical line end symbol also has special significance for the #CP function. Beginning any physical line with #CP indicates that you are entering a command that is to be processed by CP immediately. If you have set a character other than # as your logical line end symbol, you should use that character instead of a #.

Logical Line Delete

The logical line delete symbol (¢) deletes the entire previous physical line, or the last logical line back to (and including) the previous logical line end (#). You can use it to cancel a line containing many or serious errors. If a # immediately precedes the ¢ sign, only the # sign is deleted, since the # indicates the beginning of a new line, and the ¢ cancels the current line. For example:

- Logical Line Delete:

  ABC#DEF¢ deletes the #DEF and results in ABC

  ABC#¢ results in ABC

  ABC#DEF¢#GHI results in ABC#GHI

  ABC#DEF¢GHI results in ABCGHI

- Physical Line Delete:

  `ABC¢` deletes the whole line

Note: When you cancel a line by using the ¢ logical line delete symbol, you do not need to press a carriage return; you can continue entering data on the same line.

Logical Escape

The logical escape symbol (") causes VM/SP to consider the next character entered to be a data character, even if it is normally one of the logical line editing symbols (@, ¢, ", or #). For example,

`ABC"¢D` results in ABC¢D
`""ABC""` results in "ABC"

If you enter a single logical escape symbol (") as the last character on a line, or on a line by itself, it is ignored.

When you enter logical escape characters in conjunction with other logical editing characters, the results may be difficult to predict. For example, the lines

```
ABC""@DEF
ABC""@@DEF
```

both result in the line ABCDEF.

Defining Logical Line Editing Symbols

The logical line editing symbols are defined for each virtual machine during VM/SP system generation. If your terminal's keyboard lacks any of these special characters, your installation can define other special characters for logical line editing. To find out the logical line editing symbols in effect for your virtual machine, enter the command

```
cp query terminal
```

The response might be

```
LINEND # , LINEDEL ¢ , CHARDEL @ , ESCAPE "
LINESIZE 130, MASK OFF, APL OFF, ATTN OFF, MODE VM
```

You can use the TERMINAL command to change the logical line editing characters for your virtual machine. For example, if you enter

```
terminal linend /
```

then, the line

```
input # line / input / #
```

is interpreted

```
input # line
input
#
```

The terminal characteristics listed in the response to the QUERY TERMINAL command are all controlled by operands of the TERMINAL command.

Summary of Changes

Structural Changes

This book contains all class G and class Any CP commands. Commands for classes A, B, C, D, E, and F, previously in the *VM/SP CP Command Reference*, are now contained in *VM/SP CP System Command Reference*.

Technical Changes

Summary of Changes
for SC19-6211-5
for VM/SP Release 6

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. The pseudo-numbers are contained in the *VM/SP Library Guide and Master Index*, GC19-6207.

SET Command

BACKWARD and FORWARD options are added to the SET PFnn RETRIEVE subcommand. BACKWARD causes the PF key to retrieve input lines from the most current to oldest line stored. FORWARD causes the PF key to retrieve input from the oldest line saved to the most current line.

TRACE Command

The optional parameter SNS traces sense data resulting from I/O errors.

Saved Segment Management

Minor technical and editorial changes resulting from Saved System Management have also been made throughout this publication.

Shared File System

The parameter FILEPOOL is added to the IPL command. FILEPOOL, with *filepoolid* specifies the default file pool to be established as the user's A-disk.

Minor technical and editorial changes resulting from SFS have also been made throughout this publication.

MISCELLANEOUS

Information about special processing for spool files has been added to "Chapter 2. CP Command Usage."

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

Summary of Changes for SC19-6211-4 for VM/SP Release 5

IPL Command

Two optional CMS parameters are added: NOSPROF and INSTSEG. The NOSPROF parameter allows you to bypass the SYSPROF EXEC when you IPL CMS. The INSTSEG parameter links an optional Installation Discontiguous Shared Segment (DCSS) that contains the executing copy of frequently used execs and editor macros.

Destination Option for Spool Files

The DEST option allows you to select a specific printer or punch to process your print, punch, or console file.

National Language Support in CP

The CPLANG option of the QUERY command allows you to display the identifier of the language in your virtual machine which is set to issue CP messages.

Transparent Services Access Facility Support

The CPTRAP command allows the Transparent Services Access Facility (TSAF) virtual machine to send data to add to the CPTRAP file.

The QUERY CPTRAP subcommand provides you with information about the CPTRAP function.

Alternate Nucleus Support

An option is added to the SHUTDOWN command to allow you to IPL an alternate nucleus residing on a CP-owned volume.

Terminal Enhancements

VM/VTAM and remote users can issue the TERMINAL CONMODE 3270 command. Support for TERMINAL CONMODE 3270 is expanded so that virtual machine start I/Os are handled as 3270 start I/Os. The console is placed in full-screen mode with an application program controlling information sent to the terminal.

Those users that primarily use interactive applications can be protected from accidentally entering CP by using the SET CONCEAL command and the TERMINAL BRKKEY command.

Enhanced ASCII Support

ASCII support is enhanced to allow you to:

- Specify ASCII translation tables used with the using the ASCIITBL option of the TERMINAL command.
- Specify the insertion of line control characters with the CNTL option of the TERMINAL command.

DETACH Enhancements for Class B

UNLOAD and LEAVE options are added to DETACH for a tape device.

UNLOAD detaches the tape device and automatically rewinds and unloads it.

LEAVE detaches the tape device without rewinding and unloading it. The tape remains positioned as it was before the DETACH command was issued.

IX/370 Support

SVCACCL option has been added to the SET command to accelerate SVC 9, 10, and 11.

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

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.

Glossary of Terms and Abbreviations

A

abend. (1) Abnormal end of task. (2) Synonym for *abnormal termination.*

abnormal termination. The ending of processing before planned termination. Synonymous with *abend.*

access mode. A method VM/SP uses to control user access to data files. Access modes let the user read and write data to a file, or only read data from a file.

advanced function printer (AFP). An all-points-addressable printer, such as the IBM 3800-3 and IBM 3820 printers, capable of printing images as well as text.

Advanced Program-to-Program Communications (APPC). The inter-program communication service within SNA LU 6.2 on which the APPC/VM interface is based.

Advanced Program-to-Program Communications/VM (APPC/VM). An API for communicating between two virtual machines that is mappable to the SNA LU 6.2 APPC interface and based on IUCV functions. Along with the TSAF virtual machine, APPC/VM provides this communication within a single system and throughout a collection of systems.

AFP. Advanced function printer.

AP. Attached processor.

APPC. Advanced Program-to-Program Communications.

APPC/VM. Advanced Program-to-Program Communications/VM.

attached processor (AP). A processor that has no I/O capability and is always linked to the processor initialized for I/O handling.

B

basic control (BC) mode. A mode in which additional System/370 features, such as new machine instructions, are not operational. Contrast with *extended control (EC) mode.*

BC mode. Basic control mode.

C

CAW. Channel address word.

CC. Condition code.

CCW. Channel command word.

channel address word (CAW). An area in storage that specifies the location in main storage at which a channel program begins.

channel command word (CCW). A doubleword at the location in main storage specified by the channel address word. One or more CCWs make up the channel program that directs data channel operations.

channel status word (CSW). An area in storage that provides information about the termination of I/O.

channel-to-channel adapter (CTCA). A hardware device that connects two channels on the same computing system or on different systems.

channel-to-channel (CTC) device. A hardware device that connects two channels on the same computing system or on different systems. CTC devices include both CTCAs and 3088 MCUs.

clock comparator. A hardware feature (required by VM/SP) that causes an interruption when the TOD clock has equaled or exceeded the value specified by a program or virtual machine.

CMS. Conversational Monitor System.

condition code (CC). A code that reflects the result of a previous I/O, arithmetic, or logical operation.

Control Program (CP). A component of VM/SP that manages the resources of a single computer so multiple computing systems appear to exist. Each virtual machine is the functional equivalent of an IBM System/370.

Conversational Monitor System (CMS). A virtual machine operating system and component of VM/SP that provides general interactive time sharing, problem solving, program development capabilities, and operates only under the control of the VM Control Program (CP).

copy function. The function initiated by a PF key to copy the contents of a display screen onto an associated hardcopy printer. A remote display terminal copies the entire contents of the screen onto a printer attached to the same control unit. A local display terminal .copies all information from the screen, except the screen status information, onto any printer attached to any local display control unit.

count-key-data (CKD) device. A disk storage device that stores data in the format: count field, usually followed by a key field, followed by the actual data of a record. The count field contains the cylinder number, head number, record number, and the length of the data. The key field contains the record's key (search argument).

CP assist. A hardware function, available only on a processor with ECPS, that reduces CP overhead by doing the most frequently used tasks of CP routines.

CP directory. Synonym for *VM/SP directory.*

CP privilege class. See *privilege class.*

CP read. The condition when CP is waiting for a response or request for work from the user. On a typewriter terminal, the keyboard is unlocked; on a display terminal, the screen status area indicates CP READ.

CPU timer. A hardware feature that measures elapsed processor time and causes an interruption when a previously specified amount of time has elapsed. The CPU timer is decremented when the processor is executing instructions, is in a WAIT state, and is executing program loading instructions, but not when the processor is in a stopped state. A virtual machine that uses the CPU timer must have the EC mode and REALTIMER options active.

CSW. Channel status word.

CTC. Channel-to-channel.

CTCA. Channel-to-channel adapter.

D

DIAGNOSE interface. A programming mechanism that lets any virtual machine, including CMS, directly communicate with CP by way of the DIAGNOSE instruction. Specific interface codes let a virtual machine more efficiently request specific CP services.

directory. See *VM/SP directory.*

disconnect mode. The mode of operation in which a virtual machine is executing without a physical line or terminal connected as an operator console. Any attempt to issue a read to the console causes the virtual machine to be logged off after 15 minutes have elapsed, unless the user logs on again within the 15-minute interval. Note that with the SCIF, a user can be disconnected from a primary virtual console but still have console communications through the console of the secondary user.

discontiguous saved segment. One or more 64K segments of storage that were previously loaded, saved, and assigned a unique name. The segment(s) can be shared among virtual machines if the segment(s) contains reentrant code. Discontiguous segments used with CMS must be loaded into storage at locations above the address space of a user's CMS virtual machine. They can be detached when no longer needed.

distribution code. In the VM/SP directory, a one-to-eight character identification word printed or punched with the user ID in the separator page (or punched card) to further identify the location or department of the user.

E

EC mode. Extended control mode.

ECPS:VM/370. Extended Control Program Support:VM/370.

eligible list. One of two queues of virtual machines waiting to get into the run list. They are runnable but cannot fit into the run list because of the current system load.

emulation program (EP). A control program that lets an IBM 3704 or 3705 Communications Controller emulate the functions of an IBM 2701 Data Adapter Unit, an IBM 2702 Transmission Control Unit, or an IBM 2703 Transmission Control Unit.

EP. Emulation program.

expanded virtual machine assist. A hardware assist function, available only on a processor that has ECPS, that handles many privileged instructions not handled by VMA, and extends the level of support of certain privileged instructions beyond that provided by VMA.

extended control (EC) mode. A mode in which all features of a System/370 computing system, including dynamic address translation, are operational. Contrast with *basic control (BC) mode.*

Extended Control Program Support (ECPS:VM/370). A hardware assist feature that improves the performance of CP by reducing CP overhead. ECPS:VM/370 consists of CP assist, expanded virtual machine assist, and virtual interval timer assist.

F

FB-512. An FBA device that stores data in 512-byte blocks (refers to the IBM 9335, 9332, 9313, 3370, and 3310 DASDs).

FCB. (1) Forms control buffer. (2) Function control block.

file access mode. A file mode number that designates whether the file can be used as a read-only or read/write file by a user.

file pool. A collection of minidisks managed by SFS. It contains user files and directories and associated control information. Many users' files and directories can be contained in a single file pool.

file tag. A character string associated with a VM/SP spool file that contains information pertaining to that spool file.

first-level storage. Refers to real main storage. Contrast with *second-level storage* and *third-level storage.*

forms control buffer (FCB). In the 3800 Printing Subsystem, a buffer for controlling the vertical format of printed output. The FCB is analogous to the punched-paper, carriage-control tape that IBM 1403 Printers use.

free storage. Storage not allocated. The blocks of memory available for temporary use by programs or by the system.

function control block (FCB). In Subsystem Support Services (SSS), a control block that contains information such as a function's status, event control block, task I/O queue, and I/O queue.

G

GCS. Group Control System.

GPR. General-purpose register.

Group Control System (GCS). A component of VM/SP, consisting of a shared segment that the user can IPL and run in a virtual machine. It provides simulated MVS services and unique supervisor services to help support a native SNA network.

guest virtual machine (GVM). A virtual machine in which an operating machine is running.

H

handshaking feature. See *VM/VS handshaking feature.*

high-water mark. The highest contiguous address, starting from location zero, where the virtual system's real addresses equal the virtual system's virtual addresses.

host system. A data processing system that prepares programs and the operating environments for use by another computer or controller.

I

initial program load (IPL). The initialization procedure that causes an operating system to begin operation. A VM user must IPL the specific operating system into the virtual machine that will control the user's work. Each virtual machine can be loaded with a different operating system.

in-queue virtual machine. A virtual machine on the run list waiting to be dispatched.

Interactive Problem Control System (IPCS). A component of VM/SP that permits online problem management, interactive problem diagnosis, online debugging for disk related CP or virtual machine abend dumps or CPTRAP files, problem tracking, and problem reporting.

inter-user communication vehicle (IUCV). A VM/SP generalized CP interface that helps the transfer of messages either among virtual machines or between CP and a virtual machine.

IPCS. Interactive Problem Control System.

IPL. Initial program load.

IPL processor. In an AP or MP system, the processor on which the control program was first initialized during system generation. Note that both the IPL and the non-IPL processors in a real MP configuration have I/O capabilities.

IUCV. Inter-user communication vehicle.

L

local. Two entities (for example, a user and a server) are said to be local to each other if they belong to the same system within a collection or to the same node within an SNA system. Contrast with *remote.*

logically connected terminal. A terminal connected by leased or switched telecommunications or by local attachment to a multiple-access virtual machine by the CP DIAL command. See *multiple-access virtual machine.*

logical saved segment. A portion of a physical saved segment that CMS can manipulate. Each logical segment can contain different types of program objects, such as modules, text files, execs, callable services libraries, language repositories, user-defined objects, or a single minidisk directory. A system segment identification file (SYSTEM SEGID) associates a logical saved segment to the physical saved segment in which it resides. See *physical saved segment* and *saved segment.*

logical unit (LU). An entity addressable within an SNA-defined network, similar to a node within a VM network. LUs are categorized by the types of communication they support. A TSAF collection in an SNA network is viewed as one or more LUs.

logical unit name (LU name). A symbolic name given to a particular LU in an SNA-defined network.

LU. Logical unit.

M

MDISK. (1) Another name for minidisk. (2) The user directory that describes a user's storage space.

MIH. Missing interrupt handler.

minidisk. A logical subdivision (or all) of a physical disk pack that has its own virtual device address, consecutive virtual cylinders (starting with virtual cylinder 0), and a VTOC or disk label identifier. Each user virtual disk is preallocated and defined by a VM/SP directory entry as belonging to a user.

minidisk directory. Synonym for *CMS file directory.*

missing interrupt handler (MIH). A VM/SP facility that detects incomplete I/O conditions by monitoring I/O activity. It also tries to correct incomplete I/O conditions without operator intervention.

MP. Multiprocessor.

multiple-access virtual machine. A virtual machine running under VM/SP that supports teleprocessing terminals.

Multiple Virtual Storage (MVS). An alternative name for OS/VS2.

multiprocessor (MP). A computer using two or more processing units under integrated control.

MVS. Multiple Virtual Storage.

N

named system. A system that has an entry in the CP system name table (DMKSNTBL). The entry in the system name table includes the system name and other pertinent data so that the system can later be saved. See *saved system.*

native mode. Refers to running an operating system stand-alone on the real machine instead of under VM/SP.

NCP. Network control program.

network control program (NCP). An IBM licensed program that provides communication controller support for single-domain, multiple-domain, and interconnected network capability.

non-IPL processor. In an AP or MP system, the attached or second processor initialized at system generation time. Note that both the IPL processor and the non-IPL processor in a real MP configuration have I/O capabilities.

null line. A logical line with a length of zero that usually signals the CMS Editor to end input mode and enter edit mode. In VM/SP, a null line for typewriter terminals is a terminal input line consisting of a return character as the first and only information, or a logical line end symbol as the last character in the data line. For display devices, a null line is indicated by the cursor positioned at the beginning of the user input area or the data in the user input area ending with a logical line end symbol.

O

Operating System/Virtual Storage (OS/VS). A family of operating systems that control IBM System/360 and System/370 computing systems. OS/VS includes VS1, VS2, MVS/370, and MVS/XA.

OS/VS. Operating System/Virtual Storage.

OS/VS1. A virtual storage operating system that is an extension of OS/MFT.

OS/VS2. A virtual storage operating system that is an extension of OS/MVT.

overhead. The additional processor time charged to each virtual machine for the CP functions needed to simulate the virtual machine environment and for paging and scheduling time.

P

page. A fixed-length block that has a virtual address and can be transferred between real storage and auxiliary storage.

paging. Transferring pages between real storage and external page storage.

paging area. An area of direct access storage (and an associated area of real storage) that CP uses for the temporary storage of pages when paging occurs.

PF key. Program function key.

physical saved segment. One or more pages of storage that have been named and retained on a CP-owned volume (DASD). Once created, it can be loaded within a virtual machine's address space or outside a virtual machine's address space. Multiple users can load the same copy. A physical saved segment can contain one or more logical saved segments. A system segment identification file (SYSTEM SEGID) associates a physical saved segment to its logical saved segments. See *logical saved segment* and *saved segment.*

prefix storage area (PSA). A page zero of real storage that contains machine-used data areas and CP global data.

privilege class. One or more classes assigned to a virtual machine user in a VM/SP directory entry; each privilege class specified lets a user access a logical subset of the CP commands. There are eight IBM-defined privilege classes that correspond to specific administrative functions. They are:

Class A	- primary system operator
Class B	- system resource operator
Class C	- system programmer
Class D	- spooling operator
Class E	- system analyst
Class F	- service representative
Class G	- general user
Class H	- reserved for IBM use
Class Any	- available to any user.

The privilege classes can be changed to meet the needs of an installation.

PROFILE EXEC. A special EXEC procedure with a file name of PROFILE that a user can create. The procedure is usually executed immediately after CMS is loaded into a virtual machine (also known as IPL CMS).

program function (PF) key. On a terminal, a key that can do various functions selected by the user or determined by an application program.

programmable operator facility. This facility enables automatic filtering and routing of messages from a specified virtual machine (for example the operator) to a logical operator virtual machine in a local distributed or mixed environment. It also permits installation defined actions to be automatically performed.

program status word (PSW). An area in storage that indicates the order in which instructions are executed, and to hold and indicate the status of the computer system.

PSA. Prefix storage area.

pseudo page fault. A facility available with VM/VS handshaking that lets the VS1 virtual machine dispatch another task while waiting for a page-in request to be completed for some other task. Without this facility, the whole virtual machine would wait until the page request was satisfied, even if higher priority tasks were ready to execute.

PSW. Program status word.

R

raddr. The real device address of an I/O device.

read-only access. An access mode associated with a virtual disk or SFS directory that lets a user read, but not write or update, any file on the disk or SFS directory.

read/write access. An access mode associated with a virtual disk or SFS directory that lets a user read and write any file on the disk or SFS directory.

real address. The address of a location in real storage or the address of a real I/O device.

real machine. The actual processor, channels, storage, and I/O devices required for VM/SP operation.

remote. Two entities (for example, a user and a server) are said to be remote to each other if they belong to different systems within a collection, or to different nodes within an SNA network. Contrast with *local.*

Remote Spooling Communications Subsystem Networking (RSCS). An IBM licensed program and special-purpose subsystem that supports the reception and transmission of messages, files, commands, and jobs over a computer network.

RSCS. Remote Spooling Communications Subsystem Networking.

S

saved segment. A segment of storage that has been saved and assigned a name. The saved segment(s) can be physical saved segment(s) that CP recognizes or logical saved segments that CMS recognizes. The segments can be loaded and shared among virtual machines, which helps use real storage more efficiently, or a private, nonshared copy can be loaded into a virtual machine. See *logical saved segment* and *physical saved segment.*

saved system. A special nonrelocatable copy of a virtual machine's virtual storage and associated registers kept on a CP-owned disk and loaded by name instead of by I/O device address. Loading a saved system by name substantially reduces the time it takes to IPL the system in a virtual machine. In addition, a saved system such as CMS can also share one or more 64K segments of reenterable code in real storage between virtual machines. This reduces the cumulative real main storage requirements and paging demands of such virtual machines.

SCP. System control programming.

secondary user. When a user is disconnected — that is, has no virtual console on line — a secondary user can be designated to receive the disconnected user's console messages and to enter commands to the disconnected user's console.

second-level storage. The storage that appears to be real to a virtual machine. Contrast with *first-level storage* and *third-level storage.*

SFS. Shared file system.

shared file system (SFS). A part of CMS that lets users organize their files into groups known as *directories* and to selectively share those files and directories with other users.

shared segment. A feature of a saved system or physical saved segment that lets one or more segments of reentrant code or data in real storage be shared among many virtual machines. For example, if a saved CMS system was generated, the CMS nucleus is shared in real storage among all CMS virtual machines loaded by name; that is, every CMS machine's segment of virtual storage maps to the same 64K of real storage. See *discontiguous saved segment* and *saved system.*

shared system. See *saved system.*

simultaneous peripheral operations online (SPOOL). (1) (Noun) An area of auxiliary storage defined to temporarily hold data during its transfer between peripheral equipment and the processor. (2) (Verb) To use auxiliary storage as a buffer storage to reduce processing delays when transferring data between peripheral equipment and the processing storage of a computer.

single console image facility (SCIF). (1) Lets a user, who is disconnected from a primary virtual console, continue to have console communications by way of the console of the secondary user. See *secondary user.* (2) Enables a virtual machine operator to control multiple virtual machines from one physical terminal.

single processor mode. In tightly coupled MP or AP systems, single processor mode lets an installation dedicate a processor to an MVS V=R virtual machine. In single processor mode, VM/SP runs in uniprocessor mode in the main processor, and the MVS V=R virtual machine runs under VM/SP in the main processor and has the exclusive use of the other processor for MP or AP operations. However, other virtual machines can operate under VM/SP concurrently with the MVS V=R virtual machine in single processor mode (not to be confused with *uniprocessor mode*).

SIO. Start I/O.

SNA. Systems Network Architecture.

SPOOL. Simultaneous peripheral operations online.

spool file class. A one-character class associated with each virtual unit record device. For input spool files, the spool file class lets the user control which input spool files are read next; and, for output spool files, it lets the spooling operator better control or reorder the printing or punching of spool files having similar characteristics or priorities. The spool file class value can be A through Z or 0 through 9.

spool file tag. In VM/SP, a 136-character data field associated with each output spool file generated. The use, content, and format of this field is a bilateral decision between the originator and receiver of the file.

spool ID. A spool file identification number automatically assigned by CP when the file is closed. The spool ID number can be from 0001 to 9900; it is unique for each spool file. To identify a given spool file, a user must specify the owner's user ID, the virtual device type, and the spool ID.

spooling. The processing of files created by or intended for virtual readers, punches, and printers. The spool files can be sent from one virtual device to another, from one virtual machine to another, and to real devices. See *virtual console spooling.*

spooling area. Any direct access storage area that VM/SP temporarily uses to store input for a virtual card reader, output for a virtual printer or punch, or a console log of I/O for a virtual console.

spooling devices. I/O devices (card readers, punches, printers, DASD) that read input and write output.

storage key. An indicator associated with one or more storage blocks that requires that tasks have a matching protection key in order to use the blocks.

supervisor call instruction (SVC). An instruction that interrupts a program being executed and passes control to the supervisor so that it can do a specific service indicated by the instruction.

SVC. Supervisor call instruction.

system load. The combination of active devices, programs, and users that use the system resources of the processor and storage.

System Product Editor. The CMS facility, comprising the XEDIT command and XEDIT subcommands and macros, that lets a user create, change, and manipulate CMS files.

system profile. An EXEC (SYSPROF) that resides in a saved system or on a system disk and called by CMS initialization. It contains some initialization functions, and provides a means for installations to override the default CMS environment by tailoring the exec to suit the installation.

system services control point (SSCP). In SNA, the focal point within an SNA network for managing the configuration, coordinating network operator and problem determination requests, and providing directory support and other session services for end users of the network.

Systems Network Architecture (SNA). The description of the logical structure, formats, protocols, and operational sequences for transmitting information units through and controlling the configuration and operation of networks.

T

T-disk. Synonym for *temporary disk*.

temporary disk. An area on a DASD available to the user for newly created or stored files until logoff, at which time the area is released. Temporary disk space is allocated to the user during logon or when entering the CP DEFINE command. Synonymous with *T-disk*.

third-level storage. The virtual storage created and controlled by an OS/VS or VM virtual machine. Contrast with *first-level storage* and *second-level storage*.

time-of-day (TOD) clock. A hardware feature required by VM/SP. The TOD clock is incremented once every microsecond, and provides a consistent measure of elapsed time suitable for the indication of date and time; it runs regardless of the processor state (running, wait, or stopped).

TOD clock. Time-of-day clock.

top directory. The directory created for a user when the user is enrolled in a file pool. The name of the top directory is the same as the person's user ID.

translate mode. The operating mode of a virtual machine when virtual addresses are converted to real addresses by segment and page tables.

Transparent Services Access Facility (TSAF). A component of VM/SP that handles communication between systems by letting APPC/VM paths span multiple VM systems. TSAF lets a source program connect to a target program by specifying a name that the target has made known, instead of specifying a user ID and node ID.

TSAF. Transparent Services Access Facility.

TSAF virtual machine. The virtual machine that lets user programs connect to and communicate with virtual machines on different VM systems.

U

uniprocessor mode. This term indicates that there is only one processor in the physical configuration, or that VM/SP uses the facilities of one processor in an AP or MP system (not to be confused with *single processor mode*).

V

vaddr. Virtual address.

virtual address. The address of a location in virtual storage. A virtual address must be translated into a real address in order to process the data in processor storage.

virtual card reader. CP's simulation on disk of a real card reader. A virtual card reader can read card, punch, or print records of up to 151 characters in length. The virtual device type and I/O device address are usually defined in the VM/SP directory. See *spool file class*.

virtual console. A console simulated by CP on a terminal such as a 3270. The virtual device type and I/O address are defined in the VM/SP directory entry for that virtual machine.

virtual console spooling. The writing of console I/O on disk as a printer spool file instead of, or in addition to, having it typed or displayed at the virtual machine console. The console data includes messages, responses, commands, and data from or to CP and the virtual machine operating system. The user can invoke or terminate console spooling at anytime. When the console spool file is closed, it becomes a printer spool file.

virtual CPU time. The time required to execute the instructions of the virtual machine.

virtual disk. A logical subdivision (or all) of a physical disk storage device that has its own address, consecutive storage space for data, and an index or description of the stored data so that the data can be accessed. A virtual disk is also called a minidisk.

virtual interval timer assist. A hardware assist function, available only on a processor, that has ECPS. It provides, if desired, a hardware updating of each virtual machine's interval timer at location X'50'.

virtual machine (VM). A functional equivalent of a real machine.

virtual machine assist (VMA). A hardware feature available on certain VM/SP-supported System/370 models, that causes a significant reduction in the real supervisor state time that VM/SP uses to control the operation of virtual storage systems such as VSE, DOS/VS and OS/VS and, to a lesser extent, CMS, DOS, and OS when running under VM/SP. VM/SP supervisor state time is reduced because the VMA feature, instead of VM/SP, intercepts and handles interruptions caused by SVCs, other than SVC 76, and certain privileged instructions. See *CP assist, expanded virtual machine assist, Extended Control Program Support (ECPS:VM/370),* and *virtual interval timer assist.*

virtual machine communication facility (VMCF). A CP function that provides a method of communication and data transfer between virtual machines operating under the same VM/SP system.

Virtual Machine/VTAM Communications Network Application (VM/VCNA). A program that runs in the VTAM service machine. VM/VCNA controls the physical appearance of the screen when displaying output on a VM/SP terminal attached to an SNA network.

virtual printer (or punch). A printer (or card punch) simulated on disk by CP for a virtual machine. The virtual device type and I/O address are usually defined in the VM/SP directory entry for that virtual machine.

virtual=real area. The part of real storage, starting with real page 1, where a virtual=real machine can execute. CP maintains control of real page zero; only page zero of the virtual=real machine is relocated. Only one virtual machine at a time can occupy the virtual=real area. The area must be defined during VM/SP system generation to contain the largest virtual=real machine likely to run. See *virtual=real option.*

virtual=real option. A VM/SP performance option that lets a virtual machine run in VM/SP's virtual=real area. This option eliminates CP paging and, optionally, CCW translation for this virtual machine. Synonymous with *V=R*.

virtual reserve/release. A function that lets many operating systems such as MVS, SVS, VS1, and VM/SP itself all run as virtual machines under the same VM/SP operating system and have data protection on a minidisk. It prevents many users of the same data file from simultaneously accessing the same data, particularly when that data is being updated.

virtual spooling device. Unit record devices simulated on disk by CP. The virtual device type and I/O addresses are defined in the VM/SP directory or else by the CP DEFINE command. CP's spooling facilities let multiple virtual machines share the real unit record devices. In addition, VM/SP has extended the concept of spooling to include the logging of all I/O to and from the virtual machine system console. See *virtual card reader, virtual console spooling*, and *virtual printer (or punch)*.

virtual storage. Storage space that can be regarded as addressable main storage by the user of a computer system in which virtual addresses are mapped into real addresses. The size of virtual storage is limited by the addressing scheme of the computing system and by the amount of auxiliary storage available, and not by the actual number of main storage locations.

virtual storage access method (VSAM). An access method for direct or sequential processing of fixed and variable-length records on direct access devices. The records in a VSAM data set or file can be organized in logical sequence by a key field (key sequence), in the physical sequence in which they are written on the data set or file (entry-sequence), or by relative-record number.

Virtual Telecommunications Access Method (VTAM). An IBM licensed program that controls communication and the flow of data in a computer network. It provides single-domain, multiple-domain, and multiple-network capability. VTAM runs under MVS, OS/VS1, VM/SP, and VSE.

VMCF. Virtual machine communication facility.

VM/SP directory. A CP disk file that defines each virtual machine's typical configuration; the user ID, password, regular and maximum allowable virtual storage, CP command privilege class or classes allowed, dispatching priority, logical editing symbols to be used, account number, and CP options desired. Synonymous with *CP directory.*

VM/SP hardware assist. A general term that represents any or all of the following hardware assist functions: expanded virtual machine assist, ECPS, virtual interval timer assist, and VMA.

VM/VCNA. Virtual Machine/VTAM Communications Network Application.

VM/VS handshaking feature. A communication interface between VM/SP and other operating systems running a virtual machine under VM/SP. These operating systems and CP make each other aware of mutual capabilities and requirements.

volid. Volume identifier.

volume identifier (volid). The volume identification label for a disk.

V=R. Synonym for *virtual=real option.*

VSAM. Virtual storage access method.

VSCS. VTAM SNA Console Support.

VTAM. Virtual Telecommunications Access Method.

VTAM service machine (VSM). A virtual machine that contains an operating system (OS/VS1 or DOS/VSE), an access method (ACF/VTAM or ACF/VTAME), and VM/VCNA. VSM forms the interface for SNA communication in VM/SP.

W

working set. The estimated number of pages of real storage that the virtual machine needs to execute.

Numerics

2305. Refers to the IBM 2305 Fixed Head Storage Device, Models 1 and 2.

2741. Refers to the IBM 2741 Terminal. Information on the 2741 also applies to the IBM 3767 Terminal, unless otherwise noted.

3088. Refers to the IBM 3088 Multisystem Communications Unit, Models 1 and 2.

3262. Refers to the IBM 3262 Printer, Models 1 and 11.

3270. Refers to a series of IBM display devices; for example, the IBM 3275, 3276 Controller Display Station, 3277, 3278, and 3279 Display Stations, the 3290 Information Panel, and the 3287 and 3286 printers. A specific device type is used only when a distinction is required between device types. Information about display terminal usage also refers to the IBM 3138, 3148, and 3158 Display Consoles when used in display mode, unless otherwise noted.

3284. Refers to the IBM 3284 Printer. Information on the 3284 also pertains to the IBM 3286, 3287, 3288, and 3289 printers, unless otherwise noted.

3289. Refers to the IBM 3289 Model 4 Printer.

3310. Refers to the IBM 3310 Direct Access Storage Device.

3330. Refers to the IBM 3330 Disk Storage Device.

3340. Refers to the IBM 3340 Direct Access Storage Device.

3350. Refers to the IBM 3350 Direct Access Storage Device when used in native mode.

3370. Refers to the IBM 3370 Direct Access Storage Device.

3380. Refers to the IBM 3380 Direct Access Storage Device.

3725. Refers to the IBM 3725 Communication Controllers.

3800. Refers to the IBM 3800 Printing Subsystems. A specific device type is used only when a distinction is required between device types.

3850. Refers to the IBM 3850 Mass Storage System.

4245. Refers to the IBM 4245 Printer.

4248. Refers to the IBM 4248 Printer.

9313. Refers to the IBM 9313 Direct Access Storage Device.

9332. Refers to the IBM 9332 Direct Access Storage Device, Model 400.

9335. Refers to the IBM 9335 Direct Access Storage Device, Models A01 and B01.

Bibliography

Prerequisite Publications

Virtual Machine/System Product:

Introduction, GC19-6200

Terminal Reference, GC19-6206

CMS User's Guide, SC19-6210.
(This book is a prerequisite only if you edit files used for system control or accounting purposes.)

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

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

Communication Network Application General Information Manual, GC27-0501

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

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

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

ACF/VTAM Network Program Products Planning, SC23-0110

ACF/VTAM VTAM Installation and Resource Definition, SC23-0111

ACF/VTAM VTAM Customization, SC23-0112

ACF/VTAM VTAM Operation, SC23-0113.

Corequisite Publications

Virtual Machine/System Product:

Planning Guide and Reference, SC19-6201

Operator's Guide, SC19-6202

System Messages and Codes, SC19-6204

Library Guide and Master Index, GC19-6207

CMS Command Reference, SC19-6209

System Product Editor User's Guide, SC24-5220

System Product Editor Command and Macro Reference, SC24-5221

Installation Guide, SC24-5237

System Product Interpreter User's Guide, SC24-5238

System Product Interpreter Reference, SC24-5239

Distributed Data Processing Guide, SC24-5241

Interactive Problem Control System Guide and Reference, SC24-5260

Administration, SC24-5285

Application Development Guide for CMS, SC24-5286

System Facilities for Programming, SC24-5288

Release 6 Guide, SC24-5368

CMS Shared File System Administration, SC24-5367

Connectivity Programming Guide and Reference, SC24-5377

Connectivity Planning, Administration, and Operation, SC24-5378

CP Data Areas and Control Blocks, LY24-5220

CMS Data Areas and Control Blocks, LY24-5221

VM/SP Diagnosis Guide, LY24-5241.

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

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

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

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

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

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

Other corequisite publications include:

RSCS Program Reference and Operations Manual, SH24-5005

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

EREP User's Guide and Reference, GC28-1378

2821 Control Unit Component Description, GA24-3312

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

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

3800 Printing Subsystem Programmer's Guide, GC26-3846

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

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

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

Guide to Using the 3705 Control Panel, GA27-3087

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

EP/3725 Generation and Utilities Guide and Reference, SC30-3172

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

The following are corequisite publications if your installation has Print Services Facility for Advanced Function Printers:

AFP General Information Manual, G544-3415

AFP Printer Summary, G544-3135

AFP Diagnosis Guide for MVS, LV32-0516

PSF/MVS System Programmer's Guide, SH35-0091

PSF/MVS User's Programming Guide, S544-3084

PSF/MVS Messages and Codes, S544-3431 1

PSF/MVS Diagnosis Guide and Reference, LV32-0515

PSF Data Stream Reference, SH32-0073

PSF/MVS Customer Programming Interfaces, G544-3124

VM/SP RELEASE 6 LIBRARY

Evaluation

	General Information	GC20-1838
▨	Introduction	GC19-6200
	Introduction to Security	SC24-5316

Installation and Service

▨	Installation Guide	SC24-5237
▨	Service Guide	SC24-5389

Planning

▨	Planning Guide and Reference	SC19-6201
	VM Running Guest Operating Systems	GC19-6212
▨	Release 6 Guide	SC24-5368

Operation

▨	Operator's Guide	SC19-6202

Administration

▨	VM System Facilities for Programming	SC24-5288
▨	Administration	SC24-5285
▨	Connectivity Planning, Administration, and Operation	SC24-5378
▨	CMS Shared File System Administration	SC24-5367
▨	CP System Command Reference	SC24-5402

Application Development

▨	Application Development Guide for FORTRAN and COBOL	SC24-5247
▨	Programmer's Guide to the SRPI for VM/SP	SC24-5291
▨	Application Development Guide for CMS	SC24-5286
▨	Application Migration Guide for CMS	SC24-5366
▨	Application Development Reference for CMS	SC24-5284
▨	Connectivity Programming Guide and Reference	SC24-5377
▨	GCS Command and Macro Reference	SC24-5250
▨	**SAA Common Programming Interface Communications Reference**	**SC26-4399**
▨	Directory of Programming Interfaces for Customers	GC24-5417

Index/Glossary

▨	Library Guide and Master Index	GC19-6207
	Glossary	SC24-5379

End Use

	Quick Reference	SX20-4400
▨	CMS Primer	SC24-5236
	CMS Primer for Line-Oriented Terminals	SC24-5242
▨	CMS User's Guide	SC19-6210
▨	CMS Command Reference	SC19-6209
▨	System Product Editor User's Guide	SC24-5220
▨	System Product Editor Command and Macro Reference	SC24-5221
▨	System Product Interpreter User's Guide	SC24-5238
▨	System Product Interpreter Reference	SC24-5239
▨	EXEC 2 Reference	SC24-5219
▨	CP General User Command Reference	SC19-6211
▨	Terminal Reference	GC19-6206

▨ one copy of each shaded manual automatically received with product tape

VM/SP RELEASE 6 LIBRARY

Diagnosis

System Messages and Codes	SC19-6204
System Messages Cross-Reference	SC24-5264
Service Routines Program Logic	LY20-0890
Interactive Problem Control System Guide and Reference	SC24-5260
Diagnosis Guide	LY24-5241
CP Diagnosis Reference	LY20-0892
CP Data Areas and Control Blocks	LY24-5220
CMS Diagnosis Reference	LY20-0893
CMS Data Areas and Control Blocks	LY24-5221
VM CP Trace Table (Poster)	SX24-5225
Problem Determination Summary	SX24-5224

Reference Summaries

SP Editor Command Language Reference Summary	SX24-5122
EXEC 2 Language Reference Summary	SX24-5124
CMS Primer Summary of Commands	SX24-5151
SP Interpreter Reference Summary	SX24-5126
CMS Primer for Line-Oriented Terminals Summary of Commands	SX24-5159
HELP Facility Introduction	SX24-5221
CP General User Command Reference Summary	SX24-5219
CP System Command Reference Summary	SX24-5222
CMS Command Reference Summary	SX24-5220
VM Summary of End Use Tasks and Commands (Poster)	SX24-5173

Auxiliary Communication Support

VTAM Installation and Resource Definition	SC23-0111
VTAM Customization	SC23-0112
VTAM Operation	SC23-0113
VTAM Messages and Codes	SC23-0114
VTAM Programming	SC23-0115
VTAM Diagnosis Guide	LY30-5601
VTAM Programming for LU 6.2	SC30-3400
VTAM Data Areas (VM)	LY30-5593
VTAM Reference Summary	LY30-5600
VM/Pass-Through Facility Overview	GC24-5373
VM/Pass-Through Facility: Managing and Using	SC24-5374
RSCS Exit Customization	SH24-5197
RSCS General Information	GH24-5055
RSCS Planning and Installation	SH24-5057
RSCS Messages and Codes	SH24-5196
RSCS Operation and Use	SH24-5058
RSCS Diagnosis Reference	LY24-5228
RSCS Reference Summary	SX24-5135

For the relationship of the *VM/SP CP General User Command Reference* to other VM/SP prerequisite and corequisite publications, see "The VM/SP Library."

Index

A

abbreviation, use of 5
accounting information, displaying 194
ADSTOP command
  description of when to use 50
  format 50
  halting execution of virtual machine 25
  messages 51
  operands
    hexloc 50
    OFF 50
  response 51
  resuming operation with BEGIN command 50
  summary 259
  use with TRACE command 50
Advanced Function Printing 17
affinity setting in relation to the logon process 121
altering
  attributes of
    virtual printer file 54
    virtual punch file 54
    virtual reader file 54
  configuration or channel operating mode 74
  logical line-editing symbols 233
analyst function type 42
annotating your console sheet 47
APL used to control translation tables 234
ATTACH command
  use with dedicated devices 19
attached processor application using SET ASSIST 194
attached processor, INDICATE command use for monitor of system load 99
attention
  handling, terminal control of 233
  interrupt
    entering from your virtual console 52
    for a virtual machine 183
ATTN command
  description of when to use 52
  format 52
  functioning as REQUEST command 52
  simulating real console function 22
  summary 259
  used during I/O operation 52
  used with BEGIN command 52
attributes
  changing
    printer files 214
    punch files 214
    reader files 213
  modifying for 3800 printer 217
  spool file, changing 54
  virtual device modification 213

B

BEGIN command
  description of when to use 53
  format 53
  issued with specific storage location 53
  issued without hexloc 53
  message 53
  response 53
  resuming operation of virtual machine after ADSTOP 50
  summary 259
  used with ATTN command 52
  used with the DISCONN command 85
blanks as delimiters 3
block address used to load a system, at IPL 103
braces, nested to indicate choices 6, 7
brackets, nested to indicate choices 6, 7
branch traceback table 35
branches, tracing 130, 242
byte alignment on TERMINAL output 25

C

carriage control word (CCW)
  command code processing for 3215 238
  command code processing for 3270 238
  translating control 199
  use with LOADVFCB command 115
CAW (channel address word)
  displaying 89
CCW translation control 199
CCW (carriage control word)
  command code processing for 3215 238
  command code processing for 3270 238
  translating control 199
  use with LOADVFCB command 115
CE function type 42
CHANGE command
  correcting invalid or undefined option names 57
  description of when to use 54
  format 54
  messages 58
  nullifying settings or operands 57
  operands
    ALL 55
    CHars 56
    CLass 55
    COpy 55
    DEST 55
    DIst 55
    FCB 56
    FLash 55
    FORM 55
    HOld 55

CHANGE command *(continued)*
operands *(continued)*
MOdify 55
NAme 56
NOHold 55
Printer 54
PUnch 54
Reader 54
UNCONV 56
responses 57
summary 259
changing
attributes of spool files 54
configuration or channel operating mode 74
control registers 224
destination name 55
file name of a spool file 56
floating-point registers 224
general registers 224
logical line editing symbols 264
Program Status Word (PSW) 224
spool class of a file 55
spoolid of a file 55
trace elements 147
virtual storage locations 224
channel address word (CAW)
displaying 89
channel status word (CSW)
displaying 89
use with TERMINAL command 241
channel-to-channel device
connecting 65
coupling to CTCA 65
receiving spurious attention interrupt 75
character
deleting from line 263
set 3
translation tables for APL, controlling 233
class description
ANY 11
G 10
clearing
interrupts from a virtual machine 184
storage to zero 103
CLOSE command
action taken for virtual input device 61
action taken for virtual output device 62
description of when to use 59
format 59
HOLD restrictions 61
messages 63
NOHOLD restrictions 61
operands
CONsole 59
DEST 60
DIst 60
FORM 60
HOld 59
NAme 60

CLOSE command *(continued)*
operands *(continued)*
NOHold 60
Printer 59
PUnch 59
PUrge 60
Reader 59
vaddr 59
response 63
summary 259
using with spool files 15
closing
files 59
TRACE output 30
virtual devices 216
CMS initialization parameters
AUTOCR 104
BATCH 104
FILEPOOL 105
INSTEG 105
NOSPROF 105
SAVESYS 105
command environment
CP 3
defined 3
command name 4
command operands definition 4
commands
belonging to more than one functional type 64
CP summarized 259
entering from CP console function mode 23
entering from the virtual machine environment 67
entering multiple commands using #CP 48
function types 41
LANG option 12
nesting 7
notational conventions 5
privilege classes
assignments 41
description 41
summary of use 259
text format of 45
used to control a terminal session, summary of 259
COMMANDS command
description of when to use 64
format 64
response 64
summary 259
comments, how to write 4
connecting
remote terminals to virtual machines 82
virtual channel-to-channel adapters 65
console
input/output, terminating 249
sheet, annotation of 47
spool file, modifying 250
spooling, controlling 250

continuous reading of an entire file 214
Control Program, console function mode 23
controlling
  a terminal session, summary of commands used 259
  APL translation tables 233
  CCW translation 199
  editing symbols 233
  error message handling 198
  informational message handling 198
  input/output processing 233
  line editing symbols 199
  program function keys 200
  setting functions for your virtual machine 192
  use of logical line editing symbols 199
  virtual timer 206
  3270 display 234
controlling virtual machine during terminals session 10
converted file 14, 17, 56, 252
COUPLE command
  description of when to use 65
  format 65
  messages 66
  operands
    To userid 65
    vaddr1 65
    vaddr2 65
  responses 65
  summary 259
CP command environment 3
CP (Control Program) commands
  description of when to use 67
  executing while in fullscreen mode 67
  format 67
  responses 67
  summary 259
  treated as null line 67
creating disk space with DEFINE command, precautions 68
CSW (channel status word)
  displaying 89
  use with TERMINAL command 241

D

debugging
  programs 24
  using PER 130
dedicated channel detaching 81
defaults used in underscored notation 6
DEFINE command
  description of when to use 68
  format 68
  messages 76
  operands
    CHANnels 71
    CONsole 69
    CTCa 69
    GRAF 72
    LIne 72

DEFINE command *(continued)*
  operands *(continued)*
    Printer 69
    PUnch 69
    Reader 69
    STORage 73
    TIMer 69
    vaddr1 73
    1403 69
    1443 70
    2501 70
    2540P 70
    2540R 70
    3088 70
    3203 70
    3211 70
    3262 70
    3289E 70
    3505 70
    3525 70
    3800 71
    3800-1 71
    3800-3 71
    4245 70
    4248 71
  reconfiguring virtual machine 24
  responses 75
  summary 259
  temporary minidisks 72
  use with MSS devices 68
defining
  logical line editing symbols 264
  shared and nonshared device on same virtual control unit 74
  spool classes to print classification titles 56
  temporary disks 13
  temporary virtual disk 75
  virtual devices 68
destination name
  changing on spool files 55
  purging files 149
  rearranging spool files 127
  transferring files 249
DETACH CHANNEL command
  description of when to use 81
  format 81
  messages 81
  response 81
DETACH command
  description of when to use 77
  format 77
  messages 79
  operands
    vaddr 77
  removing dedicated devices 19
  responses 78
  summary 259
  temporary disks 13

DETACH command *(continued)*
use with dedicated channels 19
detaching
dedicated channels 81
dedicated 2305 77
previously attached device 77
real devices from another user 78
virtual devices 77
device-end interrupt pending for a virtual device 182
devices
detaching
previously attached 77
2305 77
virtual
linking to 109
simulating not ready status for 126
DIAL command
description of when to use 82
format 82
messages 84
operands
userid 82
vaddr 82
responses 83
restrictions 82
summary 260
directory entry, controlling, virtual machine attributes 11
DISCONN command
description of when to use 85
format 85
logging off after 15 minutes 85
messages 86
operand
HOld 85
responses 86
summary 260
writing output messages 85
disconnecting
virtual console 85
virtual machine 85, 117
DISPLAY command
byte alignment 26
description of when to use 87
entering multiple operands 89
format 87
messages 92
operands
bytecount 88
CAW 89
CSW 89
Gregl 88
hexlocl 87
hexloc2 88
Psw 89
regcount 89
reg2 88
Xregl 88
Yregl 88

DISPLAY command *(continued)*
responses 90
summary 260
terminating 90
using to display virtual storage 26
displaying
accounting information 194
Channel Address Word (CAW) 89
Channel Status Word (CSW) 89
Program Status Word (PSW) 89
registers
extended control 88
floating point 88
general 88
storage keys 87
storage location, registers, and control word contents 90
virtual storage 87
distribution code for an output file 55
DUMP command
description of when to use 93
examining large portions of storage 90
format 93
messages 96
operands 93
printing virtual storage 27
responses 95
summary 260

E

ECHO command
description of when to use 97
format 97
responses 97
summary 260
ECMODE setting 197
editing
controlling 233
editing symbols
logical line editing 263
logical, controlling use of 199
with logical line editing symbols 263
entering
a CP command from a virtual machine environment 67
commands from CP console function mode 23
CP command 3
logical line editing symbols as data 264
more than one on command line 263
multiple command lines as operands of CP command 67
error message handling 198
execution
halting at an instruction address 50
of a virtual machine, resuming 53
extended control mode, setting 197

extended control registers
  displaying 88
  printing 93
EXTERNAL command
  description of when to use 98
  format 98
  message 98
  simulating interruptions 21
  summary 260
external interrupt, simulated 98

F
FB-512 device information, obtained with QUERY command 173
FCB (forms control buffer)
  names used to avoid mismatch 57
files
  changing
    destination name 55
    spool class of 55
    spoolid for 55
  closing 59
  distribution code for 55
  holding
    before output processing 55
    from further processing 60
  holding from further processing 214
  purging 60, 217
  reclaiming 249
  reordering closed spool files 127
  spool, changing file name for 56
  terminating processing of 59
  transferring to another user 215
  transmitting to a virtual reader 249
first-level storage requested via DISPLAY command 89
floating-point registers
  changing your virtual machine 224
  displaying 88
  printing 93
forms control buffer (FCB)
  names used to avoid mismatch 57
forms controls for a virtual printer, specifying 114
function types of CP command
  analyst 42
  CE 42
  general 42
  operations 42
  programmer 42
  resource 42
  service 42
  spooling 42

G
general function type 42
general registers
  changing in your virtual machine 224
  displaying 88
  printing 93
general user description 10

H
halting execution
  at an instruction address 50
  of a virtual machine 50
hiding your password 120
holding
  files by user hold status 214
  files from further processing 60, 214

I
INDICATE command
  attached processor load indication 99
  description of when to use 99
  format 99
  messages 102
  multiprocessor load indication 99
  operands
    LOAD 99
    USER 99
  responses 99, 101
  summary 260
informational message handling 198
initial program load (IPL)
  automatic, suppression of 120
  of a virtual machine operating system 103
  stopping 103
input/output operations, tracing 242
internal print duplication of 3800 printer 55
interrupt
  clearing from a virtual machine 184
  device-end, specifying virtual 182
  external, simulating 98
  for a virtual machine 183
  range of external codes 98
  tracing 242
interrupt codes
  X'1004', clock comparator 21
  X'1005', CPU timer 21
interruption simulation 21
IPL command
  description of when to use 103
  executing system profile 107
  for named systems 103
  format 103
  messages 108
  operands
    ATTN 104
    CLear 103
    cylno 103
    nnnnnn 103
    NOCLear 103

IPL command *(continued)*
  operands *(continued)*
    PARM 104
    STOP 103
    systemname 104
    vaddr 103
  optional CMS initialization parameters
    AUTOCR 104
    BATCH 104
    FILEPOOL 105
    INSTSEG 105
    NOSPROF 105
    SAVESYS 105
  responses 108
  summary 260
IPL (initial program load)
  automatic, suppression of 120
  of a virtual machine operating system 103
  stopping 103
I/O devices, defining for your virtual machine 68

K

keys
  program function controlling 200
  storage
    displaying 87
    printing 93

L

LANG option on CP commands 12
line editing symbols control 199
line length, controlling at your terminal 233
lines, deleting at terminal before entering 263
LINK command
  batch job restrictions 109
  description of when to use 109
  format 109
  messages 113
  operands 109
  passwords 109
  responses 111
  summary 260
  2305 restrictions 109
linking a device to your virtual machine 109
load key simulation via IPL command 103
loading a system with a block address at IPL 103
loading an operating system 20
LOADVFCB command
  description of when to use 114
  format 114
  messages 116
  operands
    Fcb 114
    Index 114
    name 114
    vaddr 114
LOADVFCB command *(continued)*
  summary 260
logging off your virtual machine 117
logging on your virtual machine 120
logical editing symbols
  altering 234
  character delete 263
  controlling 234
  controlling use of 199
  defining 264
  escape 264
  line delete 263
  line end 263
logical printer
  processing
    SEND command 190
logically connected terminal 82
LOGIN (LOGON) command
  description of when to use 120
  format 120
  messages 122
  operands
    Noipl 120
    password 120
    userid 120
  reconnecting disconnected machine 85
  responses 121
  summary 260
LOGOFF (LOGOUT) command
  description of when to use 117
  format 117
  messages 119
  operand
    HOld 117
  responses 118
  summary 260
logon procedure, hiding your password during 120
LOGON (LOGIN) command
  description of when to use 120
  format 120
  messages 122
  operands
    Noipl 120
    password 120
    userid 120
  reconnecting disconnected machine 85
  responses 121
  summary 260
LOGOUT (LOGOFF) command
  description of when to use 117
  format 117
  messages 119
  operand
    HOld 117
  responses 118
  summary 260
lowercase letters, words and symbols 5

M

masking password during logon 120
Mass Storage System (MSS)
  defined with DEFINE command 73
merging trace elements 147
MESSAGE command
  description of when to use 124
  format 124
  messages 125
  obtaining operator assistance 19
  operands
    msgtext 124
    OPerator 124
    userid 124
    * 124
  responses 125
  summary 260
messages
  controlling transmission of 199, 204
  handling error messages 198
  handling informational messages 198
  sending to operator 124
  sending to other users 124
modifying
  spooling control options 214
  storage, using ADSTOP command 50
  virtual device attributes 213
MSS (Mass Storage System)
  defined with DEFINE command 73
multiprocessor, INDICATE command to display system load 99

N

named systems of IPL command 103
nested commands 7
notational conventions 5
NOTREADY command
  description of when to use 126
  format 126
  messages 126
  summary 260

O

operands definition 4
operating system
  for a virtual machine, passing parameters to 103
  initial program load for 103
  loading 20
  passing parms at IPL 20
operations function type 42
operator
  CP commands 45
  primary system 42
  sending messages to 124
  spooling 42
  system resource 42
options on trace elements 147
ORDER command
  description of when to use 127
  format 127
  messages 128
  operands
    CLass 127
    DEST 127
    FORM 127
    Printer 127
    PUnch 127
    Reader 127
    spoolid 127
  reordering spool files 18
  response 128
  summary 260

P

parameters, passing to a virtual machine operating system 103
password
  hiding during logon procedure 120
  invalid format for commandline, precaution 120
  with the LINK command 109
PER command
  description of when to use 129
  event types
    BRanch 131
    G 133
    hardware usage 143
    Instruct 131
    Mast 132
    STore 132
  format 130
  messages 148
  options
    BOth 135
    CMd 37, 135
    DAT 136
    DATOFF 136
    FRom 135
    GUESTR 38, 136
    GUESTV 38, 136
    NORun 136
    PAss 135
    Printer 135
    Range 134
    RUN 136
    STEp 136
    TErminal 135
  program debugging, using 34
  responses 145
  selectivity, with 32
  storage alteration 38
  subcommands
    COunt 35, 137
    END 138
    GET 138

PER command *(continued)*
  subcommands *(continued)*
    SAve 137
    TAble 137
  summary 260
  suspending 33
  terminating 33
  trace elements
    changing options 139
    changing specific ones 140
    merging 139
  using 31
power off vs logoff with 3270 terminal 117
print file requirements for 3800 printer 56
printer files, virtual, altering the attributes of 56
printer, virtual, specifying forms controls for 114
printing
  classification titles on separator pages 56
  contents of virtual machine components 93
  extended control registers 93
  first-level storage with DUMP command 93
  floating-point registers 93
  general registers 93
  Program Status Word (PSW) 93
  virtual storage keys 93
  virtual storage locations 93
privilege classes
  CP command 41
  description 41
  user and function 41
privileged instructions, tracing 242
processor, determining if attached or main 156
program debugging using PER 34
program function (PF) keys control 200
program status word (PSW)
  changing your virtual machine 224
  displaying 89
  printing 93
  used with BEGIN command 53
programmer function type 42
PSF 17
PSW (program status word)
  changing your virtual machine 224
  displaying 89
  printing 93
  used with BEGIN command 53
punch files, virtual, altering the attributes of 56
PURGE command
  deleting spool files 18
  description of when to use 149
  format 149
  messages 150
  operands
    ALL 149
    CLass 149
    DEST 149
    FORM 149
    Printer 149
    PUnch 149

PURGE command *(continued)*
  operands *(continued)*
    Reader 149
    spoolid 149
  response 150
  summary 260
purging
  closing files 60
  spooling files 217
purging closed spool files 150

Q

QUERY command
  description of when to use 151, 156
  determining current attributes of files 56
  determining device or file attributes 15
  messages 180
  operands and responses
    CPLANG 151
    CPLEVEL 158
    CPUid 158
    Files 159
    LINKS 160
    LOGmsg 152
    Names 152
    PER 153
    PFnn 160
    Printer 161
    PROCessr 165
    PUnch 161
    Reader 161
    SCREen 165
    SECuser 166
    Set 167
    SPMODE 167
    S370E 168
    TERMinal 168
    Time 169
    USERID 170
    Users 155
    Virtual 171—179
    VMSAVE 179
  summary 260
  verifying new attributes of files 56

R

reader files, virtual, altering the attributes of 56
reading an entire file continuously 214
READY command
  description of when to use 182
  format 182
  messages 182
  summary 261
read/write access with LINK command 109
real computer
  reset button simulation 227

real computer *(continued)*
- restart button simulation 227

reclaiming
- files in order to purge 150
- transmitted files 251

reconfiguring your virtual machine 24
registers
- extended control
  - displaying 88
  - printing 93
- floating-point
  - displaying 88
  - printing 93
- general
  - displaying 88
  - printing 93

remote spooling networks
- receiving files 222
- transmitting files 221

remote spooling using TAG command 229
remote terminal, connecting to virtual machines 82
removing a virtual device from your virtual machine 77
REQUEST command
- description of when to use 183
- format 183
- functioning as ATTN command 52
- simulating real console function 22
- summary 261
- used during I/O operation 52

reset button simulation 227
RESET command
- description of when to use 184
- format 184
- messages 184
- response 184
- summary 261

resource function type 42
responses to CP commands, types of 45
responses vs system messages 45
restart button simulation 227
resume
- execution of your virtual machine 53
- use of your virtual machine after disconnecting it 85

REWIND command
- description of when to use 185
- format 185
- messages 185
- response 185
- summary 261

rewinding a real tape 185

S

SCREEN command
- description of when to use 186
- example of use 188

SCREEN command *(continued)*
- format 186
- messages 189
- operands
  - area 186
  - extcolor 187
  - exthilight 187
- response 189
- summary 261

selectivity using PER 32
SEND command
- description of when to use 190
- format 190
- messages 191
- operands
  - CP 190
  - lprt 190
  - text 190
  - userid 190
- summary 261

sending messages
- to other users 124
- to the operator 124

service function type 42
SET command
- format 192—193
- messages 207
- operands
  - ACNT 194
  - AFFInity 194
  - ASsist 194
  - AUTOPoll 196
  - CONCeal 196
  - CPCONIO 196
  - CPUid 197
  - ECmode 197
  - EMSG 197
  - IMSG 198
  - ISAM 198
  - LINEDit 199
  - MIH 199
  - MSG 199
  - NOTRans 199
  - PAGEX 200
  - PFnn 200—203
  - RUN 204
  - SMsg 204
  - STMulti 205
  - SVCAccl 205
  - SYBypass 204
  - TIMER 206
  - VMCONIO 206
  - VMSAVE 206
  - WNG 206
  - 370E 207
- resetting after DISCONN command 85
- summary 261
- when to use 192

sharing virtual disks 13
simulating
  not ready status for a virtual device 126
  the reset button on a real computer 227
  the restart button on a real computer 227
SLEEP command
  description of when to use 209
  format 209
  message 210
  summary 261
SMSG command
  description of when to use 211
  format 211
  messages 212
  operands
    msgtext 211
    userid 211
  summary 261
SNA (System Network Architecture)
  use of DIAL command 83
specific trace elements, changing 147
specifying device-end interrupt for a virtual device 182
spool class
  for a file, changing 55
  modifying 214
SPOOL command
  description of when to use 213
  format 213
  messages 223
  operands
    CHars 217
    CLass 214
    CLOSE 216
    CONsole 214
    CONt 214
    COpy 216
    DEST 218
    Dist 219
    EOF 215
    FCB 218
    FLash 217
    For 216
    FORM 218
    HOld 214
    MOdify 217
    NOCont 214
    NOEof 215
    NOHold 215
    NOTErm 218
    OFF 216
    Printer 214
    PUnch 214
    PURGE 217
    Reader 213
    STArt 218
    STop 218
    TErm 218
    To 215
    vaddr 214
SPOOL command *(continued)*
  overridden by CLOSE command 61
  response 222
  summary 261
  use with TAG command 221
spool file
  altering TAG information 229
  changing file name for 56
  changing the attributes of 54
  characteristics
    changing 15
    querying 15
    setting 15
  reading after using CLOSE command 61
  reordering 127
  spoolid 220
  TAG command 229
  transmitting locally 220
  transmitting to remote location 221, 229
spoolid
  closed file 14
  converted file 14, 57, 252
  definition of use 14
  used to resequence files 18
spoolid number 220
spooling
  considerations during abend 222
  considerations during shutdown 222
  control, options, modifying 213
  files containing special 3800 options 57
  operator 42
  virtual console 17
spooling function type 42
spooling output to a 3800 printer 219
spurious attention interrupt 75
status
  virtual machine, determining 157
  words, tracing 242
stopping initial program load (IPL) procedure 103
storage
  alteration tracing using PER 38
  clearing to zeros 103
  examining large portions using DUMP 90
  modification of 50
  storage
    displaying 87
    printing 93
  virtual, altering 28
storage locations
  virtual machine
    altering 224
    displaying 87
    printing 93
STORE command
  altering virtual storage 28
  description of when to use 224
  format 224
  messages 226

STORE command *(continued)*
operands
Greg 225
hexloc 224
Lhexloc 224
Psw 225
Shexloc 224
STATUS 225
Xreg 225
Yreg 225
response 226
simulating hardware store 29
summary 261
storing
CPU status for virtual machine 224
virtual machine data 224
summary of CP commands 259
suppressing
automatic IPL of a virtual machine operating system 120
suspending PER 33
SYSTEM command
description of when to use 227
format 227
messages 228
operands
CLEAR 227
RESET 227
RESTART 227
responses 227
summary 261
system messages vs CP command responses 45
System Network Architecture (SNA)
use of DIAL command 83
system resources, indication of 99

T

TAG command
altering TAG information 229
description of when to use 229
format 229
messages 232
operands
DEv 229
FIle 229
tagtext 229
responses 231
summary 261
transmitting files to remote location 229
use with SPOOL command 221
temporary disk space, formatting and clearing 68
temporary, virtual disks 13
TERMINAL command
description of when to use 233
format 234
local console options not supported across DISCONN 86
TERMINAL command *(continued)*
messages 241
operands
APL 236
ASCiitbl 239
ATtn 237
BREakin 238
BRKkey 238
CHardel 235
CNTL 239
CONmode 238
EScape 235
HIlight 237
LINEDel 235
LINENd 235
LINESize 238
MODE 237
PROMpt 239
SCRNsave 237
SCROll 239
TABchar 236
TEXT 236
TYpe 239
resetting after DISCONN command 85
summary 261
use with #CP command 48
used to set logical line editing symbols 264
terminals
control of attention handling 233
disconnecting from your virtual machine 85
input/output processing control 233
line length, controlling 229
remote, connecting to a virtual machine 82
session
altering virtual storage 28
changing spool file characteristics 15
commands used to control, summary of 259
controlling I/O functions 11
controlling virtual machine 20
debugging programs 24
dedicated channels 19
dedicated devices 18
displaying virtual storage 25
logging on 20
purging spool files 18
querying spool file characteristics 15
reordering spool files 18
setting spool file characteristics 14
simulating interruptions 21
spooling 14
testing programs 24
tracing virtual machine activity 30
virtual console spooling 17
virtual disks 13
virtual disks, permanent 13
virtual disks, temporary 13
virtual unit record devices 14
testing using the ECHO command 97

terminating
- console input/output 218
- DISPLAY function 90
- PER 33
- processing of files 59

testing
- programs 24
- terminal 97

text format of CP commands 45
TEXT used to control translation tables 233
time, determining length for a terminal session 156
TRACE command
- description of when to use 242
- following virtual machine activities 30
- format 242
- messages 248
- operands
  - ALL 243
  - BOTH 243
  - BRanch 243
  - CCW 243
  - CSW 243
  - END 243
  - EXTernal 242
  - INSTruct 243
  - I/O 242
  - NORun 243
  - OFf 243
  - PRINter 243
  - PRIV 242
  - PROgram 242
  - RUN 244
  - SIO 242
  - SNS 243
  - SVC 242
  - TERMinal 243
- responses 245
- summary 262
- use with ADSTOP command 50

TRACE output, closing 30
tracing storage alteration using PER 38
tracing virtual machine activity 30, 242
TRANSFER command
- description of when to use 249
- format 249
- messages 211, 253
- operands
  - ALL 249
  - CLass 249
  - DEST 249
  - FORM 249
  - From 250
  - Printer 249
  - PUnch 249
  - Reader 249
  - spoolid 249
  - To 249
- responses 251

TRANSFER command *(continued)*
- summary 262

transferring
- files
  - to a virtual reader 249
  - to another user 215

translating CCWs control 199
transmitting spool files
- locally 220
- to remote locations 221

U

unconverted file 14, 17, 56, 252
underscore used in notation 6
uppercase letters and words 6
using virtual forms control buffer
- with the 3262 printer 114
- with the 3289E printer 114

using #CP as a logical line-end symbol 48

V

virtual console
- disconnecting from your virtual machine 85
- read environment 23
- spooling 17, 214

virtual devices
- action taken on CLOSE for input device 61
- action taken on CLOSE for output device 62
- closing 216
- defining 68
- detaching from your virtual machine 77
- determining number of your virtual machine 151, 180
- determining the status of 151, 180
- I/O, defining for your virtual machine 68
- removing from your virtual machine 77
- simulating not ready status for 126
- specifying a device-end interrupt for 182

virtual disks, permanent 13
virtual disks, temporary 13
virtual machine environment, entering a CP command from 67
virtual machines
- activity, monitoring 129
- activity, tracing 242
- configuration, altering 68
- determining status of 157
- device, linking to 109
- directory entry 11
- disconnecting 117
- execution, resuming 53
- halting execution 50
- logging off of 117
- logging on 120
- operating system, passing parameters to 103
- placing in a dormant state 209

virtual machines *(continued)*
  reconfiguring 24
  sending and processing special messages 211
  setting control functions for 192
  storing information from 224
  tracing activity 30
virtual printer, specifying forms controls for 114
virtual storage location
  changing 224
  printing 93
virtual storage, altering 28
virtual timer control 206
virtual unit record devices 14
VMDUMP command
  description of when to use 254
  format 255
  messages 258
  operands
    bytecount 256
    DSS 256
    END 255
    FORMAT 256
    hexloc1 255
    hexloc2 255
    SYSTEM 256
    TO 256
    0 255
    *dumpid 257
  response 257
  summary 262

W

writing comments in VM/370 4

Z

zeros
  clearing storage to via IPL 103
  clearing storage to via SYSTEM command 227

Numerics

2305 device
  detaching 77
3203 printer
  LOADVFCB restrictions 114
  specifying forms controls for 70
3211 printer
  specifying forms controls for 70
3262 printer
  loading a virtual forms control buffer 70
3270 device
  CCW command code processing 238
3270 device message length 211
3270 terminal power off vs. logoff precautions 117
3289E printer
  loading a virtual forms control buffer 70
3800 printer
  available width codes 74
  modifying device attributes 217
  modifying spool file attributes 55
  of SPOOL command 219
  print file requirements 56
  restrictions with CLOSE command 61
  spooling output to one 219
  using CHANGE command 54
3850 Mass Storage Systems
  See MSS (Mass Storage System)

Special Characters

¢ logical line delete symbol 263
* (asterisk) command
  description of when to use 47
  format 47
  summary 259
#CP command
  conditions for using 48
  description of when to use 48
  entering multiple command lines 48
  examples of use 49
  executing while in fullscreen mode 67
  format 48
  summary 259
  system responses 49
  use with TERMINAL command 48
  using as the logical line-end symbol 48

Program Number
5664-167

File Number
S370/4300-36

Printed in U.S.A.

VM/SP CP General User
Command Reference
Order No. SC19-6211-05

READER'S
COMMENT
FORM

Is there anything you especially like or dislike about this book? Feel free to comment on specific errors or omissions, accuracy, organization, or completeness of this book.

If you use this form to comment on the online HELP facility, please copy the top line of the HELP screen.

______________ ______________ _______ Help Information line ____ of ____

IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you, and all such information will be considered nonconfidential.

Note: Do not use this form to report system problems or to request copies of publications. Instead, contact your IBM representative or the IBM branch office serving you.

Would you like a reply? ___YES ___NO

Please print your name, company name, and address:

____

____

____

____

IBM Branch Office serving you: ____

Thank you for your cooperation. You can either mail this form directly to us or give this form to an IBM representative who will forward it to us.

SC19-6211-05

Reader's Comment Form

CUT
OR
FOLD
ALONG
LINE

Fold and tape — Please Do Not Staple — Fold and tape

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST-CLASS MAIL PERMIT NO. 40 ARMONK, NY

POSTAGE WILL BE PAID BY ADDRESSEE:

INTERNATIONAL BUSINESS MACHINES CORPORATION
DEPARTMENT G60
PO BOX 6
ENDICOTT NY 13760-9987

Fold and tape — Please Do Not Staple — Fold and tape

®

VM/SP CP General User
Command Reference
Order No. SC19-6211-05

READER'S
COMMENT
FORM

Is there anything you especially like or dislike about this book? Feel free to comment on specific errors or omissions, accuracy, organization, or completeness of this book.

If you use this form to comment on the online HELP facility, please copy the top line of the HELP screen.

______________ ______________ ________ Help Information line ____ of ____

IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you, and all such information will be considered nonconfidential.

Note: Do not use this form to report system problems or to request copies of publications. Instead, contact your IBM representative or the IBM branch office serving you.

Would you like a reply? ___YES ___NO

Please print your name, company name, and address:

____

____

____

____

IBM Branch Office serving you: ____

Thank you for your cooperation. You can either mail this form directly to us or give this form to an IBM representative who will forward it to us.

Reader's Comment Form

CUT
OR
FOLD
ALONG
LINE

Fold and tape — Please Do Not Staple — Fold and tape

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL
FIRST-CLASS MAIL PERMIT NO. 40 ARMONK, NY

POSTAGE WILL BE PAID BY ADDRESSEE:

INTERNATIONAL BUSINESS MACHINES CORPORATION
DEPARTMENT G60
PO BOX 6
ENDICOTT NY 13760-9987

Fold and tape — Please Do Not Staple — Fold and tape

®

VM/SP CP General User
Command Reference
Order No. SC19-6211-05

**READER'S
COMMENT
FORM**

Is there anything you especially like or dislike about this book? Feel free to comment on specific errors or omissions, accuracy, organization, or completeness of this book.

If you use this form to comment on the online HELP facility, please copy the top line of the HELP screen.

____________ ____________ _______ Help Information line ___ of ___

IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you, and all such information will be considered nonconfidential.

Note: Do not use this form to report system problems or to request copies of publications. Instead, contact your IBM representative or the IBM branch office serving you.

Would you like a reply? ___YES ___NO

Please print your name, company name, and address:

____

____

____

____

IBM Branch Office serving you: ____

Thank you for your cooperation. You can either mail this form directly to us or give this form to an IBM representative who will forward it to us.

SC19-6211-05

Reader's Comment Form

CUT
OR
FOLD
ALONG
LINE

Fold and tape | Please Do Not Staple | Fold and tape

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST-CLASS MAIL PERMIT NO. 40 ARMONK, NY

POSTAGE WILL BE PAID BY ADDRESSEE:

INTERNATIONAL BUSINESS MACHINES CORPORATION
DEPARTMENT G60
PO BOX 6
ENDICOTT NY 13760-9987

Fold and tape | Please Do Not Staple | Fold and tape

®

Program Number
5664-167

File Number
S370/4300-36

SC19-6211-05

Printed in U.S.A.