

File No. S370-36
Order No. GC20-1820-0

Systems

IBM Virtual Machine Facility/370: CP Command Reference for General Users

Release 3 PLC 1

This publication is a reference manual for the general class of users that are running systems such as OS, DOS, OS/VS, DOS/VS, CMS, and RSCS in a virtual machine under VM/370.

Each CP command available to the general class of user is listed alphabetically and contains general usage information, the command line format, descriptions of all operands and options, allowable values for operand variables, and default values for optional operands. Also included are tables showing the relationship of the general class of CP commands to the entire set of VM/370 commands.

Prerequisite Publications

IBM Virtual Machine Facility/370:

Operating Systems in a Virtual Machine,
Order No. GC20-1821

Terminal User's Guide, Order No. GC20-1810

The IBM logo, consisting of the letters "IBM" in a bold, sans-serif font with a distinctive horizontal line through the middle of the letters.

First Edition (February 1976)

This edition corresponds to Release 3 PLC 1 (Program Level Change) of IBM Virtual Machine Facility/370, and to all subsequent releases unless otherwise indicated in new editions or Technical Newsletters.

Changes are periodically made to the specifications herein; before using this publication in connection with the operation of IBM systems, consult the latest IBM System/370 Bibliography, Order No. GC20-0001, for the editions that are applicable and current.

Requests for copies of IBM publications should be made to your IBM representative or to the IBM branch office serving your locality.

A form for readers' comments is provided at the back of this publication. If the form has been removed, comments may be addressed to IBM Corporation, VM/370 Publications, 24 New England Executive Park, Burlington, Massachusetts 01803. Comments become the property of IBM.

Preface

This publication is a reference manual that discusses VM/370 CP commands available to the general user.

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

Other VM/370 CP commands, available only to system operators, system programmers and analysts, and service representatives, are described in the publications:

IBM Virtual Machine Facility/370:

Operator's Guide, Order No. GC20-1806

System Programmer's Guide, Order No. GC20-1807

OLTSEP and Error Recording Guide, Order No. GC20-1809

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

Part 1 has three sections:

"Section 1. Introduction and General Concepts" is a summary of the concepts you should be familiar with when you are using VM/370.

"Section 2. The VM/370 Command Languages" describes the VM/370 command environments, the general structure of the command languages, and command truncation and abbreviation.

"Section 3. The CP Command Language" provides additional information on the CP subset of VM/370 commands.

Part 2 has two sections:

"Section 4. Notational Conventions" describes the syntax that is used in this publication to display command formats.

"Section 5. Format of CP Commands" contains reference data for all the Class G and Any commands in alphabetical sequence by command name. The command line format includes all operands along with a description of each. Any limits or

restrictions applicable to the values of variable operands are also defined.

"Appendix A: VM/370 Command Summary" contains all the VM/370 commands, by VM/370 component, with a brief description of each command's function. This is intended to give the general user a perspective of the position that the CP class G and class Any commands occupy in the overall VM/370 command structure.

In this publication, the following terminology is used:

- 2305 refers to the IBM 2305 Fixed Head Storage, Models 1 and 2.
- 3270 refers to both the IBM 3275 Display Station, Model 2 and the IBM 3277 Display Station, Model 2.
- 3330 refers to the IBM 3330 Disk Storage Models 1, 2, 11; the IBM 3333 Disk Storage and Control Models 1 and 11; and the 3350 Direct Access Storage operating in 3330/3333 Model 1 or 3330/3333 Model 2 compatibility mode.
- 3340 refers to the IBM 3340 Disk Storage, Models A2, B1 and B2; and, the 3344 Direct Access Storage, Model B2.
- 3350 refers to the IBM 3350 Direct Access Storage, Models A2 and B2, in native mode.

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

Information on the IBM 3344 Direct Access Storage Device and the IBM 3350 Direct Access Storage contained in this publication is for planning purposes only until the availability of the product.

For a glossary of VM/370 terms, see the IBM Virtual Machine Facility/370: Glossary and Master Index, Order No. GC20-1813.

PREREQUISITE PUBLICATIONS

The IBM Virtual Machine Facility/370: Operating Systems in a Virtual Machine,

Order No. GC20-1821, provides functionally organized usage information and examples pertaining to the class G and Any CP commands, as well as the other classes of CP commands.

The physical operation of the virtual machine console is covered in the IBM Virtual Machine Facility/370: Terminal User's Guide, Order No. GC20-1810.

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

COREQUISITE PUBLICATIONS

Error messages that may be issued to a user during a terminal session are fully

documented in IBM Virtual Machine Facility/370: System Messages, Order No. GC20-1808.

General users intending to run CMS or RSCS will find the respective command languages documented in the following publications:

IBM Virtual Machine Facility/370:

CMS User's Guide, Order No. GC20-1819

CMS Command and Macro Reference, Order No. GC20-1818

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

References in text to titles of prerequisite and corequisite VM/370 publications will be given in abbreviated form.

Contents

PART 1. GENERAL INFORMATION.	9	COUPLE	46
SECTION 1. INTRODUCTION AND GENERAL		CP	48
CONCEPTS.	11	DEFINE	49
Virtual Machine Facility/370	11	DETACH	54
VM/370 Control Program	11	DETACH CHANNEL	56
Time Management.	11	DIAL	57
Storage Management	11	DISCONN.	59
Virtual I/O Management	12	DISPLAY.	61
SECTION 2. THE VM/370 COMMAND LANGUAGES.	13	DUMP	67
VM/370 Command Environments.	13	ECHO	70
Interrupting the Execution of a		EXTERNAL	71
Command	13	INDICATE	72
VM/370 Command Structure	17	IPL.	75
The Command Name	17	LINK	77
The Command Operands	17	LOADVFCB	81
The Command Options (CMS only)	17	LOGOFF	82
Comments in the CP Command Language.	17	LOGON.	84
Character Set Usage.	18	MESSAGE.	86
Truncating and Abbreviating Commands	18	NOTREADY	88
SECTION 3. THE CP COMMAND LANGUAGE	21	ORDER.	89
Privilege Classes for CP Commands.	21	PURGE.	91
The G Class Commands	24	QUERY.	93
The Any Class of Commands.	25	QUERY Command for Class G Users.	94
PART 2. REFERENCE INFORMATION.	27	QUERY Command for All Classes of Users	
SECTION 4. NOTATIONAL CONVENTIONS.	29	(Except Class Any)	104
SECTION 5. FORMAT OF CP COMMANDS	33	READY.	106
*.	34	REQUEST.	107
#CP.	35	RESET.	108
ADSTOP	37	REWIND	109
ATTN	39	SET.	110
BEGIN.	40	SLEEP.	117
CHANGE	41	SPOOL.	118
CLOSE.	43	STORE.	125
		SYSTEM	128
		TAG.	129
		TERMINAL	132
		TRACE.	135
		TRANSFER	140
		APPENDIX A: VM/370 COMMAND SUMMARY	143
		INDEX.	153

Figures

Figure 1.	Attention Handling in VM Mode.....	15	Figure 5.	Commands Accepted from Each User Class.....	23
Figure 2.	Attention Handling in CP Mode.....	16	Figure 6.	Spool Command Options Set for a Virtual Device.....	44
Figure 3.	Character Sets and Their Contents.....	18	Figure 7.	CP Command Summary.....	143
Figure 4.	CP Privilege Class Descriptions.....	22	Figure 8.	CMS Command Summary.....	148
			Figure 9.	RSCS Command Summary.....	152

Part 1. General Information

This part of the publication contains general information about the VM/370 system. It contains three sections.

"Section 1. Introduction and General Concepts" is a summary of the concepts you should be familiar with when you are using VM/370.

"Section 2. The VM/370 Command Language" describes the VM/370 command environments, the general structure of the command languages, and command truncation and abbreviation.

"Section 3. The CP Command Language" provides additional information on the CP subset of VM/370 commands.

Section 1. Introduction and General Concepts

Virtual Machine Facility/370

Virtual Machine Facility/370 (VM/370) is a system control program (SCP) that manages a real computing system so that its resources (CPU, storage, and input/output devices) are made available to many users. The sharing of these resources is performed in such a way that each user appears to have all the facilities required to accomplish his task. Each user has at his disposal the functional equivalent of a real, dedicated computing system. Because this functional equivalent is simulated for the user by VM/370 and does not really exist, it is called a virtual machine.

VM/370 has four components: the Control Program (CP), the Conversational Monitor System (CMS), the Remote Spooling Communications Subsystem (RSCS), and the Interactive Problem Control System (IPCS). CP controls the resources of the real machine to provide an environment in which multiple virtual machines can exist. CMS, a subsystem of VM/370, runs in a virtual machine and provides conversational time-sharing facilities such as the creation and management of files, and the compilation, testing, and execution of problem programs. RSCS, another subsystem of VM/370, also runs in a virtual machine and provides users with the facility to transmit files to and receive files from remote stations. IPCS provides VM/370 problem analysis and problem management facilities. This includes the creation of problem reports, tracking of problems, and an online CP dump analysis.

VM/370 Control Program

The VM/370 Control Program (CP) has at its disposal only a finite amount of real CPU time, real storage, and real input/output devices. Using techniques such as time slicing, paging, and spooling, CP manages these resources so that each virtual machine appears to have the resource it needs, at the time it needs it.

TIME MANAGEMENT

Time slicing is a technique by which each virtual machine is periodically allowed access to the real CPU for an increment of time called a time slice. A scheduling algorithm, or set of rules, in CP controls the access to the real CPU based on the virtual machine's type of operation, readiness for execution, and assigned priority.

STORAGE MANAGEMENT

Paging is a technique used by CP to distribute the available real storage to the virtual machines that are executing. Regardless of the actual size of virtual storage, only those areas (pages) that are currently being referenced are kept in real storage. The rest of virtual storage resides on some direct access storage device (DASD). As

the virtual machines alternate usage of the real CPU and as different parts of a program are executed, pages of virtual storage are shuttled between real storage and DASD.

VIRTUAL I/O MANAGEMENT

The virtual machine operating system is responsible for the operation of all virtual devices associated with it, except for spooled devices as described below. CP is responsible for the interface between the virtual devices and their real counterparts. Under CP, a real device also may be dedicated to a virtual machine. There would then be a direct one-to-one correspondence between a virtual device and an equivalent real device. Real devices may also be shared, where multiple virtual machines have controlled accessibility to one real device.

CP spooling facilities allow multiple virtual machines to share the real unit record devices. Virtual unit record devices that have been designated as spooled are controlled by CP. The data being moved is re-formed by CP into page-sized records (4096-byte blocks) and stored in a VM/370 DASD area. This area serves as intermediate storage between the virtual machine and the real unit record device. Input spool files are made available to designated virtual card readers. Output spool files are scheduled for the real printer or punch whenever the files are closed or when the user logs off the system.

Section 2. The VM/370 Command Languages

Each of the components of VM/370 has its own command language. The CP command language is described in this publication and in the VM/370: Operating Systems in a Virtual Machine. The CMS command language is described in the VM/370: CMS User's Guide and VM/370: CMS Command and Macro Reference. The RSCS command language is described in the VM/370: Remote Spooling Communications Subsystem (RSCS) User's Guide. The IPCS command language is described in the VM/370: Interactive Problem Control System (IPCS) User's Guide.

This section describes the VM/370 command environments, the general structure of the CP, CMS and RSCS command languages, and the command abbreviations and truncations.

There are two types of VM/370 commands: system commands and user-defined commands. The system commands are those defined by the CP, CMS and RSCS command languages. User-defined CMS commands are those you can create yourself using the EXEC command or the LOAD and GENMOD commands. The procedure for creating user-defined CMS commands is described in the VM/370: CMS User's Guide. User-defined CP commands are also allowed; however, your installation's system programmer must create them. The procedure for creating user-defined CP commands is described in the VM/370: System Programmer's Guide.

VM/370 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/370 and issue CP commands.

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

If you are operating under CMS, you can determine which command environment you are in by entering a null line (that is, pressing the Enter key, or equivalent, with no data). VM/370 responds to a null line by displaying the current command environment, CMS or CP.

INTERRUPTING THE EXECUTION OF A COMMAND

Using the Attention key, you can switch command environments (that is, switch from the CP to the virtual machine command environment or vice versa), at the same time that you request an interruption.

Each of the terminals that can be used with the VM/370 system has a key that is the equivalent of the Attention key on the 2741 (with which you signal an attention interruption). Unless otherwise noted, where the term "Attention key" is used in this publication, the phrase "(or equivalent)" is implied.

The name and physical location of the Attention key varies with different styles of terminals supported by VM/370. See the VM/370:

Terminal User's Guide for details on various styles of terminals. What actually happens when you press the Attention key depends on three factors: the command environment of your virtual machine, the mode setting of your terminal, and the number of times the Attention key is pressed.

The commands you issue determine the command environment of your virtual machine; you are either issuing CP commands or some virtual machine operating system commands.

The mode of your terminal is set via the CP TERMINAL command. The MODE operand of this command allows you to set your terminal in either the CP or VM mode. How CP interprets attention interruptions issued by the virtual machine users depends on whether the terminal mode is set to CP or VM.

VM is the default mode setting for all VM/370 users except the primary system operator whose default mode is CP. To enter CP terminal mode, enter:

```
TERMINAL MODE CP
```

If your default terminal mode is CP, or if you issued the above command, pressing the Attention key one or more times forces your virtual machine into CP mode.

To return to the VM terminal mode, enter:

```
TERMINAL MODE VM
```

If the terminal mode is VM, pressing the attention key once passes an interruption pending condition to the virtual machine operating system. Pressing the Attention key twice quickly (or pressing the 3270's PA1 key) places your virtual machine in CP mode.

If you execute CMS from a 3270 device, you may issue the ATTN or REQUEST command if you are in CP mode and want to return to the virtual machine with an attention interrupt. Pressing the Attention key (or the PA1 key for a 3270) will have the same effect. To return to the virtual machine, issue the BEGIN command.

Figures 1 and 2 show the effect of an attention interrupt when your terminal is in either VM or CP mode. Figure 1 shows the action taken by VM/370 when your terminal is in VM mode and you press the Attention key. The first column represents the state of the terminal keyboard when you press the Attention key. The second column represents the number of times you press the Attention key. If your terminal is in the VM mode, one attention requests an interrupt and two quick attentions force your virtual machine to the CP command environment. The third column represents the action VM/370 takes, depending on the number of times you press the Attention key.

Note: If the operator sets his terminal mode to VM and resets his virtual machine, his console acts as described in the first state of Figure 1 (that is, more than one attention is required to unlock the keyboard for CP input).

State	Press ATTN Key	Action
Terminal idle; keyboard entry blocked; virtual machine running	1 >1	Attention interruption pending; virtual machine running Keyboard activated for CP input
Terminal receiving output from virtual machine	1 >1	Attention interruption pending; virtual machine running Keyboard activated for CP input at completion of console I/O
Keyboard activated for input to virtual machine; no data entered or all data deleted	1 >1	Device end (DE) and attention status pending; virtual machine running Unit exception (UE) status pending; keyboard activated for CP input
Keyboard activated for input to virtual machine; some data entered	1 >1	Unit exception (UE) status pending; virtual machine running Device end (DE) status pending; keyboard activated for CP input
Keyboard entry blocked; executing CP command	1 or >1	Attention ignored
Keyboard entry blocked; in SLEEP mode entered via command	1 or >1	Keyboard activated for CP input
Keyboard entry blocked; in SLEEP mode entered via Diagnose instruction	1 or >1	Virtual machine resumes execution
Terminal receiving output from CP but not from user command	1 >1	Attention interruption pending; virtual machine running Keyboard activated for CP input
Terminal receiving output in response to CP command	1 or >1	Output line cancelled and in some cases command output cancelled
Keyboard activated for CP input; no data entered or all data cancelled	1 or >1	Attention interruption made pending; virtual machine running
Keyboard activated for CP input; some data entered	1 or >1	Input line cancelled; keyboard activated for CP input

Figure 1. Attention Handling in VM Mode

Figure 2 shows the action taken by VM/370 when your virtual console is in CP mode. If your virtual console is in CP mode and you press the Attention key one or more times, you force the virtual machine to the CP command environment.

State	Action
Terminal idle; keyboard entry blocked; virtual machine running	Keyboard activated for CP input
Terminal receiving output from virtual machine	Keyboard activated for CP input
Keyboard activated for input to the virtual machine; no data entered or all data deleted	Unit exception (UE) status pending; keyboard activated for CP input
Keyboard activated for input to virtual machine; some data entered	Device end (DE) status pending; keyboard activated for CP input
Keyboard entry blocked; executing CP command	Attention ignored
Keyboard entry blocked; in SLEEP mode entered via command	Keyboard activated for CP input
Keyboard entry blocked; in SLEEP mode entered via Diagnose instruction	Virtual machine resumes execution
Terminal receiving output from CP but not from user command	Keyboard activated for CP input
Terminal receiving output in response to CP command	Output line cancelled and in some cases command output cancelled
Keyboard activated for CP input; no data entered or all data cancelled	Attention interruption made pending; virtual machine running
Keyboard activated for CP input; some data entered	Input line cancelled; keyboard activated for CP input

Figure 2. Attention Handling in CP Mode

VM/370 Command Structure

A VM/370 command consists of a command name, usually followed by one or more positional operands and, in some cases, by an option list. The general form for the command line is:

```
command name | [operand...] [(option...)] ]
```

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

THE COMMAND NAME

The command name is an alphameric symbol of not more than eight characters. In general, the names are verbs that describe the function you want the system to perform. For example, you may want to find out whether a certain user is logged on the VM/370 system. In this case, you would use the CP QUERY command.

THE COMMAND OPERANDS

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

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

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

THE COMMAND OPTIONS (CMS only)

The command options are keywords used to control the execution of the command. The option list must be preceded by a left parenthesis; the closing parenthesis is not necessary.

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/370 commands are entered using a combination of characters from six different character sets. The contents of each of the character sets is described in Figure 3.

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

Figure 3. Character Sets and Their Contents

TRUNCATING AND ABBREVIATING COMMANDS

To make the entering of commands on the keyboard more convenient, VM/370 allows many commands (as well as many operands and options) to be used in a shortened form. The shortened form can be either an abbreviation or a truncation to its minimum form. Minimum truncations are shown within the format box of each command and abbreviations are shown with the description that follows the format box for all the command formats in Section 5.

An example of command truncation is the QUERY command. The format shows the command as "Query". This means you can enter the QUERY command in any of the following forms:

```
query
quer
que
qu
q
```


Q is the minimum truncation, but CP accepts all of the above forms of the command.

Abbreviations are shorter forms of commands and operands. Abbreviations for commands are shown below the full name of the command in the format box. Abbreviations for operands are shown in the description of the individual operands that follow the format box.

An example of command abbreviation is the MESSAGE command. The format shows the command as "Message" and "MSG". This means that not only can you truncate the command to a minimum truncation of "M" but you can also use the abbreviation "MSG". Therefore, CP will accept all the following forms of the MESSAGE command:

```
message
messag
messa
mess
mes
me
m
msg
```


Section 3. The CP Command Language

The CP commands represent a set of interactive console functions that are used (1) by the operator to control the VM/370 system and (2) by a user to control a virtual machine. In this publication, only those commands that are used to control a virtual machine are fully documented. Other commands are referenced only to the extent necessary to show relationships and provide general knowledge.

Privilege Classes for CP Commands

Each user of VM/370 is assigned one or more privilege classes as part of the directory entry of the virtual machine. Each user class is indicative of a specific function of a virtual machine and entitles the user to a specific subset of the CP command language.

Figure 4 lists each CP privilege class along with the associated type of user and function performed. Figure 4 also identifies the specific publications in which each class subset of CP commands is described in detail.

Figure 5 is a list of CP commands by privilege classes. If the available operands for a command vary by class, the valid operands are also listed.

Although users are functionally divided into seven classes, A through G, with a corresponding privilege class of CP commands, there is a small group of commands available to all users. This eighth classification of CP commands is the Any class.

This publication addresses itself as a reference manual for only the class G and Any subsets of the CP command language.

Class	User and Function
A ¹	<p><u>Primary System Operator</u>: The class A user controls the VM/370 system. Class A is assigned to the user at the VM/370 system console during IPL. The primary system operator is responsible for the availability of the VM/370 system and its communication lines and resources. In addition, the class A user controls system accounting, broadcast messages, virtual machine performance options and other command operands that affect the overall performance of VM/370.</p> <p><u>Note</u>: The class A system operator who is automatically logged on during CP initialization is designated as the primary system operator.</p>
B ¹	<u>System Resource Operator</u> : The class B user controls all the real resources of the VM/370 system, except those controlled by the primary system operator and spooling operator.
C ^{1,2}	<u>System Programmer</u> : The class C user updates certain functions of the VM/370 system.
D ¹	<u>Spooling Operator</u> : The class D user controls spool data files and specific functions of the system's unit record equipment.
E ^{1,2}	<u>System Analyst</u> : The class E user examines and saves certain data in the VM/370 storage area.
F ^{1,3}	<u>Service Representative</u> : The class F user obtains, and examines, in detail, certain data about input and output devices connected to the VM/370 system.
G ⁴	<u>General User</u> : The class G user controls functions associated with the execution of his virtual machine.
Any ^{1,4}	The Any classification is given to certain CP commands that are available to any user. These are primarily for the purpose of gaining and relinquishing access to the VM/370 system.
H	Reserved for IBM use.
<p>¹Described in the <u>VM/370: Operator's Guide</u>.</p> <p>²Described in the <u>VM/370: System Programmer's Guide</u> and the <u>VM/370: System Logic and Problem Determination Guide</u>.</p> <p>³Described in the <u>VM/370: OLTSEP and Error Recording Guide</u>.</p> <p>⁴Described in this publication.</p>	

Figure 4. CP Privilege Class Descriptions

Class	Commands	Operands	Class	Commands	Operands
		PAGING USER QUEUES		PURGE QUERY	CHANNELS FILES LINKS LOGMSG NAMES PFnn PRINTER PUNCH READER SET TERMINAL TIME userid USERS VIRTUAL
	LOCATE MONITOR	DISPLAY ENABLE INTERVAL START STOP			
	QUERY	LOGMSG NAMES PAGING PRIORITY SASSIST userid USERS			
F	SAVESYS NETWORK QUERY	TRACE LOGMSG NAMES userid USERS		READY REQUEST RESET REWIND SET	ACNT ASSIST ECMODE EMSG IMSG ISAM LINEDIT MSG NOTRANS PAGEX PFnn PFnn COPY PFnn TAB RUN TIMER WNG
	SET	RECORD MODE			
G	ADSTOP ATTN BEGIN CHANGE CLOSE COUPLE DEFINE DETACH DETACH DISPLAY DUMP ECHO EXTERNAL INDICATE	CHANNEL LOAD USER		SPOOL STORE SYSTEM TAG	DEV FILE QUERY
	IPL LINK LOADVFCB NOTREADY ORDER			TERMINAL TRACE TRANSFER	

Figure 5. Commands Accepted from Each User Class (Part 2 of 2)

THE G CLASS COMMANDS

The general user uses the class G commands to control the virtual machine, much as an operator controls a real machine. The user can stop execution in the virtual machine; restart execution; simulate external, attention, and device ready interruptions; inspect and modify virtual storage, registers and PSW; initiate tracing facilities; and invoke the spooling and disk sharing functions of CP.

THE ANY CLASS OF COMMANDS

The Any privilege class contains those commands that are available, with the same format and operands, to all user classes. Some of these commands are used to gain initial access to the VM/370 system. The LOGON command identifies you to the system and sets up your virtual machine. The DIAL command allows you to establish a logical connection between your terminal and a previously logged-on multiple-access virtual machine. If you cannot log on for some reason, such as a faulty line connection or someone else using your user identification, the MESSAGE command allows you to communicate with the system operator or some other user.

Part 2. Reference Information

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

"Section 4. Notational Conventions" describes the syntax that is used in this publication to display command formats.

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

Section 4. Notational Conventions

The notation used to define the command syntax for VM/370 is:

- Truncations and Abbreviations of Commands

Where truncation of a command name is permitted, the shortest acceptable version of the command is represented by uppercase letters. (Remember, however, that VM/370 commands can be entered with any combination of uppercase and lowercase letters.) The example below shows the format specification for the INDICATE command.

INDicate

This representation means that IND, INDI, INDIC, INDICA, INDICAT, and INDICATE are all valid specifications for this command name.

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

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

Reader

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

READER
RDR

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

In some cases what appears to be a minimum truncation is really the only valid abbreviation. For example, the abbreviation for MEMBER is MEM. Only these two forms are valid and no truncations are allowed. The format box contains

MEMBER { name }
 { * }

and the description that follows the format box is

MEMBER { name }
MEM { * }

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

```

underscore      -
braces          { }
brackets        [ ]
ellipsis        ...

```

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

```

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

```

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

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

```

A
B
C

```

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

Example

The representation

```

A
B
C

```

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

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

Example

The representation

```

[ A ]
< B >
[ C ]

```

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

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

Example

The representation

```
[ A ]  
[ B ]  
[ C ]
```

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

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

Example

The representation

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

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

Section 5. Format of CP Commands

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

*

*

Privilege Class: Any

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

```
| *          | anycomment |
```

Responses

None.

#CP

Privilege Class: Any

Use the #CP command to execute a CP command while in a virtual machine command environment without first signaling attention to get to the CP command environment. The format of the #CP command is:

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

where:

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

Usage Notes

1. The pound sign (#) shown in the preceding format represents the logical line end symbol currently in effect for your virtual machine. If you have redefined the logical line end symbol, #CP is an invalid command; you must substitute your line end symbol for the pound sign when using this command.
2. For the command to operate, the following conditions must be met:
 - The virtual machine must be operating with SET LINEDIT ON (a default).
 - The first three characters of the edited line must be #CP (uppercase or lowercase) with the "#" representing the logical line end symbol currently defined.
 - At least one blank must separate the #CP from any command line. Do not use attention interruption in any part of the line or to enter the line.
 - If in a virtual machine command environment, the virtual machine must have issued a READ for at least four bytes of data. If the READ is for less than four bytes you must use the Attention key to enter the CP environment.
3. You can enter multiple command lines as operands of the #CP command provided that (a) the virtual machine's console READ is for at least the number of bytes required to enter the entire line and (b) all command lines to be executed must be separated by the current logical line end symbol.
4. If you enter #CP with no operands, the virtual machine enters the CP environment. CP cancels the virtual machine's console READ by returning a unit exception status for the virtual console. The virtual operating system then reissues the console READ to allow you to key in the appropriate response to a previous message from that machine's operating system.

#CP

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

Responses:

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

CP

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

ADSTOP

Privilege Class: G

Use the ADSTOP command to halt the execution of a virtual machine at a virtual instruction address. Execution halts when the instruction at the address specified in the command is the next instruction to be executed. The format of the ADSTOP command is:

ADSTOP		{ hexloc }
		{ OFF }

where:

hexloc is the hexadecimal representation of the virtual instruction address where execution is to be halted. Since ADSTOP modifies storage, an address specified within a shared segment results in the virtual machine being placed in nonshared mode with its own copy of the shared segment. A fresh copy of the shared segment is then loaded for the use of the other users.

OFF cancels any previous ADSTOP setting.

Usage Notes

- 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 may be removed by one of 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
- Since the ADSTOP function modifies storage by placing a CP SVC X'B3' at the specified location, you should not:
 - Examine the two bytes at the instruction address because CP does not verify that the location specified contains a valid CPU instruction.
 - Use the TRACE command with the INSTRUCT, BRANCH, or ALL operands if any traced instruction is located at the ADSTOP address.
- Address stops may not be set in an OS/VS or DOS/VS virtual machine's virtual storage; address stops may be set only in the virtual=real partitions or regions of those virtual machines.

ADSTOP

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.

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.

ATTN

Privilege Class: G

Use the ATTN command to make an attention interruption pending at your virtual console. The format of the ATTN command is:

```
ATTN |
```

Usage Notes

1. The REQUEST command performs the same function as ATTN; the two commands can be used interchangeably.
2. The BEGIN command is not required after you issue ATTN.

Responses

None.

BEGIN

BEGIN

Privilege Class: G

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

```
Begin | [hexloc]
```

where:

hexloc is the hexadecimal storage location where execution is to begin.

Usage Notes

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

Responses

None. The virtual machine begins execution.

CHANGEPrivilege Class: G

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

Change	Reader Printer Punch	{ Class c1 spoolid ALL }	{ Class c2 COPY nn [HOLD] [NOHOLD] DIST distcode }	NAME { fm [ft] } { dsname }	} ¹	}

¹One of these operands must be chosen; however, more than one may be specified. They may be combined in any order, except for NAME, which if specified, must be the last entry in the command line. This is contrary to the notation normally used in this publication.

where:

READER changes the reader spool file.
RDR

PRINTER changes the printer spool file.
PRT

PUNCH changes the punch spool file.
PCH

CLASS c1 designates an existing class. The class, c1, is a one-character alphanumeric field from A to Z or from 0-9. Refer to the VM/370: Operator's Guide for a detailed description of spool classes.

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

ALL changes all your spool files.

CLASS c2 changes the spool class of the file to c2.

COPY nn specifies the number of copies of the file to be spooled to the virtual output device. This option is valid for printer and punch files only. The value of nn (number of copies) must be a number from 1 through 99. For nn less than 10, the leading zero is optional.

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

NOHOLD releases the specified file from user HOLD status.

DIST distcode changes the distribution code specified in the VM/370 directory to the distcode specified on the command line, for

CHANGE

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.

NAME fn [ft]

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

NAME dsname

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

Usage Notes

1. Issue the QUERY command to determine the current attributes of the file.
2. In order to change an output file, the file must have been closed but not yet selected for printing or punching.
3. An input (READER) file can be changed at any time before it is opened, that is, before the first read is issued for the file.
4. Use the QUERY command to verify the new attributes of the file or files.
5. If the spoolid operand is used and if more than one spool file with that number exists, then only the first spool file encountered will be changed.

Response

```
{ nnnn } FILES CHANGED  
{ NO }
```

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

CLOSE

cannot be read by the virtual machine until it is changed to NOHOLD by the receiving virtual machine user.

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

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

DIST distcode uses the 1- to 8-character alphameric identification (distcode) on the output separators of printer and punch instead of the identification specified in the VM/370 directory. The distribution code is changed for this file only and does not affect other files or change the VM/370 directory. If the file is transferred to another user, this option has no effect.

NAME fn [ft] assigns identification to the spool file in the CMS format filename and filetype. fn is a 1- to 8-character alphameric filename assigned to the file for identification. ft is a 1- to 8-character alphameric filetype assigned to the file for identification. If ft is not specified, the filetype is set to blanks.

NAME dsname assigns identification to the spool file in a non-CMS format. The dsname field is a 1- to 24-character field suitable for specifying OS or DOS files (for example, SYS1.SYSLIB.MYMAC). Only 20 characters of the 24-character dsname are displayed by QUERY, even though a name of up to 24 characters is valid.

Usage Notes

1. If the file is an input reader file, the file being processed is purged unless SPOOL READER HOLD was previously specified (see the SPOOL command). The effect of HOLD or NOHOLD for a particular file established by the SPOOL command can be overridden by specifying NOHOLD or HOLD, respectively, in the CLOSE command.
2. If the file is an output file on a printer, punch, or console, the file is either queued for output on a real unit record device, or, if the virtual output device is transferred (by use of the "SPOOL vaddr T0 userid" command), the file is queued for input to the receiving user.
3. You can specify a filename and filetype and an optional distribution code to aid in later identification of the file and its contents.
4. Figure 6 depicts the status of the file in your virtual device, depending on the previously stipulated SPOOL command options in effect when the CLOSE command is invoked. The CLOSE command can result in a file being held, saved, or purged.

Virtual Device Status	SPOOL Command Options Set for a Virtual Device			
(CLOSE Command Setting)	NOHOLD NOCONT	HOLD NOCONT	NOHOLD CONT	HOLD CONT
Normal EOF (default CLOSE)	File released for proces- sing	File saved	File released for proces- sing	File held
CLOSE	File released for proces- sing	File saved	File released for proces- sing	File held
CLOSE HOLD	File saved	File saved	File held	File held
CLOSE NOHOLD	File released for proces- sing	File released for proces- sing	File released for proces- sing	File released for proces- sing

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

Figure 6. Spool Command Options Set for a Virtual Device.

Responses

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

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

COUPLE

COUPLE

Privilege Class: G

Use the COUPLE command to connect your virtual (non-dedicated) channel-to-channel adapter (CTCA) to another user's virtual channel-to-channel adapter (or to another one of your own virtual channel-to-channel adapters). The format of the COUPLE command is:

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

where:

vaddr1 is the virtual address (cuu) of your channel-to-channel adapter.

[TO] userid is the user identification of the virtual machine to which vaddr1 is to be connected. If vaddr1 is to be connected to your own virtual machine, userid may be specified as an asterisk (*). The user must be logged on and have a virtual channel-to-channel adapter 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 adapter to be connected to vaddr1.

Responses

CTCA vaddr1 COUPLE TO userid vaddr2

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

vaddr1 is the address of your channel-to-channel adapter.

userid is the identification of the receiving virtual machine.

vaddr2 is the address of the channel-to-channel adapter of the receiving user (or a different channel-to-channel adapter in your own virtual machine).

CTCA vaddr2 COUPLE BY userid vaddr1

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

vaddr2 is the address of the channel-to-channel adapter of the receiving user.

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

vaddr1 is the address of the issuing user's channel-to-channel adapter.

CTCA vaddr1 DROP FROM userid vaddr

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

CTCA vaddr1 COUPLE TO userid vaddr2

CP

CP

Privilege Class: Any

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

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

where:

commandline specifies the name and operands for the CP command or commands you want to issue. You must precede the first 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. The CP command is useful because it lets the CMS user enter commands without having to know which command environment (CP or virtual machine) he is in.
3. You can enter multiple command lines as operands of the CP command, but you must separate each command line by the logical line end (#) symbol. If you enter only CP with no operands, the virtual machine enters the CP environment. CP cancels the virtual machine's console READ by returning a unit exception status for the virtual console. The virtual operating system then reissues the console READ to allow you to key in the appropriate response to a previous message from that machine's operating system.

Responses

If you enter CP without a commandline, you receive this message:

```
CP
```

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

DEFINE

Privilege Class: G

Use the DEFINE command to alter your virtual machine configuration or channel operating mode. The format of the DEFINE command is:

DEFINE	Reader Printer PUnch [AS] vaddr CONsole CTCa TImEr 1403 1443 2501 2540P 2540R 3211 3505 3525 CHANnels [AS] { SEL } { BMX } LIne [AS] vaddr [IBM[1]] [TELE[2]] GRAF cuu [3270] [3158] vaddr1 [AS] vaddr2 T2305 T2314 T2319 T3330 [AS] vaddr [CYL] nnn T3340 T3350 STORAge [AS] { nnnnnK } { nnM }
--------	--

where:

- READER [AS] vaddr adds a spooling 2540 card reader with the address specified by vaddr to the virtual machine configuration.
 RDR
- PRINTER [AS] vaddr adds a spooling 1403 printer with the address specified by vaddr to the virtual machine configuration.
 PRT
- PUNCH [AS] vaddr adds a spooling 2540 card punch with the address specified by vaddr to the virtual machine configuration.
 PCH
- CONSOLE [AS] vaddr adds a virtual system console to the virtual machine at the address specified by vaddr.

DEFINE

CTCA [AS] vaddr adds a virtual channel-to-channel adapter with the address specified by vaddr to the virtual machine configuration. The control unit address must end in zero, and must not already be in use. Once the control unit is defined, other virtual devices may not be defined for the same CTCA.

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.

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.

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.

CHANNELS [AS] { SEL }
 { BMX } redefines the channel mode of operation for the virtual machine to either selector or block multiplexer. Use of the SEL (selector channel) or BMX (block multiplexer channel) operand sets the mode of operation for all channels except virtual channel 0. Channel 0 always operates in byte multiplexer mode. The real or virtual channel-to-channel adapter always operates in selector mode.

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

Note: The virtual machine is immediately reset when this set of operands is executed.


```
LINE [AS] vaddr [ IBM1 ]
                [ TELE2 ]
```

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

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

```
GRAF cuu [ 3270 ]
          [ 3158 ]
```

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

```
vaddr1 [AS] vaddr2
```

redefines the device represented by vaddr1 as vaddr2. The virtual address, vaddr1, must represent a defined device in the virtual machine configuration.

If vaddr1 is a dedicated 2703, both vaddr1 and vaddr2 must be specified as base addresses (xx0 or xx8). Also, when you redefine a dedicated 2305, all 8 exposure addresses are redefined. For example, if 1b0 is the virtual address of a dedicated 2705, the command

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

redefines addresses 1d0 through 1d7 as 1b0 through 1b7, respectively.

```
T2305 [AS] vaddr [CYL] nnn
```

```
T2314
T2319
T3330
T3340
T3350
```

adds a temporary virtual disk of the specified type to the virtual machine configuration. The vaddr specifies the address of the disk, and must not be on a virtual control unit already defined as a CTCA. CYL nnn specifies the number of cylinders that the disk contains. Specify T3350 if a 3350 is used in native mode; specify T3330 if a 3350 is used in 3330 compatibility mode.

Note: Temporary disk space is assigned from a pool of DASD resources. You should always format your temporary disk space before you use it. For the same reason, you should always clear temporary disk space before you either release it or log off. Uncleared data on your detached T-disk could be accessed by another user assigned that same area in a subsequent DEFINE operation.

```
STORAGE [AS] { nnnnnK }
              { nnM }
```

redefines the size of the virtual storage for the virtual machine as nnnnnK (where K represents 1024 bytes) or nnM (where M represents 1,048,576 bytes). The value specified becomes the new virtual storage

DEFINE

size. Sizes must be in 4K increments and are limited by the maximum value in the VM/370 directory entry. The minimum size you can specify is 8K. All entries not specified in a 4K increment are rounded up to the next 4K boundary. Changing the virtual storage size (increasing or decreasing) causes a virtual system reset and clears all virtual storage to binary zeros.

Usage Notes

1. When you alter the configuration or channel operating mode of your virtual machine via the DEFINE command, the changes are temporary and are in effect for the current terminal session only.
2. If either storage or channel operating mode is altered, the virtual machine is immediately reset and the IPL procedure must be performed again.
3. If storage is redefined, any pages that have been locked via the LOCK command become unlocked.
4. If your virtual machine is operating in the Basic Control Mode (ECMODE OFF), virtual addresses specified in vaddr, vaddr1, and vaddr2 can be no higher than 5FF (channels 0-5). If operating in Extended Control Mode (ECMODE ON), all addresses through FFF are valid.

Responses

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

type vaddr DEFINED

where the possible values for type have the following meanings:

<u>Type</u>	<u>Meaning</u>
DASD	Direct access storage device
TAPE	Magnetic tape
LINE	Communication line
RDR	Card reader
PRT	Line printer
PUN	Card punch
GRAF	Graphics device
CONS	Console
CTCA	Channel-to-channel adapter

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) and any channel that has a virtual or real channel-to-channel adapter (always a selector channel).

STORAGE = nnnnnK

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

DETACH

DETACH

Privilege Class: G

Use the DETACH command to remove a virtual device from the virtual machine. The format of the DETACH command is:

```
DETach | vaddr
```

where:

vaddr is the virtual address (cuu) of the device to be detached from your virtual machine. If the device being detached is a dedicated 2305, the virtual address you specify must be a base address (xx0 or xx8).

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/370 directory entries or during CP system initialization.
2. When you detach a virtual device, it becomes inaccessible to your virtual machine. If the device was previously attached to your virtual machine by an ATTACH command, it is released and becomes available for attachment to your virtual machine, another user, or the CP system. Tape devices are automatically rewound and unloaded when detached.
3. If you detach a device that was previously attached to your virtual machine by the operator, a message is sent to the operator informing him that the device is free.

Responses

Messages are sent to you, the operator who issued the command, and the primary system operator (if different from the operator who issued the command), to indicate that the DETACH was successful.

```
type vaddr DETACHED
```

This is the response you receive when you detach one of your own devices. You do not receive this response if you have issued the CP SET IMSG OFF command.

```
type vaddr DETACHED BY operator
```

This is the response you receive if an operator detaches one of your devices.

```
type raddr DETACHED [userid]
                    [SYSTEM]
                    [
```

This is the response sent to the operator if you detach a previously attached device, or if the operator detaches it from a user or the system.

```
type raddr DETACHED [userid] BY operator
                    [SYSTEM]
                    [
```

This is the response sent to the primary system operator if he did not issue the DETACH command and the device had been previously attached.

```
CTCA vaddr DROP FROM userid vaddr
```

This is the response if the device detached was a virtual CTCA connected (via the COUPLE command) to another CTCA on the virtual machine specified by the userid. This response is always followed by the response:

```
CTCA vaddr DETACHED
```

unless the SET IMSG OFF command was issued.

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

<u>Type</u>	<u>Meaning</u>
DASD	Direct access storage device
TAPE	Magnetic tape
LINE	Communication line
RDR	Card reader
PRT	Line printer
PUN	Card punch
GRAF	Graphics device
CONS	Console
CTCA	Channel-to-channel adapter
DEV	Any other device

DETACH CHANNEL

DETACH CHANNEL

Privilege Class: G

Use the DETACH CHANNEL command to detach a dedicated channel from your virtual machine when that channel path is no longer required for virtual machine I/O operations. The format of the DETACH CHANNEL command is:

```
DETach | CHANnel c
```

where:

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

Usage Notes

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

Response

CHANNEL c DETACHED userid

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

DIAL

Privilege Class: Any

Use the DIAL command to logically connect a switched line, leased line, or locally attached terminal to a previously logged-on multiple-access virtual machine. The format of the DIAL command is:

```

|-----|
| DIAL      |   userid [vaddr]
|-----|

```

where:

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

vaddr is the address of the virtual communication line to which the connection is to be made.

Usage Notes

1. A DIAL command is accepted only at logon, and only as a substitute for a LOGON command. The type of terminal used must be supported by both VM/370 and the multiple-access virtual machine. See the VM/370: Terminal User's Guide for details on running and gaining access to multiple-access machines.
2. If the DIAL command is issued from a real 3277 terminal, the virtual system must use the CP command RESET to drop the dialed connection.
3. The DIAL command is not supported for the 3066 system console or for terminals that are using NCP lines in a 3704/3705 control unit.
4. 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.
5. The DIAL command is not supported for remote 3270 terminals.

Responses

DIALED TO userid vaddr

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

DIAL

```
{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/370 lines (nnn) currently connected to other virtual machines. DEV resid indicates the resource identification of a 3704/3705 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.

DISCONN

Privilege Class: Any

Use the DISCONN command to disconnect your terminal from the VM/370 system while the virtual machine continues operation. The format of the DISCONN command is:

```

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

```

where:

HOLD specifies that the communication line is not to be disabled. This option allows you to disconnect your terminal, and, at the same time, to avoid the process of telephone dialing into the system to access your virtual machine again. If specified, control returns to CP and the "VM/370 online" message is displayed.

Usage Notes

1. The terminal remains disconnected until it is reconnected via a LOGON command. The virtual machine is logged off 15 minutes after an attempt is made to read from the terminal or if the virtual machine goes into a disabled WAIT state.
2. If your terminal connection is broken because of terminal, line, or TP 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. If you do not log on within the 15-minute interval, the virtual machine is logged off.
3. Unless the CP command, SPOOL CONSOLE START, is issued to spool the virtual console output, all "writes" or output messages to the virtual console are ignored. When the terminal is reconnected via the normal logon procedure, the terminal is placed in CP console function mode. To resume execution of the virtual machine, enter the BEGIN command.

Responses

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

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

is the response to the user who issued the command.

DISCONN

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

is the response to the primary system operator informing him that the user represented by "userid" has been disconnected from the VM/370 system. The "nnn" is the total number of users remaining in the system. DEV resid indicates the resource identification of a 3704/3705 line.

DISPLAY

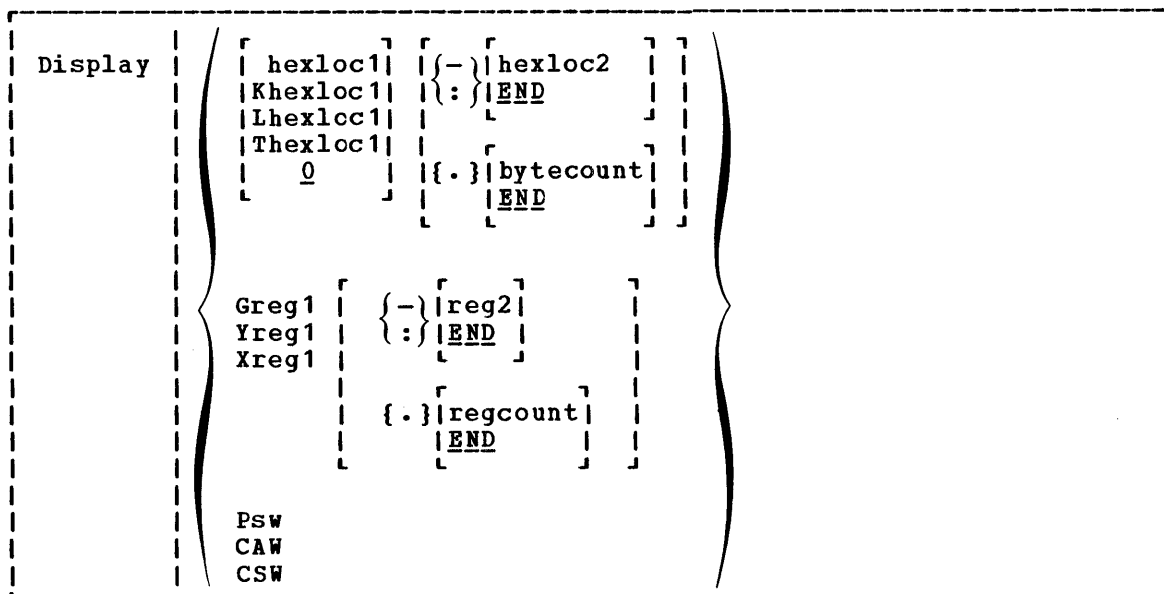
Privilege Class: G

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

- Virtual storage locations (1st 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 storage.

The format of the DISPLAY command is:



where:

hexloc1	is the first, or only, hexadecimal storage location
Lhexloc1	that is to be displayed at the terminal. If
Thexloc1	L or no letter prefix is specified, the storage
Khexloc1	contents are displayed in hexadecimal. If T is
0	specified, the storage contents are displayed in
	hexadecimal, with EBCDIC translation. 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 hexloc1 is not specified, the display begins at storage location 0. If L, T, or K are entered either

DISPLAY

without any operands, or followed immediately by a blank, the contents of all storage locations or all the storage keys are displayed. If L, T, or K are not specified and this is the first operand, then the default value of zero is assumed. The address, hexloc1, may be one to six hexadecimal digits; leading zeros are optional.

{-}hexloc2
{:}END

is the last of the range of hexadecimal storage locations whose contents are to be displayed at the terminal. Either a - or a : must be specified to display the contents of more than one location by storage address. If 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 below for a discussion on discontinuous shared segments.) The address, hexloc2, may be from one to six hexadecimal digits; leading zeros are optional.

{.}bytecount
 END

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

Greg1

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

Yreg1

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

Xreg1

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

{-}reg2
{:}END

is a number representing the last register whose contents are to be displayed at the terminal. Either a - or a : must be specified to display the contents of

more than one register by register number. If reg2 is not specified, the contents of all registers from reg1 through the last register of this type are displayed. The operand, reg2, must be equal to or greater than reg1. If Greg1 or Xreg1 are specified, reg2 may be a decimal number from 0-15 or a hexadecimal number from 0-F. If Yreg1 is specified, reg2 may be 0, 2, 4, or 6. The contents of registers reg1 through reg2 are displayed at the terminal.

{.}regcount <u>END</u>	is a decimal number from 1 to 16 or a hexadecimal number from 1 to F specifying the number of registers (starting with reg1) whose contents are to be displayed at the terminal. If the display type G or X is specified, regcount can be a decimal number from 1 to 16 or a hexadecimal number from 1 to F. If display type Y is specified, regcount must be 1, 2, 3, or 4. The sum of reg1 and regcount must be a number that does not exceed the maximum register number for the type of registers being displayed.
PSW	displays the current virtual machine PSW (program status word) as two hexadecimal words.
CAW	displays the contents of the CAW (channel address word at hexadecimal location 48) as one hexadecimal word.
CSW	displays the contents of the CSW (channel status word at hexadecimal location 40) as two hexadecimal words.

Usage Notes

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

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

DISPLAY

displays the following, respectively:

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

4. To terminate the DISPLAY function while data is being displayed at the terminal, press the Attention key (or its equivalent). When the display terminates, another command may be entered.
5. The DISPLAY command does not distinguish between shared and non-shared storage; it displays any of the virtual machine's addressable storage whether shared or not.
6. Use the DISPLAY command to display the contents of various storage locations, registers, and control words at the terminal. By examining this type of information during the program's execution, you may be able to determine the cause of program errors. Usually, an address stop is set to stop the program execution at a specified point. The system enters the CP environment and you may then issue the DISPLAY command.
7. When you must examine large portions of storage, use the DUMP command rather than the DISPLAY command. Because the terminal operates at a much slower speed than the printer, only limited amounts of storage should be printed (via the DISPLAY command) at the terminal.
8. When running with a discontinuous saved segment (DCSS), you can display storage locations outside the range of your virtual machine size if they are within the DCSS. If there exist locations between the upper limit of your virtual machine and the address at which the DCSS was saved, an attempt to display those locations (or associated keys) will result in a "non-addressable 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 are represented by nonprintable characters. Multiple lines are used (if required) for a range of locations. If translation to EBCDIC is requested (Thexloc), alignment is made to the next lower 16-byte boundary; otherwise, alignment is made to the next lower

fullword boundary. If the location is at a 2K page boundary, the key for that page is also displayed.

Displaying Storage Keys

xxxxxx TO xxxxxx KEY = kk

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

Displaying General Registers

GPR n = genreg1 genreg2 genreg3 genreg4

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

Displaying Floating-Point Registers

FPR n = xxxxxxxxxxxxxxxx .xxxxxxxxxxxxxxxxxx E xx

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

Displaying Control Registers

ECR n = ctlreg1 ctlreg2 ctlreg3 ctlreg4

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

Displaying the PSW

PSW = xxxxxxxx xxxxxxxx

The contents of the PSW are displayed in hexadecimal.

DISPLAY

Displaying the CAW

CAW = xxxxxxxx

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

Displaying the CSW

CSW = xxxxxxxx xxxxxxxx

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

DUMP

Privilege Class: G

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

- Virtual program status word (PSW)
- General registers
- Floating-point registers
- Control registers (if you have the ECMODE option specified in your VM/370 directory entry)
- Storage keys
- Virtual storage locations (1st level virtual storage only; see Usage Notes.)

Note: Use the NETWORK DUMP command to dump the contents of 3704/3705 storage. This command is described in the VM/370: Operator's Guide.

The format of the DUMP command is:

```

DUMP { Lhexloc1 | Thexloc1 | hexloc1 | 0 | { - } hexloc2 | : } END [ *dumpid ]
      { . } bytecount | END

```

where:

Lhexloc1 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.

0

The address, hexloc1, may be one to six hexadecimal digits; leading zeros are optional. If hexloc1 is not specified, the dump begins at storage location 0.

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

{ - } hexloc2
 { : } END

is the last hexadecimal storage location whose contents are to be dumped to the printer. 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.

DUMP

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

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

Usage Notes

1. Only first level storage (storage that is real to the virtual machine) can be dumped. Operating systems such as DOS/VS and OS/VS have virtual storage of their own. This second level virtual storage cannot be dumped directly. The user or the virtual operating system is responsible for converting any second level storage locations to first 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. 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

4. 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/370 directory entry, the control registers are also printed. To print the dump on the real printer, a CLOSE command must be issued for the spooled virtual printer.
5. Normally, you should define beginning and ending dump locations in the following manner:

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

If, however, a blank follows the type character (L or T) or the character and the hexloc, the default dump starting and ending locations are assumed to be the beginning and/or end of virtual storage. Blanks are used to separate operands or sets of operands if more than one operand is entered on the same command line.

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 - . :

```

6. When running with a discontinuous saved segment (DCSS), you can dump storage locations outside the range of your virtual machine size if they are within the DCSS. If there exist locations between the upper limit of your virtual machine and the address at which the DCSS was saved, an attempt to dump those locations (or associated keys) will result in a "non-addressable storage" message appearing in the printer output.

Responses

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

DUMPING LOC hexloc

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

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

COMMAND COMPLETE

is the response indicating normal completion of the dump function.

ECHO

ECHO

Privilege Class: G

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

```
Echo | [ nn ]  
      | [ 1 ]
```

where:

nn is the number of times the line is to be sent. The default is 1. An invalid entry (that is, one that is greater than 99 or contains non-numeric 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 (or its equivalent).
3. No line editing is done; thus, the output line is the same as the input line and may contain any of the logical line editing symbols.

Responses

ECHO ENTERED; TO TERMINATE TEST, TYPE END

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

ENTER LINE

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

EXTERNALPrivilege Class: G

Use the EXTERNAL command to simulate an external interruption to the virtual machine and return control to that machine. The format of the EXTERNAL command is:

```

External | [code]
          | [ 40 ]
          | [   ]
  
```

where:

code is the interruption code, a hexadecimal number to be associated with the external interruption. All codes less than or equal to X'FF' are valid. If your virtual machine has the ECMODE option specified in the VM/370 directory, you can also code 1004 (clock comparator interrupt) or 1005 (CPU Timer interrupt). The default is the External Interrupt Button on the system console, X'40'.

Usage Notes

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

Responses

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

INDICATE

INDICATE

Privilege Class: G

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

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

where:

LOAD displays CPU use, CPU contention, main storage use, and main storage contention.

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

Usage Notes

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

Response for INDICATE LOAD

CPU-nnn% Q1-nn Q2-nn STORAGE-nnn% RATIO-n.n

where:

CPU-nnn% represents the percentage of time that the VM/370 system is running, that is, not waiting.

Q1-nn Q2-nn represent the contention for the CPU in terms of the average number of active users both interactive (Q1) and non-interactive (Q2).

STORAGE-nnn% is a measure of real storage use. It represents an average ratio of the combined working sets of the active users to the total number of pageable pages in the system. It is possible for nnn to exceed 100. (This is a function of system load and VM/370's user scheduling algorithm).

RATIO-n.n represents the contention for real storage. It is defined as:

$$\text{RATIO} = \frac{E+M}{M}$$

where:

E is the number of users that are eligible to become active when enough storage becomes available.

M is the number of users currently active.

Thus, RATIO is the ratio of potential users to users currently active. It is 1.0 for optimum response. Optimum response occurs when enough storage exists to accommodate all potential users, assuming that the system at this time is not CPU bound.

If E and M are both zero, the value of RATIO is set to 1.0.

Given the values of RATIO and M ($M = Q1 + Q2$), the number of users waiting to become active can be calculated as:

$$E = M (\text{RATIO} - 1)$$

Response for INDICATE USER

PAGES: RES-nnnn WS-nnnn READS=nnnnnn WRITES=nnnnnn DISK-nnnn DRUM-nnnn
VTIME=mm:ss TTIME=mm: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.

DISK-nnnn is the current number of virtual pages allocated for you on the system paging disk(s). This number is taken at an instant of time during the execution of the INDICATE command.

DRUM-nnnn is the current number of virtual pages allocated for you on the system paging drum(s). This number is taken at an instant of time during the execution of the INDICATE command.

INDICATE

VTIME=mm:ss is your total virtual machine time since you logged on or since the last ACNT command was issued for your virtual machine.

TTIME=mm:ss is your total virtual machine time plus total CPU 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 non-spooled I/O requests that you have issued since you logged on or since the last ACNT command was issued for your virtual machine.

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

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.

IPL

Privilege Class: G

Use the IPL command to simulate an initial program load function for a virtual machine. The format of the IPL command is:

```

Ipl | { vaddr [cylno] [Clear] [NOClear] [STOP] } [PARM {p1 p2...}]
    | { systemname }

```

where:

```

vaddr [cylno] [CLEAR] | [STOP]
           [NOCLEAR]

```

simulates the IPL function when loading by device address.

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

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

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. 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.

systemname simulates the IPL function when loading a named system that was previously saved via the SAVESYS command. It is loaded into virtual storage and given control. For more information about saved systems, see the VM/370: System Programmer's Guide.

IPL

Note: You cannot load a saved system into a virtual-real virtual machine. An attempt to do so results in an error message.

PARM {p1 p2... }

passes up to 64 bytes of data (including embedded blanks) to your virtual machine's general registers (4 bytes per register), starting with the high order byte of general register 0. Whenever PARM is specified, the remaining characters in the command line are treated as parameters to be passed to your virtual machine; therefore, PARM must be the last operand entered on the command line.

Usage Notes

1. IPL simulates the LOAD button and the device address switches on the real computer console. The specified virtual address is accessed and the required input/output operations are performed to retrieve the IPL data.
2. Optionally, the IPL procedure can be stopped just before loading the virtual PSW except when initial program loading a named system. Also, parameters can be passed to the virtual machine's general registers. When the simulated load function is complete, CP initiates execution of the virtual machine by loading the IPL PSW which was stored during the simulation process.
3. Care must be used when passing parameters to a named system (systemname). Named systems expect certain registers to be initialized when they are given control. Indiscriminate use of the PARM option could overlay a previously initialized register causing unpredictable results.
4. The IPL simulator program occupies one page (4096 bytes) of storage in your virtual machine. Therefore, if the system being loaded utilizes data remaining in storage by the previous system, care should be taken as to where that data is located. The starting address of the overlaid virtual storage is:

virtual machine size	or	hexadecimal 20000

2		

whichever is smaller.

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

Responses

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

LINK

Privilege Class: G

Use the LINK command to make a device that is associated with another virtual machine, available to your virtual machine configuration, based upon information in that user's VM/370 directory entry. The format of the LINK command is:

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

where:

[TO] userid is the name of the user whose VM/370 directory is to be searched for device vaddr1. An asterisk (*) is used to specify that the device is in your own VM/370 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/370 directory for that userid.

[AS] vaddr2 is the virtual address (cuu) which 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.

mode is the access mode; the primary access requested (read-only, write, or multiple), and the alternate access (read-only or write) desired if the primary access is not available. Valid modes are:

Mode Meaning

R Read-only access. The link is not done if any other user has the disk in write status. R is the default mode if the link is to another userid.

RR Read-only access. The link is established even if another user has the disk in write status.

W Write access. The link is not done if any other user has the disk in read or write status.

WR Write and read access. If another user has the disk in read or write status, an alternate access of read-only is acceptable.

M Multiple access. This means that a write-link is to be given to the disk unless another user already has write access to it, in which case no link is to be done.

MR Write-link. If another user already has write access to the disk, a read-link is to be done.

Note: Unpredictable results can occur when one user has a R/O link to a device that is being updated by a user who has the device in R/W status.

MW Write-link. This link is established in all cases.

LINK

Caution: More than one user writing to the same virtual device can result in permanent loss of data.

If the mode is omitted, the default is R if the userid is another user. If you are linking to one of your own disks, the default is the "user access mode" as specified in the VM/370 directory for your disks.

PASS= password is a one- to eight-character string that must match the access mode password for device vaddr1 in the VM/370 directory for the user (userid) specified. The password should be specified only when the LINK is executed by a virtual machine (for example, from CMS), since the password is not print suppressed when included with the LINK command. The 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.

Note: The access mode password should not be confused with a user password.

Usage Notes

1. If you link to one of your own disks, no password is required. Also, if the link is to a device whose password is ALL, meaning that the device can be used by all users, the password is not required. However, if the link is to any other userid, a password for the desired device must be provided.
2. The access allowed by the LINK command to the vaddr1 device belonging to userid is summarized below. You read the columns down to determine the type of link that results. The first row indicates the primary (and, optionally, the alternate) access mode requested. The second row indicates whether read, write, or multiple passwords exist in the VM/370 directory for the disk being linked. The third row indicates whether the disk is already being used, and if so, the mode of its access. The last row indicates the type of link established. For example, the third column is interpreted as follows: if you request a read access link (R) to a disk that has a read password defined and that already is accessed in read mode, you can establish a read link.

Primary access requested:	R R R R R	W W W W W W	M M M M M M
Alternate access (if any):	R	R R	R W
Read password in directory:	N Y Y Y Y		
Write password in directory:		N Y Y Y Y Y	
Mult. password in directory:			N Y Y Y Y Y
Any existing links:	N R W W	N R R W W	N R W W W
Access established:	N R R N R	N W N R N R	N W W N R W
<u>where:</u> N=no or none; R=read; W=write; M=multiple; Y=yes			

Responses

ENTER READ PASSWORD:

XXXXXXXXXX

Type the read password over the mask to obtain read access to the desired disk.

ENTER WRITE PASSWORD:

XXXXXXXXXX

Type the write password over the mask to obtain write access to the desired disk.

ENTER MULT PASSWORD:

XXXXXXXXXX

Type the multiple password over the mask to obtain write access to a disk for which other users may already have access.

Note: If LINK is issued from a virtual machine with the password included on a command line, and the password is incorrect, then CP counts these incorrect passwords. If a total of ten such incorrect passwords is entered, the LINK command from a virtual machine is subsequently disallowed for that user for the remainder of the session. LINK can still be issued directly from the terminal (that is, in CP command mode), or the LINK command can be reinstated as a valid command from your virtual machine by logging off and logging on again. (This procedure is designed to protect password security if a virtual machine issues the LINK command repeatedly with trial passwords.)

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; R/O BY { nnn USERS }l
                             { userid }r           { userid }l
```

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

LINK

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 { nn 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.

LOADVFCBPrivilege Class: G

Use the LOADVFCB command to specify the forms control buffer image for a virtual spooled 3211 printer. The format of the LOADVFCB command is:

```
LCADVFCB | vaddr   FCB name [Index [nn]]
```

where:

vaddr is the virtual device address (cuu) of the virtual spooled 3211 printer.

FCB is a required reserved keyword meaning Forms Control Buffer.

name is a system-defined name for the 3211 FCB image which is to be the controlling virtual FCB image.

INDEX [nn] 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/370: System Programmer's Guide for a discussion of the FCB macro and forms control images.

Usage Notes

1. The LOADVFCB command can be used with installations that do not have a 3211 printer. The virtual machine's VM/370 directory entry must indicate a 3211, even though the program and operating system have a 1403 printer defined. Then the LOADVFCB command can be used to specify a virtual FCB image for 1403 printers so that programs that use printer overflow sensing may be spooled to disk.
2. VM/370 provides one FCB image called FCB1 with the following format:

Line spacing = 6 lines/inch													
Length of page = 66 lines													
Page line	1	3	5	7	9	11	13	15	19	21	23	64	
Channel skip	1	2	3	4	5	6	7	8	10	11	12	9	

Responses

None.

LOGOFF

LOGOFF

Privilege Class: Any

Use the LOGOFF command to terminate virtual machine execution and disconnect your virtual machine from the VM/370 system. The format of the LOGOFF command is:

```
LOGoff | [HOLD]
LOGout |
```

where:

HOLD retains the connection for a switched communication line to enable you to log on without redialing the VM/370 system.

Usage Notes

1. This command causes all active spool files to be closed, temporary disks to be relinquished, dedicated devices to be detached, and an accounting record to be created for the user.
2. You should always log off of your terminal in addition to turning power off on the terminal. Terminal power off is not synonymous with logoff.

If you turn power off at the terminal instead of logging off, logoff occurs by one of the following methods:

- Typewriter Terminal and Remote 3270 Display Terminal --Logoff takes place after a 15-minute interval has elapsed. This occurs if no attempt is made to power on the terminal and re-establish communications with the still logged-on virtual machine during this 15-minute period.
- Local 3270 Display Terminal --Logoff only takes place 15 minutes after VM/370 discovers that the terminal has been turned off (that is, VM/370 attempts to send a message to the terminal, but gets back an error code indicating that the terminal is turned off). Because many hours may pass before VM/370 discovers that the terminal is turned off, you run the risk of compromising the security of the virtual machine and data files. Anyone turning the 3270 power back on has access to the virtual machine without logging on. This is because the machine is still logged on, although inactive.

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=hh:mm:ss
 the virtual CPU time used in the current terminal session in
 minutes:seconds.hundredths of seconds.

TOTCPU=hh:mm:ss
 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.

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

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

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

is the response to the primary system operator if the logoff is
 forced by a line timeout or a terminal power-off. DEV rid
 specifies the resource identification of a 3704/3705 line.

USER DSC LOGOFF AS userid USERS = nnn

is the response to the primary system operator when logoff occurs
 for a user who had previously disconnected using the DISCONN
 command.

LOGON

LOGON

Privilege Class: Any

Use the LOGON command to identify yourself to the VM/370 system and to access that system. The format of the LOGON command is:

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

where:

- userid** is the identifier assigned to you in the VM/370 system.
- password** is your password. Specify this field if no protection (that is, masking characters) is desired.
- MASK** types masking characters to cover your password on typewriter terminals without the print inhibit feature. The mask types on the line following a prompting message from VM/370 requesting you to enter your password. Should you forget to ask for masking when you type LOGON, you can press the carriage return after the prompt for the password types, and VM/370 then types out the masking characters.
- NOIPL** specifies that the IPL device or name in the VM/370 directory should not be used for an automatic IPL.

Usage Notes

1. Upon successful logon, VM/370 creates a virtual machine configuration from information in the VM/370 directory. The LOGON command name may not be entered using any line-editing symbols, but the operands may use these symbols. See the VM/370: Terminal User's Guide for a detailed description of logon procedures.
2. 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.

Responses

ENTER PASSWORD:

indicates that the userid has been accepted. You should type in the password, or signal a carriage return if a mask is desired for the password, and MASK was not included on the command line.

LOGMSG- hh:mm:ss mm/dd/yy

indicates the time and date at which the system log message was generated or most recently revised. All lines of the log message for which the first character is an asterisk are displayed at this point. If you wish to see all of the system log messages, you must issue the CP command QUERY LOGMSG.

FILES: { nnn } RDR, { nnn } PRT, { nnn } PUN
 { NO } { NO } { NO }

This message is omitted if all counts are zero, otherwise it indicates the number of spool files that exist for you at logon time.

LOGON AT hh:mm:ss zone weekday mm/dd/yy

-- or --

RECONNECTED AT hh:mm:ss zone weekday mm/dd/yy

indicates the time, day of the week, and date at which the LOGON or RECONNECT is complete.

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

-- or --

{ 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 line.

MESSAGE

MESSAGE

Privilege Class: Any

Use the MESSAGE command to transmit message text to a specified userid or to the primary system operator. The format of the MESSAGE command is:

```
Message | {userid } msgtext
Msg     | { *
        | {OPERATOR }
```

where:

userid is the identification of the user who is to receive the message.

* specifies that you are sending a message to yourself.

OPERATOR sends the message to the primary system operator regardless of his userid.

msgtext is the text of the message which is to be transmitted. The length of the message is limited only by the number of characters remaining on the input line.

Usage Notes

1. If the user designated to receive the message is not logged on or has suppressed the receiving of messages (SET MSG OFF), the message is not transmitted and the sender receives a diagnostic message to this effect. A message which is not received by a user is not saved and must be sent at a later time when the user is receiving messages.
2. The message is displayed at the terminal when the terminal is ready to receive output. If a typewriter terminal (or a display terminal having AUTOREAD set ON) is entering data, the message is not displayed until an end-of-line (carriage return or ENTER) signal is received.
3. If the message is received by the primary system operator console and SET MSG ON is in effect, the alarm at the central computer console sounds.
4. If the user receiving the message is logged on with a 3277 virtual console that has the alarm feature, the alarm sounds.
5. MSG * is useful in identifying the current userid of an abandoned terminal, or it can be used in a program intended to be executed under several userids.

Responses

hh:mm:ss
MSG FROM OPERATOR: msgtext

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

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

is the format of the message sent to another user or to the system operator, where userid is that of the sender. If the user sending the message is not logged on to VM/370, 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.

NOTREADY

NOTREADY

Privilege Class: G

Use the NOTREADY command to cause a virtual device to appear as if it had changed from ready to not ready status. The format of the NOTREADY command is:

```
| NOTReady | vaddr |
```

where:

vaddr is the virtual device address (cuu) of the unit to be removed from ready status.

Usage Notes

1. This command is for use with spooled unit record devices and virtual consoles only.
2. Any I/O operation to the specified device, in progress at the time the command is issued, is completed. On the next Start I/O (SIO) instruction, the not ready condition is in effect.

Response

None.

ORDERPrivilege Class: G

Use the ORDER command to place your closed spool files in a specific order by device type. The format of the ORDER command is:

ORDER		{ Reader Printer PUnch }	}	{ Class c1 Class c2... } ¹ { spoolid1 spoolid2... }
-------	--	--------------------------------	---	---

¹A combination of CLASS and spoolid specifications may be entered.

where:

READER RDR specifies that the reader spool files are to be placed in order.

PRINTER PRT specifies that the printer spool files are to be placed in order.

PUNCH PCH specifies that the punch spool files are to be placed in order.

CLASS c1 CLASS c2... are the class(es) to be rearranged and the sequence in which they are to be processed. CLASS is a required reserved word and c1, c2,... are one-character alphameric fields (with values from A to Z and from 0 to 9) that represent spooling classes.

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

Usage Notes

1. You may order only your own files.
2. Use the QUERY command to determine spool file attributes such as class and spoolid for files you want to rearrange.
3. A combination of class and spoolid specifications can be used to rearrange files. For example:

ORDER PRINTER CLASS A 1963 CLASS C

specifies that printer files are processed in the following order: all class A files, the file with spoolid 1963, all class C files, followed by any other printer spool files not specified in the ORDER command line.

ORDER

4. When duplicate spoolids are present, the ORDER command resequences only the first duplicate spool file found.
5. The order of your spool files may change without warning between terminal sessions if the system was restarted with a checkpoint restart.

Response

```
{ nnnn } FILES ORDERED  
{ NO   }
```

indicates the number of files that were placed in order. It is not displayed if you issued the CP SET IMSG OFF command.

PURGEPrivilege Class: G

Use the PURGE command to remove your own closed spool files from the system before they are printed or punched by the spooling devices, or before they are read by a user. The format of the PURGE command is:

```

PURge | ( Reader [Class c1 Class c2... ]1 )
      | ( Printer |spoolid1 spoolid2... | )
      | ( PUnch |ALL | )
      | ( ALL | | )

```

¹A combination of CLASS and spoolid specifications may be entered.

where:

READER purges reader files.
RDR

PRINTER purges printer spool files.
PRT

PUNCH purges punch spool files.
PCH

ALL purges all reader, printer, and punch spool files. When ALL is specified instead of a device type, all other operands are ignored.

CLASS c1 CLASS c2...
are the class(es) of files to be purged. CLASS is a required reserved word and c1, c2,... are one-character alphameric fields (with values from A to Z and 0 to 9) that represent spooling classes.

spoolid1 spoolid2...
are the spoolids of specific files to be purged.

ALL purges all files of the specified type (reader, printer, or punch).

Usage Notes

1. Any closed spool file, owned by you, may be purged, regardless of its status, as long as it has not been selected for processing.
2. Purging can be done using a combination of CLASS and spoolid specifications. For example:

```
PURGE PRINTER CLASS A 1932 CLASS D 619
```

specifies that all class A and class D printer files and printer files with spoolids of 1932 and 619 are to be purged.

PURGE

3. A file that has been spooled to another user's reader cannot be purged unless it is first reclaimed via the TRANSFER command.

Response

```
{ nnnn } FILES PURGED  
{ NO   }
```

indicates the number of files purged. It is not displayed if you issued the CP SET IMSG OFF command.

QUERY

Privilege Class: G and all classes except class Any

Use the class G QUERY command to determine your system status and machine configuration by requesting:

- The time you have used during a terminal session.
- The number of closed input and output spool files associated with your virtual machine.
- 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.
- Identification and attributes associated with your virtual printer, punch, and reader spool files.

There are other operands you can use with the QUERY command if you have the privilege class required to use them. These are described in the VM/370: Operator's Guide. Also, if you are a CMS user, you can use the CMS QUERY command to query the status of your CMS virtual machine.

Note: Use the NETWORK QUERY command to query 3704/3705 resources and remote 3270 devices attached to a 2701 or 2703 or a 3704/3705 in EP or PEP mode.

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

QUERY

QUERY COMMAND FOR CLASS G USERS

The format of the Class G QUERY command is:

Query	Time	
	Set	
	TERMinal	
	Files [Class c]	
	[Virtual]	(CHANNELS)
		GRAF
		CONSOLE
		DASd
		TAPes
		LINES
		UR
		STORAge
		ALL
		vaddr
	Links	vaddr
	Reader	[spoolid]
	Printer	ALL
	PUnch	Class c
	PF[nn]	

where:

- TIME** displays the current time, time zone, weekday, date, connect and CPU time for the current terminal session.
- SET** displays the status of the SET command functions.
- TERMINAL** displays the current options in effect for your virtual console environment.
- FILES [CLASS c]** displays the number of spooled input and output files for your virtual machine. Files currently being processed are not included in the totals. If CLASS is specified, the number of spooled input and output files of the class specified is displayed.
- VIRTUAL** displays the status of all virtual devices.
- CHANNELS** displays the channel mode of operation for the virtual machine.
- GRAF** displays the status of all your virtual display devices that are locally attached.
- CONSOLE** displays the status of your virtual consoles.
- DASD** displays the status of all your virtual direct access storage devices.

TAPES displays the status of all your virtual magnetic tape devices.

LINES displays the status of all your virtual communication lines.

UR displays the status of all your unit record devices.

STORAGE displays the size of your virtual storage.

ALL displays the status of all your virtual devices.

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

LINKS vaddr displays the userid, device address, and access mode at the terminal for all users linked to the specified virtual address (vaddr).

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

RDR

PRINTER

PRT

PUNCH

PCH

- Userid (of user who created the file)
- Spool file identification (spoolid)
- Class and originating device type
- Number of logical records in the file
- Number of copies specified for the file (has no effect for reader files)
- File hold status

One line of information is displayed for each spool file.

{ READER }
 { PRINTER } spoolid
 { PUNCH }

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

- Date and time the file was created
- Filename and filetype of file (if any); if your file was assigned a dsname and you later issue QUERY, only the first 20 characters of the 24-character field are displayed.
- Distribution code of the file (printer and punch files only)

Only one line of data is displayed (that data pertaining to the spool file specified by spoolid).

{ READER }
 { PRINTER } ALL
 { PUNCH }

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

QUERY

- Date and time the file was created
- Filename and filetype of file (if any); if your file was assigned a dsname and you later issue QUERY, only the first 20 characters of the 24-character field are displayed.
- Distribution code of the file (PRINTER and PUNCH files only)

One line of information is displayed for each spool file of the type specified.

{ READER
PRINTER } CLASS c
{ PUNCH }

displays the basic information for all spool files of the class specified by c. This operand must follow the READER, PRINTER, or PUNCH operands. One line of information is displayed for each spool file of the specified class.

PF[nn] displays the 3270 Program Function key number specified, along with its associated command lines. If nn is not specified, all 12 program function keys and their associated data lines are displayed. The value, nn, is a number from 1 (or 01) to 12. See the CP SET command for an explanation of how to define and use program function keys.

QUERY Command Responses for Class G Users

This section describes the messages that CP issues in response to your specific QUERY operands.

QUERY TIME

TIME IS hh:mm:ss zone weekday mm/dd/yy

The current real clock time in hours:minutes:seconds, the time zone (for example, EST), the day of the week and the calendar date (month/day/year) are displayed.

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

The time spent in the current terminal session is displayed.

where:

CONNECT= hh:mm:ss is the actual clock time spent in the current terminal session in hours:minutes:seconds.

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

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

QUERY SET

MSG { ON } , WNG { ON } , EMSG { ON } , ACNT { ON } , RUN { ON }
 { OFF } , { OFF } , { OFF } , { OFF } , { OFF }
 { CODE }
 { TEXT }
 LINEDIT { ON } , TIMER { ON } , ISAM { ON } , ECMODE { ON }
 { OFF } , { OFF } , { OFF } , { OFF }
 { REAL }
 ASSIST { ON } { SVC }
 { OFF } { NOSVC } , PAGEX { ON }
 { OFF }
 IMSG { ON }
 { OFF }

The settings of all functions controlled by the SET command and the VM/370 directory ISAM and ECMODE options are displayed. Refer to the discussion of the SET command for explanations of the functions.

QUERY VIRTUAL CONSOLE

```

CONS vaddr ON { GRAF } raddr { TERM } { STOP }
              { LINE } { NOTERM } { START }

vaddr CL c { CONT } { HOLD } COPY nn { READY }
           { NOCONT } { NOHOLD } { NOTREADY }

vaddr { TO }
      { FOR } userid DIST distcode

```

For virtual machine consoles, a three-line response is displayed. The first line shows the console status and options and the next two 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.

nn is the number of copies spooled.

userid is the user identification.

distcode is the distribution code.

The other fields indicate the setting of the respective options in the SPOOL command.

The default settings for a virtual console are:

```

CONS vaddr ON DEV raddr TERM STOP
      vaddr CL T NOCONT NOHOLD COPY 01 READY
      vaddr FOR userid DIST distcode

```

QUERY VIRTUAL DASD

```

DASD vaddr type volser { R/W }
                       { R/O } nnn CYL

```

The status of each virtual disk defined for your system is displayed.

where:

vaddr is the virtual address to which the DASD device is attached.

QUERY

type is one of the following device types:

| 231T (2311 at top of 2314)
| 231B (2311 at bottom of 2314)
2311
2305
2314
3330
3340
| 3350

volser is the volume serial number of the system disk on which this virtual disk resides.

R/W indicates the read/write status of the disk.
R/O

nnn is the number of cylinders on the virtual disk.

QUERY VIRTUAL TAPES

TAPE vaddr ON DEV raddr

The status of each tape defined for your system is displayed.

where:

vaddr is the virtual address to which the tape is attached.

raddr is the real address of the tape.

QUERY VIRTUAL LINES

LINE vaddr ON DEV raddr

The status of all communication lines defined in your virtual machine is displayed.

where:

vaddr is the virtual address to which the line is attached.

raddr is the real address of the line.

LINE vaddr { ENABLED }
{ DISABLED }

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

QUERY VIRTUAL UR

RDR vaddr CL c { CONT } { HOLD } { EOF } { READY }
{ NOCONT } { NOHOLD } { NOEOF } { NOTREADY }

The status of all the virtual readers attached to your virtual machine is displayed.

where:

vaddr is the virtual device address of the virtual reader.

c is the spool file class which the device services. A class of * indicates the device serves all classes of spool files for input.

The other fields indicate the setting of the respective options in the SPOOL command.

The default settings for a reader are:

RDR vaddr CL * NOCONT NOHOLD READY EOF

```
{ PRT }
{ PUN } vaddr CL c { CONT } { HOLD } COPY nn { READY }
          { NOCONT } { NOHOLD } { NOTREADY }
```

```
vaddr { TO }
      { FOR } userid DIST distcode
```

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

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.

nn is the number of copies of each output file to be produced.

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/370 directory.

The other fields indicate the setting of the respective options in the SPOOL command.

The default settings are:

```
{ PRT }
{ PUN } vaddr CL A NOCONT NOHOLD COPY READY 01
          vaddr FOR userid DIST distcode
```

QUERY

where:

userid and distcode are assigned for the virtual machine.

QUERY VIRTUAL STORAGE

STORAGE = nnnnnK

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

QUERY VIRTUAL ALL

Has the same effect as if all the following commands were issued:

```
QUERY VIRTUAL STORAGE
QUERY VIRTUAL LINES
QUERY VIRTUAL TAPE
QUERY VIRTUAL UR
QUERY VIRTUAL DASD
QUERY VIRTUAL GRAF
QUERY VIRTUAL CONSOLE
QUERY VIRTUAL CHANNELS
```

QUERY VIRTUAL vaddr

The response is in the same form as QUERY VIRTUAL DASD, TAPES, LINES, or UR, depending on virtual device type.

QUERY LINKS vaddr

```
userid vaddr { R/O }, ...
.         . { R/W }
.         .
.         .
```

A list of users who are currently linked to the device at virtual address vaddr is displayed.

where:

userid is the identification of the user who originated the link.

vaddr is the virtual address by which the user (userid) refers to the device.

R/O is the type of access the user (userid) has to the device.
R/W

QUERY READER, QUERY PRINTER, QUERY PUNCH

```

ORIGINID FILE CLASS RECDS CPY HOLD [DATE TIME NAME TYPE DIST ]
userid spoolid c typ norecs nn stat [mm/dd hh:mm:ss fn ft distcode]
: : : : : : : : : : : :
: : : : : : : : : : : :
: : : : : : : : : : : :

```

where:

- userid is the user who originally created the file.
- spoolid is a unique, system-assigned number which is used by VM/370 to identify the file.
- c is the spool file class.
- typ is the originating device type (PRT, PUN, CON, or RDR).
- norecs is the number of logical records contained in the file.
- nn is the number of copies assigned to the file (it has no effect for virtual reader files).
- stat is the file hold status: NONE (no hold), USER (user hold), SYS (system hold), or USYS (system and user hold).
- 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.
- fn is the filename assigned to the file (if any). If the file has a 24-character data set name (dsname), only 20 characters are displayed. These characters extend from the "name" field through the "type" field.
- ft is the filetype assigned to the file (if any).
- distcode is the distribution code assigned to the file.

When you issue QUERY READER, QUERY PRINTER, or QUERY PUNCH commands, CP responds by listing (in the form described) all the 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, name of file, filetype of file, and file distribution code) is displayed only when you specify the ALL or spoolid operands.

QUERY PFnn

```

PFnn {IMMED}
      {DELAY} pfddata1...

```

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

QUERY

QUERY COMMAND FOR ALL CLASSES OF USERS (EXCEPT CLASS ANY)

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

- Display the log messages.
- List all the users that are logged on.
- Display the number of users that are logged on or dialed to VM/370.

The format for this QUERY command is:

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

where:

LOGMSG displays the log messages of the day.

NAMES displays a list of all the users logged on and the real address of the line to which each is connected. If a user is disconnected, DSC is printed instead of the line address.

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

USERS userid displays the user identification and the terminal device address of the specified user if he is logged on. If the user is not logged on, a message to this effect is issued. Use the QUERY USERS userid format if the userid is the same as an operand of the QUERY command (for example, TAPES).

Note: It is possible for the number of users logged on as indicated by the "NAMES" operand to differ from the number of users indicated by the "USERS" operand if the QUERY command is executed while users are in the process of logging on or logging off the system.

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

QUERY LOGMSG

```
* logmsg text line 1
.
.
.
* logmsg text line n
```

All lines (both those with an asterisk and without) in the log message file are displayed.

QUERY NAMES

```
userid - { DSC } , ...
         . { raddr }
         . { resid }
userid - { DSC } , ...
         . { raddr }
         . { resid }
```

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 line (resid) to which he is connected is displayed; if he is not connected to the system, DSC is displayed.

QUERY USERS

nnn USERS, mmm DIALED

The number of users logged on and dialed to VM/370 is displayed.

where:

nnn is the total number of logged-on users.

mmm is the total number of users attached via DIAL to virtual machines.

Note: DIALED means the line is not available to CP because it is logically attached to a multiple-access virtual machine and is a part of that user's virtual machine operation.

QUERY userid or QUERY USERS userid

```
userid - { raddr }
         { resid }
```

The real address (raddr) or the resource identification of a 3704/3705 line (resid) to which the specified user is connected is displayed.

READY

READY

Privilege Class: G

Use the READY command to set a device-end interruption pending for a specified virtual device. The format of the READY command is:

```
READY | vaddr
```

where:

vaddr is a virtual device address (cuu).

Usage Notes

1. The status of the virtual machine is unchanged.
2. Other than having a device-end interruption pending, the virtual device is unchanged.

Responses

None.

REQUEST

Privilege Class: G

Use the REQUEST command to make an attention interruption pending at your virtual console. The format of the REQUEST command is:

```
REQuest |
```

Usage Note

The REQUEST command performs the same functions as ATTN and the two commands can be used interchangeably.

Responses

None.

RESET

RESET

Privilege Class: G

Use the RESET command to clear all pending interruptions from the specified virtual device. The format of the RESET command is:

```
| RESET | vaddr |
```

where:

vaddr is a virtual device address (cuu) of the device to be reset.

Usage Note

In addition to clearing all pending interruptions, all error conditions occurring as a result of unit checks and virtual sense bytes are reset.

Responses

DEVICE RESET

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

REWIND

Privilege Class: G

Use the REWIND command to rewind (but not unload) a real tape unit attached to your virtual machine at a specific virtual device address. The format of the REWIND command is:

```
| REWInd | vaddr |
```

where:

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

Usage Note

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

Responses

REWIND COMPLETE

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

REWIND NOT PERFORMED

is the response you receive if the real tape unit is not ready.

SET

SET

Privilege Class: G

Use the SET command to control various functions within your virtual system. The format of the SET command is:

SET	ACNT	{ ON }
	MSG	{ OFF }
	WNG	
	IMSG	
	RUN	
	LINEDit	
	ECmode	
	ISAM	
	NOTrans	
	PAGEX	
	EMSG	{ ON }
		{ OFF }
		{ CODE }
		{ TEXT }
	TIMER	{ ON }
		{ OFF }
		{ REAL }
	ASsist	{ [ON] [SVC] }
		{ [] [NOSVC] }
		{ OFF }
	PFnn	{ [IMMed] [pfdata 1#pfdata 2#...pfdata n] }
		{ [DELayed] }
	PFnn	[TAB n1 n2 ...]
	PFnn	COPY [resid]
	PFnn	COPY [cuu]

where:

ACNT { ON } 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/370, ACNT is set on.

MSG { ON } controls whether messages sent by the MSG command from other users are to be received at the terminal. If ON is specified, the messages are displayed. If OFF is specified, no messages are received. In addition to controlling messages generated by the MSG command, spooling messages generated by users sending punch,

printer or reader files to another virtual machine are also suppressed if OFF is specified. When you log on VM/370, MSG is set on.

WNG { ON }
 { OFF }

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 VM/370, WNG is set on.

IMSG { ON }
 { OFF }

controls whether certain informational responses issued by the CP CHANGE, DEFINE, DETACH, ORDER, PURGE, and TRANSFER commands are displayed at the terminal or not. The descriptions of these CP commands tell which responses are affected. If ON is specified the informational responses are displayed. If OFF is specified, they are not. The SET IMSG ON or OFF command line has no effect on the handling of error messages set by the SET EMSG command. When you log on VM/370, IMSG is set on.

RUN { ON }
 { OFF }

controls whether the virtual machine stops when the Attention key is pressed. ON allows you to activate the Attention key (causing a read of a CP command) without stopping your virtual machine. When the CP command is entered, it is immediately executed and the virtual machine resumes execution. OFF places the virtual machine in the normal CP environment, so that when the Attention key is pressed, the virtual machine stops. When you log on VM/370, RUN is set off.

LINEDIT { ON }
 { OFF }

controls the line editing functions. ON specifies that the line editing functions and the symbols of the VM/370 system are to be used to edit virtual CPU console input requests. This establishes line editing features in systems that do not normally provide them. OFF specifies that no character or line editing is to be used for the virtual machine operating system. When you log on VM/370, LINEDIT is set on.

ECMODE { ON }
 { OFF }

controls whether the virtual machine operating system may use System/370 extended control mode and control registers 1 through 15. Control register zero may be used with ECMODE either ON or OFF. When you log on VM/370, 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.

ISAM { ON }
 { OFF }

controls whether additional checking is performed on virtual I/O requests to DASD in order to support the OS Indexed Sequential Access Method (ISAM). When you log on VM/370, ISAM is set according to the user's directory options; ON if ISAM was specified and OFF if not.

SET

NOTRANS { ON } controls CCW translation for CP. NOTRANS can be
 { OFF } specified only by a virtual machine that occupies the
virtual=real space. It causes all virtual I/O from the
issuing virtual machine to bypass the CP CCW translation
except under the following conditions:

- SIO tracing active
- 1st CCW not in the V=R region
- I/O operation is a sense command
- I/O device is a dial-up terminal
- I/O is for a non-dedicated device
- Pending device status

Any of the above conditions will force CCW translation.

To be in effect in the virtual=real environment, SET
NOTRANS ON must be issued after the virtual=real machine
is loaded via the IPL command. (IPL sets the NOTRANS
option to an OFF condition.)

PAGEX { ON } controls the pseudo page fault portion of the
 { OFF } VM/VS Handshaking feature. PAGEX ON or OFF should only be
issued for an OS/VS1 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/370, PAGEX is set OFF. Also, each time you
IPL VS1 in 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 IPL
VS1.

EMSG { ON } controls error message handling. ON specifies that both
 { OFF } the error code and text are displayed at the terminal.
 { CODE } TEXT specifies that only text is displayed. CODE
 { TEXT } specifies that only the error code is to be displayed.
OFF specifies that no error message is to be displayed.
When you log on VM/370, EMSG is set to TEXT.

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

Note: CMS 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.

TIMER { ON } controls the virtual timer. ON specifies that the
 { OFF } virtual timer is to be updated only when the virtual CPU
 { REAL } is running. OFF specifies that the virtual timer is not
to be updated. REAL specifies that the virtual timer is
to be updated during virtual CPU run time and also during
virtual wait time. If the REALTIMER option is specified
in your VM/370 directory entry, TIMER is set to REAL when
you log on; otherwise it is set to ON when you log on.

```

ASSIST { [ ON ] [ SVC ]
         [   ] [ NOSVC ]
         [   ] [   ]
         OFF

```

controls the availability of the virtual machine assist feature for your virtual machine. The assist feature is available to your virtual machine when you log on if (1) the real CPU has the feature installed and (2) the system operator has not turned the feature off. The SVC handling portion of the assist feature is invoked when you log on unless your VM/370 directory entry has the SVCOFF option. Issue the QUERY SET command line to see if the assist feature is activated and whether the assist feature or VM/370 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 virtual machine assist feature is turned off, the appropriate bits are set. Later, if the feature is turned on again, the operand you specified while it was off becomes effective. ON sets the assist feature on for the virtual machine; OFF turns it off. SVC specifies that the assist feature handles all SVC interruptions except SVC 76 for the virtual machine; NOSVC means VM/370 handles all the SVC interruptions. See the VM/370: System Programmer's Guide for information on how to use the assist feature.

```

PFnn [ IMMED ] [pfdata1#pfdata2#...pfdataN]
     [ DELAYED ]

```

defines a program function for a program function key on a 3277 Display Station and indicates when that function is to be executed. See the VM/370: Terminal User's Guide for a description of how to use the 3277 program function keys.

nn is a number from 1 (or 01) to 12 that corresponds to a key on a 3277. The program function is a programming capability you create by defining a series of VM/370 commands or data you want executed. This series of commands executes 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.

pfdata1#pfdata2#...pfdataN defines the VM/370 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

SET

line with #CP, turn line editing off, or precede each pound sign with the logical escape character ("). For further explanation, see the "Usage Notes" section that follows. If no command lines are entered, PFnn is a null command. Program functions cannot be embedded within one another.

PFnn TAB n1 n2 ...

specifies a program function 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 12. For examples of how this feature is used, see the VM/370: CMS User's Guide.

TAB is a keyword identifying the tab function. The tab settings (n1 n2 ...) may be entered in any order.

PFnn COPY [resid]

specifies that the program function key, numbered nn, performs a COPY function for a remote 3270 terminal. nn must be a value from 1 (or 01) to 12. The COPY function produces a printed output of the entire screen display at the time the PF key is actuated. The output is printed on an IBM 3284, 3286 or 3288 printer connected to the same control unit as your display terminal.

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

PFnn COPY [cuu]

specifies that the program function key, numbered nn, performs a COPY function for a local 3270 terminal. nn must be a value from 1 (or 01) to 12. When the PF key is actuated, the COPY function produces a printed output of the entire local screen display except for the status field which is replaced with blanks.

cuu is the real hardware address of the 3284, 3286, or 3288 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 will be selected.

Note: For both remote and local COPY functions:

You will receive a NOT ACCEPTED message, displayed in the screen status field of your 3270, if any of the the following situations occur:

- The printer is already busy, or all printers are busy.
- The printer is turned off.
- The printer is out of paper or is in any other intervention required condition.
- The designated device is not a 328X type printer.
- The SET PFnn COPY command 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.

Usage Notes

1. Both SET PFnn TAB and SET PFnn COPY are immediate commands: their function is executed immediately upon pressing the appropriate program function key. If you insert the keyword DELAYED after the PFnn operand, the command will be accepted, however, the program function will still be executed immediately.
2. If you use the SET PFnn command to set up a series of concatenated commands, you should be aware of the following situation:

If you enter one of the following commands while in CMS mode:

```
SET PF02 IMMED Q RDR#Q PRT#Q PUN
```

-- or --

```
CP SET PF02 IMMED Q RDR#Q PRT#Q PUN
```

and then press the Enter key:

1. The Enter key causes immediate execution,
2. Only the Q PRT and Q PUN commands execute, and
3. Q PRT and Q PUN are stripped from the PF02 key assignment leaving Q RDR, which was not executed.

The following examples demonstrate two methods for avoiding the problem.

Example 1

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
```

Now 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.

SET

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.

Responses

None

SLEEP

Privilege Class: Any

Use the SLEEP command to place the virtual machine in a dormant state but allow messages to be displayed. You can specify a sleep interval in the command line and the virtual machine is awakened automatically when the specified interval has elapsed. The format of the SLEEP command is:

```

Sleep      | | [ [ ] ]
           | | [SEC|]
           | | nn |MIN|]
           | | [HRS|]
           | | [ ] ]

```

where:

```

[ [ ] ]
|nn |MIN|]
[ [HRS|]
[ ] ]

```

indicates the number of seconds, minutes, or hours, measured by the time-of-day clock, that the virtual machine is to remain dormant. The value nn can be any decimal number from 00 through 99. If you specify no time unit, the value of nn is taken to be in minutes.

Usage Notes

1. During the dormant period, the virtual machine does not run but connection time is still being counted.
2. The terminal can be awakened at any time by signaling attention.
3. If no interval is specified, the virtual machine remains dormant until awakened by signaling attention.
4. If you issue the SLEEP command from a CP read or from a VM read using the CP "escape" function (#CP SLEEP), the end of the time interval or signaling attention returns you to the CP environment.
5. If you issue the SLEEP command while in virtual machine mode (for example, CMS execution of the command line CP SLEEP), the end of the time interval or signaling attention returns your terminal to virtual machine mode without entering the CP environment.
6. The SLEEP command, with the time interval, is a convenient way to delay or schedule the execution of certain jobs that could be run more efficiently at a later time; for example, second shift.

Responses

None.

SPOOL

Privilege Class: G

Use the SPOOL command to modify the spooling control options in effect for a given virtual spooling device or for a group of devices. The SPOOL command can also start or stop the spooling of virtual console input and output. You can direct a file to a remote location by using the SPOOL command in conjunction with the TAG command. This latter function is discussed below in the section "Transmitting Files to Remote Locations." The format of the SPOOL command is:

```

SPOOL {Reader } { [Class {c}] [CONT ] [HOLD ] [EOF ] } 1
      {vaddr } { [ * ] [NOCont] [NOHold] [NOEof] }

      {Printer } { [To ] [userid ] [HOLD ] [CONT ] } 1
      {PUnch } { [For] [ * ] [NOHold] [NOCont] [Class c] [Copy nn] }
      {vaddr } { [ SYSTEM ] }
              [ OFF ]
              [ CLOSE ]
              [ PURGE ]

      {CONsole } { [STart] [HOLD ] [CONT ] [TErm ] [To ] [userid] } 1
      {vaddr } { [STop ] [NOHold] [NOCont] [NOTerm] [For] [ * ] }
              [Class c] [Copy nn] [CLOSE] [OFF]
              [PURGE] [ SYSTEM ]

```

¹At least one of the options within braces must be selected; however, more than one may be specified, and they may be entered in any order.

where:

READER modifies the options for all reader spool files.
RDR

PRINTER modifies the options for all printer spool files.
PRT

PUNCH modifies the options for all punch spool devices.
PCH

CONSOLE modifies the options for the virtual console spool file and/or initiates or stops the spooling of virtual console input and output, including CP input/output.

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

CLASS { c } specifies the spool class of the device. c is a one-character alphameric field whose values can be A through Z, or 1 through 9.

Unless your virtual reader class is asterisk (*), you must ensure that any files to be read by your virtual reader are of the same spool class as your virtual reader. The * is the universal class; if your virtual reader is class *, it can read any file, regardless of class.

CONT ignores intermediate end-of-file indications or CLOSE requests. For virtual readers, reading is continuous, with all end-of-file indicators ignored until all files spooled to the virtual machine are read in. If this option is not in effect, a unit exception is reflected to the virtual machine at the end of each spooled file. CONT specified for the punch or printer causes all CLOSE requests to be ignored until reset by NOCONT. If CONT is specified, NOCONT cannot be specified.

CONT specifies that reading is to continue, without intervening end-of-file indications until all files in the system that belong to the user are read. If CONT is not in effect or is reset by specification of NOCONT, an end-of-file indication is reflected to the virtual machine at the end of each SPOOL file in the system. The nature of the end-of-file indication to be reflected is set by the EOF and NOEOF options. If the EOF option is in effect, end-of-file is signaled by a unit exception: this corresponds to pressing the end-of-file button on a real card reader. If NOEOF is in effect for a virtual reader, end-of-file is signaled by the reflection of a unit check/intervention required status.

NOCONT resets the continuous spooling option. If NOCONT is specified, CONT cannot be specified.

HOLD places all files created by the specified device in a user hold status. For READER files, this option specifies that input files for the specified reader are not deleted from the system after they are read. The status of all files must be changed by the CHANGE command. The status of output devices is changed by the SPOOL command. If HOLD is specified, NOHOLD cannot be specified.

If the HOLD option is specified for a virtual printer or punch that is transferred to a user for input (TO userid), that virtual device places a user hold status on the reader file. The user receiving the file cannot read its status until it is changed by issuing the CHANGE command with the NOHOLD operand. The spool file class of the virtual output device must match the class of the receiver's virtual reader (or the virtual reader must have a class of *) in order for the spool file to be processed. If these conditions are not satisfied, the reader appears empty to the virtual machine attempting to read a file, even though reader files do exist.

SPOOL

If a virtual reader is operating with CONT and HOLD, then virtual reader files are saved and placed in a user HOLD status. The file cannot be read until it is changed (using the CHANGE command) to a NOHOLD status.

- NOHOLD** resets the HOLD operand. Future files are not held. NOHOLD resets the HOLD operand in effect for the specified reader. This operand can be overridden for an active file being closed by the CLOSE command using the HOLD or NOHOLD operand. If NOHOLD is specified, HOLD may not be specified.
- EOF** sets a virtual end-of-file condition on the specified reader, thereby ensuring that a unit exception condition is reflected on the read that follows the reading of the last card in a file. If EOF is specified, NOEOF may not be specified.
- NOEOF** specifies that the reading continues to physical end-of-file. The virtual reader stops when no cards are left in the reader and when a unit check/intervention required status is pending. If NOEOF is specified EOF may not be specified.

[TO] userid

*

SYSTEM

transfers the output of the virtual device to the virtual card reader of the specified userid. If TO is omitted, the userid may not be TO or T. TO * may be coded if the output is to be transferred to your own virtual card reader. If TO userid is specified, neither OFF nor FOR may be specified on the same command line.

If you specify COPY with TO userid, the number of copies you specify has no effect on the receiver of the spool file; he receives only one copy. However, if OFF or FOR are specified on a subsequent command, the receiver of your spool file receives the number of copies you specify via COPY. For example, if the following command is entered:

```
SPOOL PUN TO USERA COPY 3 CLASS B
```

the COPY operand has no effect on the file going to USERA. However, if the command:

```
SPOOL PUN OFF
```

```
-- or --
```

```
SPOOL PUN FOR USERA
```

is entered, the COPY 3 specified in the first statement will affect subsequent punch files.

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

FOR userid

indicates the userid under which printed or punched output is produced. The userid becomes the owner of the output spool file and the distcode on the file is the distcode for the user that is specified in the VM/370 directory. The file is not transferred to the user's reader input. The default setting is for your own virtual machine identification. FOR *, or FOR

SYSTEM can be coded to specify your own identification and is equivalent to the OFF option.

OFF resets the transferred spool option.

COPY nn 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, nn must be between 1 and 99 (leading zeros need not be specified).

CLOSE closes the specified device regardless of the CONT setting for the device. If CLOSE is specified, PURGE may not be specified. CLOSE does not affect the setting of any other operand and is provided as a convenience to close a virtual output device. As an example, this sequence of commands:

```
SPOOL PRT CONT
(print file)
(print file)
(print file)
SPOOL PRT NOCONT
CLOSE PRT
SPOOL PRT CONT
(print file)
(print file)
.
.
.
```

can be replaced with the following sequence to achieve the desired result:

```
SPOOL PRT CONT
(print file)
(print file)
(print file)
SPOOL PRT CLOSE
(print file)
(print file)
.
.
.
```

PURGE closes and purges the spool file from the specified virtual output device regardless of the CONT setting for the device. If PURGE is specified, CLOSE cannot be specified. PURGE does not affect the setting of any other operand and is equivalent to issuing the CLOSE command for a device (or type of device) with the PURGE operand. This form of the SPOOL command is provided for your convenience.

START places all console input and output in a spool file. Until a CLOSE is issued for the console, characteristics of the console spool file may be changed by use of the SPOOL CONSOLE command. After the console is closed, the file becomes a printer spool file whose characteristics can be changed by issuing the CHANGE PRINTER command.

SPOOL

STOP terminates the spooling of console input and output. The command SPOOL CONSOLE STOP does not close the console spool file.

TERM displays the virtual console input and output at the terminal in addition to placing it in a spool file. The TERM operand has no effect until the START operand is specified.

NOTERM suppresses the display of console input and output of a system running in a virtual machine. The display of console input and output is not suppressed if:

- it is the result of CP commands entered from CP mode.
- it is 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 will be printed twice in the spooled console file).

The NOTERM operand has no effect until the START operand is specified.

Usage Notes

Unless otherwise set, the following options are default values for spool files:

Spool File	Default Options
Reader	NOHOLD NOCONT CLASS *1 EOF
Printer Punch	NOHOLD NOCONT CLASS A1 OFF COPY 01
Console	NOHOLD NOCONT CLASS T1 OFF COPY 01 TERM

¹These are default classes only if a specific class option is not part of the VM/370 directory entry.

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/370 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. You can find out various kinds of information about a file using the spoolid with the many forms of the QUERY command.

Transmitting Files to Remote Locations

To direct files to remote stations, use the CP TAG and SPOOL commands in conjunction with a command that causes the file to be closed and sent to a virtual device (for example, a virtual printer or punch). Use the TAG command to specify the device to be spooled and to associate with that device the location identifier (locid) for the destination of the file:

```
TAG DEV device locid
```

where device is the virtual device type (for example, PRINTER or PUNCH) or virtual device address (vaddr) and locid is the name of the destination to which the file is to be transmitted.

Use the SPOOL command to specify that output to the device specified in the command is to be sent to the RSCS virtual machine, which performs the actual transmission of the file:

```
SPOOL device TO userid
```

where device is the same virtual device type or virtual device address specified in the TAG command and userid is the userid of the RSCS virtual machine at your installation. You can find out the userid of your installation's RSCS virtual machine and the locid for the various remote stations from your installation's system programmer.

After you issue the TAG and SPOOL commands, use a command (such as the CMS PRINT or PUNCH command or the CP CLOSE command) to cause the spool file to be generated, closed, and spooled to the specified virtual device.

The following example shows how to use these three commands to transmit a CMS file to a remote location:

```
TAG DEV PUNCH CAMBRIDG
```

```
SPOOL PUNCH TO NET
```

```
PUNCH MYPROG ASSEMBLE
```

The TAG command defines the type of file to be transmitted, a punch file, and the remote station to which you want it transmitted, CAMBRIDG. NET is the userid of the virtual machine controlling the RSCS network: you direct your file to that virtual machine with the SPOOL command. The PUNCH command causes the file MYPROG ASSEMBLE to be punched on your virtual machine card punch, closed, and then spooled to the virtual reader of the RSCS virtual machine, which you specified in the SPOOL command. The RSCS virtual machine then processes your file (now a VM/370 spool file) and transmits it across the RSCS network.

Receiving Files from the RSCS Network

If your virtual machine is logged on VM/370, RSCS 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/370 can accumulate files from the RSCS network destined for your virtual card reader, regardless of whether you are logged on your virtual machine or not. If you are logged on your virtual machine, issue the QUERY command to see if you have any files in your virtual

SPOOL

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).

Responses

If the SPOOL command is used to close an output device, the message

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

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 the SET MSG option is OFF, this message is not issued, even though the above conditions are met.

STORE

Privilege Class: G

Use the STORE command to alter the contents of specified registers and locations of the virtual machine. The contents of the following can be altered:

- Virtual storage locations (1st level virtual storage only; see Usage Notes)
- General registers
- Floating-point registers
- Control registers (if available)
- Program status word

The STORE command can also save virtual machine data in low storage. The format of the STORE command is:

Store	hexloc	hexword1 [hexword2...]
	Lhexloc	hexword1 [hexword2...]
	Shexloc	hexdata...
	{ Greg }	hexword1 [hexword2...]
	{ Xreg }	hexword1 [hexword2...]
	{ Yreg }	hexdword1 [hexdword2...]
	Psw	[hexword1] hexword2
	STATUS	

where:

hexloc
Lhexloc hexword1 [hexword2...]
stores the specified data (hexword1 [hexword2...]) in successive fullword locations starting at the address specified by hexloc. The smallest group of hexadecimal values that can be stored using this form is one fullword. Either form (hexloc or Lhexloc) can be used.

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

hexword1 [hexword2...]
each represents up to sixteen hexadecimal digits. If the value being stored is less than a fullword (eight hexadecimal digits), it is right-adjusted in the word and the high order bytes of the word are filled with zeros. If two or more hexwords are specified, they must be separated by one or more blanks.

Shexloc hexdata...
stores the data specified (hexdata...) in the address specified by hexloc, without word alignment. The shortest string that can be stored is one byte (two hexadecimal digits). If the string contains an odd number of characters, the last character is not stored, an error message is sent, and the function is terminated.

STORE

hexdata... is a string of two or more hexadecimal digits with no embedded blanks.

Greg hexword1 [hexword2...] stores the hexadecimal data (hexword1 [hexword2...]) in successive general registers starting at the register specified by reg. The reg operand must be either a decimal number from 0-15 or a hexadecimal digit from 0-F.

hexword1 [hexword2...] each represents up to eight hexadecimal digits. If the value being stored is less than a fullword (eight hexadecimal digits), it is right-adjusted in the word and the high order bytes of the word are filled with zeros. If two or more hexwords are specified, they must be separated by one or more blanks.

Xreg hexword1 [hexword2...] stores the hexadecimal data (hexword1 [hexword2...]) in successive control registers starting at the register specified by reg. The reg operand must either be a decimal number from 0-15 or a hexadecimal digit from 0-F. If the virtual machine is in basic control mode, you can store data in register 0 only.

hexword1 [hexword2...] each represents up to eight hexadecimal digits. If the value being stored is less than a fullword (eight hexadecimal digits), it is right-adjusted in the word and the high order bytes of the word are filled with zeros. If two or more hexwords are specified, they must be separated by one or more blanks.

Yreg hexdword1 [hexdword2...] stores the hexadecimal data (hexdword1 [hexdword2...]) in successive floating-point registers starting at the register specified by reg. The reg operand must be a digit from 0-7. If reg is an odd number, it is adjusted to the preceding even number.

hexdword1 [hexdword2...] each represents up to sixteen hexadecimal digits. If the value being stored is less than a doubleword (sixteen hexadecimal digits), it is left justified in the doubleword and low order positions are filled with zeros. If two more more hexdwords are specified, they must be separated by one or more blanks.

PSW [hexword1] hexword2 stores the hexadecimal data in the first and second words of the virtual machine's program status word (PSW). If only hexword2 is specified, it is stored into the second word of the PSW.

[hexword1] hexword2 each represents up to eight hexadecimal digits. These operands must be separated by one or more blanks. If the value being stored is less than a fullword (eight hexadecimal digits), it is right-adjusted in the word and the high order bytes of the word are filled with zeros.

STATUS stores selected virtual machine data in certain low storage locations of the virtual machine, simulating the hardware

store status facility. These locations are permanently assigned locations in real storage. To use the STATUS operand, your virtual machine must be in the extended control mode. The STATUS operand should not be issued for CMS virtual machines or for DOS virtual machines generated for a CPU smaller than a System/360 Model 40. The STATUS operand stores the following data in low storage:

<u>Decimal</u> <u>Address</u>	<u>Hexadecimal</u> <u>Address</u> _____	<u>Length</u> <u>in Bytes</u>	<u>Data</u>
216	D8	8	CPU Timer
224	E0	8	Clock Comparator
256	100	8	Current PSW
352	160	32	Floating-point registers 0-6
384	180	64	General registers 0-15
448	1C0	64	Control registers 0-15

Usage Notes

1. Only first level storage (storage that is real to the virtual machine) can be stored into. Operating systems such as DOS/VS and OS/VS have virtual storage of their own. This second level virtual storage cannot be stored into directly. The user or the virtual operating system is responsible for converting any second level storage locations to first level storage.
2. The operands may be combined in any order desired, separated by one or more blanks, for up to one full line of input. If an invalid operand is encountered, an error message is issued and the store function is terminated. However, all valid operands entered, before the invalid one, are processed properly.
3. If you combine the operands for storing into storage, registers, the PSW, or the status area on a single command line, all operands must be specified; default values do not apply in this case.
4. If the STORE command is used by your virtual machine to alter the contents of a shared segment, your virtual machine will be placed in non-shared mode with your own copy of the shared segment. A fresh copy of the shared segment is then loaded for use by the other users.
5. With the STORE command, data is stored either in units of one word with fullword boundary alignment or in units of one byte without alignment.
6. The STORE STATUS command stores data in the extended logout area. The STORE STATUS command stores CPU Timer and Clock Comparator values that may then be displayed at the terminal via the DISPLAY command. The procedure is the only way to get timer information at the terminal.

Response

STORE COMPLETE

is the response at the successful completion of the command.

SYSTEM

SYSTEM

Privilege Class: G

Use the SYSTEM command to simulate the action of the RESET and RESTART buttons on the real computer console, and to clear storage. The format of the SYSTEM command is:

SYStem		{	CLEAR	}
			RESET	}
			RESTART	}

where:

- CLEAR clears virtual storage and virtual storage keys to binary zeros.
- RESET clears all pending interruptions and conditions in the virtual machine.
- RESTART simulates the hardware system RESTART function by storing the current PSW at virtual location eight and loading, as the new PSW, the doubleword from virtual location zero. Interrupt conditions and storage remain unaffected.

Usage Notes

1. The RESET function and the CLEAR function leave the virtual machine in a stopped state.
2. After issuing the SYSTEM command with RESET or CLEAR specified, either STORE a PSW and issue BEGIN or issue BEGIN with a hexadecimal storage location specified, to resume operation. The virtual machine automatically restarts at the location specified in the new PSW (which is loaded from the doubleword at location zero) after the SYSTEM RESTART command is processed.

Responses

STORAGE CLEARED - SYSTEM RESET

is the response given if the command SYSTEM CLEAR is entered.

SYSTEM RESET

is the response given if the command SYSTEM RESET is entered.

If the command SYSTEM RESTART is entered, no response is given; the virtual machine resumes execution at the address in the virtual PSW loaded from virtual storage location zero.

TAG

Privilege Class: G

Use the TAG command to associate file descriptive information with a VM/370 spool file. The format of the TAG command is:

Tag	DEV	{ Printer PUncH CONSOLE vaddr }	[tagtext]
	FILE	spoolid	[tagtext]
	QUery	{ DEV { Printer PUncH CONSOLE vaddr } FILE spoolid }	

where:

DEV { PRINTER
PRT
PUNCH
PCH
CONSOLE
vaddr }

specifies a spooling device whose output is to be associated with the tag information or whose tag information is to be queried.

FILE spoolid

specifies a previously closed spool file whose tag information is to be replaced or queried. This form of the TAG command can only be used for reader spool files that are queued on your virtual reader.

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 the RSCS component of VM/370, can be specified in this field. For details on how to use the tagtext operand when transmitting files across the RSCS network, refer to the following "Usage Notes" section.

TAG

```

      ( PRINTER
      ( PRT
      ( PUNCH
      ( PCH
      ( CONSOLE
      ( vaddr
      ( FILE  spoolid

```

displays, at your terminal, the current tag information associated with a given virtual device or spool file. The operands used with the TAG QUERY command correspond to the operands used with TAG itself.

Usage Notes

1. The RSCS control program interprets the tag information as being addressing and control parameters. When you spool a file to the RSCS virtual machine to be transmitted to a remote station, code the tagtext operand as follows:

```
locid [userid] [priority]
```

where:

locid is the location identifier (one to eight alphanumeric digits) of the location to which the file is being transmitted. Your system programmer can give you the locids of remote stations accessible to your virtual machine.

userid is the userid of the VM/370 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 the RSCS virtual machine and want the files sent to a particular VM/370 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 will be ignored by RSCS.

priority is the requested transmission priority, a decimal number between 0 and 99. The highest transmission priority is 0, next highest is 1, and so on. If you wish to specify this operand, you must also specify a userid operand.

2. If you enter the TAG command with no tagtext information, the tag area associated with the device or file is set to all blanks.
3. A spool file that you have created and spooled to another virtual machine is owned by the other user as soon as the file is closed. If you want to query or change the tag data, you must first reclaim the file via the TRANSFER command.

Responses by Command Operands

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

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

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

```
TAG QUERY DEV vaddr
```

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

```
  { PRT }
  { PUN } vaddr TAG:
  { CONS}
tagtext.....
           if tag data exists
```

```
TAG NOT SET
           if the TAG command was never issued to that device.
```

```
TAG QUERY FILE spoolid
```

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

```
tagtext.....
           if tag data exists
```

```
(TAG BLANK)
           if the tag is all blanks
```

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

TERMINAL

TERMINAL

Privilege Class: G

Use the **TERMINAL** command to control the following functions associated with your virtual console:

- Logical line-editing symbols
- Masking of password
- The APL character set
- Signaling of an attention interruption
- Attention handling mode for your virtual console
- Line length for output on your virtual console

The format of the **TERMINAL** command is:

TERMinal		CHardel	{ ON }	} ¹
		LINEDel	{ OFF }	
		LINENd	{ char }	
		EScape		
		Mask	{ ON }	
APL	{ OFF }			
ATtn				
MODE	{ CP }			
	{ VM }			
LINESize	nnn			

¹ More than one function can be specified in a single entry of the **TERMINAL** command. For example:

```
TERMINAL CHARDEL OFF MASK ON LINESIZE 90
```

where:

CHARDEL { ON } defines the logical character delete symbol. If ON is
 { OFF } is specified, the default symbol becomes the logical
 { char } character delete symbol. The default symbol is normally
 @, but depends on what is specified in your VM/370
 directory entry. If OFF is specified, no logical
 character delete symbol is allowed. If char is specified,
 that character (which must be a special character)
 becomes the logical character delete symbol. When you
 log on, CHARDEL ON is in effect; if your virtual machine
 is logged on by the system operator via the AUTOLOG
 command, CHARDEL OFF is the default.

LINEDEL { ON } defines the logical line delete symbol. If ON is specified, the default symbol becomes the logical line delete symbol. The default symbol is normally #, but depends on what is specified in your VM/370 directory entry. If OFF is specified, no logical line delete symbol is allowed. If char is specified, that character (which must be a special character) becomes the logical line delete symbol. When you log on, LINEDEL ON is in effect; if your virtual machine is logged on by the system operator via the AUTOLOG command, LINEDEL OFF is the default.

LINEND { ON } defines the logical line end symbol. If ON is specified, the default symbol becomes the logical line end symbol. The default symbol is normally #, but depends on what is specified in your VM/370 directory entry. If OFF is specified, no logical line end symbol is allowed. If char is specified, that character (which must be a special character) becomes the logical line end character. When you log on, LINEND ON is in effect; if your virtual machine is logged on by the system operator via the AUTOLOG command, LINEND OFF is the default.

ESCAPE { ON } defines the logical escape character. If ON is specified, the default symbol becomes the logical escape character. The default symbol is normally ", but depends on what is specified in your VM/370 directory entry. If OFF is specified, no logical escape character is allowed. If char is specified, that character (which must be a special character) becomes the logical escape character. When you log on, ESCAPE ON is in effect; if your virtual machine is logged on by the system operator via the AUTOLOG command, ESCAPE OFF is the default.

MASK { ON } controls the typing of a mask line at a typewriter terminal that is not equipped with the print inhibit feature, when a password is to be entered. If MASK ON is specified, VM/370 types the mask line. If MASK OFF is specified, the mask line is not typed and it is up to each user to preserve the security of his password. The MASK operand does not apply to the IBM 3215 Console Printer-Keyboard or to similar system console or display terminals that do not have a print inhibit feature. Unless otherwise specified, MASK OFF is in effect.

APL { ON } controls the use of APL character translation tables. If APL ON is specified, CP uses the translation tables for display stations equipped with the APL hardware feature and for typewriter terminals equipped with the standard APL typing element. APL ON also applies to the 3767 terminal equipped with the APL alternate character selection. If APL OFF is specified, CP uses the normal translation tables (that is, BCD or correspondence code). Unless otherwise specified, APL OFF is in effect.

Note: This operand is not valid for a terminal connected to a 3704/3705 device in NCP mode or in the NCP portion of PEP mode.

TERMINAL

If APL ON is specified, the LINESIZE value is overridden (see the explanation for the LINESIZE operand, below).

ATTN { ON }
 { OFF }

controls signaling of an attention interruption. If ATTN ON is specified, the exclamation point is displayed when an attention interruption occurs. The OFF option suppresses the displaying of the exclamation point (!) and carrier return for those systems that perform special line editing using the Attention key. Unless otherwise specified, ATTN ON is in effect.

Note: The ATTN operand is not valid for display type terminals.

MODE { CP }
 { VM }

controls the terminal attention environment. CP specifies that one or more attentions force the virtual machine into the CP environment. VM specifies that one attention is reflected to your virtual machine and that more than one attention forces your virtual machine into the CP environment. VM is the default for all VM/370 users except the primary system operator. For more information on this, see "Interrupting the Execution of a Command" in Section 2.

LINESIZE nnn specifies the maximum allowable line length for terminal output. nnn can be a number from 1 through 255.

Note: 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.

Usage Notes

1. The terminal settings you specify with the TERMINAL command are in effect only for the duration of that terminal session. Whenever you initially log on, the system defaults are in effect. However, the settings you specify for line-editing and MODE are still in effect when you log on after disconnecting. All the other operands (MASK, APL, ATTN, and LINESIZE) are reset if you log on after disconnecting.
2. Although you can define line-editing symbols and status with the TERMINAL command, the LINEDIT operand of the SET command determines whether the VM/370 line-editing functions are on or off.
3. If an error occurs during processing of the command, all functions preceding the one with the error are in effect.
4. You can not use any of the letters, A through Z, as a symbol (char) in association with any of the CHARDEL, LINEDEL, LINEND, and ESCAPE operands.

Responses

None.

TRACE

Privilege Class: G

Use the TRACE command to trace specified virtual machine activity and to record the results at the terminal, on a virtual spooled printer, or on both terminal and printer. The format of the TRACE command is:

```

Trace  {
        {
          SVC
          I/O
          PROgram
          EXTERNAL
          PRIV
          SIO
          CCW
          BRANCH
          INSTRUCT
          ALL
          CSW
        }
        {
          PRINter
          [TERMinal] [NORun]
          [BOTH]     [RUN]
          Off
        }
      }
END

```

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

```
TRACE SVC PROGRAM SIO PRINTER
```

where:

SVC traces virtual machine SVC interruptions.

I/O traces virtual machine I/O interruptions.

PROGRAM traces virtual machine program interruptions.

EXTERNAL traces virtual machine external interruptions.

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

SIO traces TIO, CLRIO, HIO, HDV, and TCH instructions to all virtual devices. Also traces SIO and SIOF instructions for nonconsole and nonspool devices only.

CCW traces virtual and real CCWs for nonspool nonconsole device I/O operations. When CCW tracing is requested, SIO and TIO instructions to all devices are also traced.

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

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

ALL traces all instructions, interruptions, succesful branches, privilege instructions, and virtual machine I/O operations.

CSW provides contents of virtual and real channel status words at I/O interruption.

TRACE

- END** terminates all tracing activity and prints a termination message.
- PRINTER** directs tracing output to a virtual spooled printer.
- TERMINAL** directs tracing output to the terminal (virtual machine console).
- BOTH** directs tracing output to both a virtual spooled printer and the terminal.
- OFF** halts tracing of the specified activities on both the printer and terminal.
- NORUN** stops program execution after the trace output to the terminal and enters the CP command environment.
- Note: If a Diagnose code X'008' is being traced, NORUN has no effect and program execution does not stop.
- RUN** continues the program execution after the trace output to the terminal has completed and does not enter the CP command environment.

Usage Notes

1. If your virtual machine has the virtual=real option and NOTRANS set on, CP forces CCW translation while tracing either SIO or CCW. When tracing is terminated with the TRACE END command, CCW translation is bypassed again.
2. If the virtual machine assist feature is enabled on your virtual machine, CP turns it off while tracing SVC, PRIV, BRANCH, INSTRUCT, or ALL activities. After the tracing is terminated with the TRACE END command line, CP turns the assist feature on again.
3. If trace output is being recorded at the terminal, the virtual machine stops execution and CP command mode is entered after each output message. This simulates the instruction step function.

| However, all processing associated with the event being traced
| will be completed and, therefore, execution may have stopped after
| an instruction has executed and the PSW has been updated.

| For example, a privileged instruction traced with the PRIV
| operand will stop after the privileged instruction executes,
| whereas the same instruction traced with the ALL operand will stop
| before the instruction executes.

| To determine whether the traced instruction has executed,
| display the virtual machine PSW.

 To resume operation at the virtual machine, the BEGIN command must be entered. If the RUN operand is specified, the virtual machine is not stopped after each output message.

4. If trace output is being recorded on a virtual spooled printer, a CLOSE command must be issued to that printer in order for the trace output to be printed on the real printer.
5. Successful branches to the next sequential instruction and branch-to-self instructions are not detected by TRACE.

6. Instructions that modify or examine the first two bytes of the next sequential instruction cause erroneous processing for BRANCH and INSTRUCT tracing.
7. 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.
8. If the BRANCH, INSTRUCT, or ALL activities are being traced by a virtual machine using a shared system, the virtual machine is placed in nonshared mode with its own copy of the shared segment. A fresh copy of the shared segment is then loaded for use by the other users.
9. I/O operations for virtual channel-to-channel adapters, with both ends connected to the same virtual machine, cannot be traced.
10. Use the TRACE command to trace specified virtual machine activity and to record the results at the terminal, at a virtual printer, or at both. This command is useful in debugging programs because it allows you to trace only the information that pertains to a particular problem.
11. 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 would cause a system abend.

Responses

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

<u>Symbol</u>	<u>Meaning</u>
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

TRACE

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
```

The IDAL or SEEK line is included only if applicable. The virtual IDAL is not printed if the real CCW operation code does not match the real CCW.

INSTRUCTION TRACING:

Privileged Instruction:

```
::: vvvvvv SSM xxxxxxxx ss (normal SSM)
::: vvvvvv SSM xxxxxxxx ss ttttt (switch to/from translate mode)
::: vvvvvv STOSM xxxxxxxx ns (normal STOSM)
::: vvvvvv STOSM xxxxxxxx ns ttttt (switch to translate mode)
::: vvvvvv STNSM xxxxxxxx ns (normal STNSM)
::: vvvvvv STNSM xxxxxxxx ns ttttt (switch from translate mode)
::: vvvvvv LPSW xxxxxxxx ttttttt ttttttt (WAIT bit on)
::: vvvvvv LPSW xxxxxxxx ==> ttttttt ttttttt (WAIT bit not on)
::: vvvvvv mnem xxxxxxxx (all others)
```

Executed Instructions:

vvvvvv EX xxxxxxxx zz vvvvvv mnem xxxx xxxxxxxx

For an executed instruction, where zz (see preceding explanation of symbols) is nonzero, the mnemonic for the executed instruction is given as if the zz byte had been put into the instruction with an OR operation.

All Other Instructions:

vvvvvv mnem xxxxxxxx xxxx

SUCCESSFUL BRANCH:

vvvvvv mnem xxxxxxxx ==> tttttt

INTERRUPTION (SVC, PROGRAM, or EXTERNAL)

*** vvvvvv int code ==> tttttt

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

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

BRANCH TRACE: (ALL option selected)

Entry for 'branch from' instruction

vvvvvv mnem xxxxxxxx tttttt

Entry for 'branch to' instruction

==> vvvvvv mnem xxxxxxxxxxxx

TRANSFER

TRANSFER

Privilege Class: G

Use the TRANSFER command to direct your input file to a specified reader or to reclaim virtual reader files that you spooled to another user. The format of the TRANSFER command is:

```
TRANSFER | { spoolid } { [To] userid }  
          | { Class c } { From { userid } }  
          | { ALL }     { ALL }
```

where:

spoolid is the input file to be directed to or retrieved from the named userid.

CLASS c is the class of input files that are to be transferred. The c is a one-character alphanumeric field with values from A to Z and from 0 to 9.

ALL specifies that all input spool files are to be transferred.

[TO] userid is the user to whom the files are to be directed. If the optional keyword TO is omitted, the userid may not be TO or T. The file is deleted from your reader if you use this option.

FROM {userid}
{ALL} is the user from whom input spool files are to be reclaimed. ALL may be specified to reclaim input spool files that were originated by your virtual machine from all users.

Usage Notes

1. The TRANSFER command does not transfer any active spool files.

Responses

```
RDR FILE spoolid TRANSFERRED { TO } userid  
                             { FROM }
```

where:

spoolid is the spool file identification number of the file that is spooled. The number does not change.

TO userid is the response to the user who transferred the file verifying that userid is the recipient of the file.

TRANSFER

FROM userid is the response to the user who receives the transferred file indicating that userid is the sender.

FROM userid is also the response to a user who reclaims a transferred file and userid is the user from whom the file was reclaimed.

{ nnnn } FILES TRANSFERRED
{ NO }

is also a response you receive when you issue the TRANSFER command. It is not displayed if you issued the CP SET IMSG OFF command line.

Appendix A: VM/370 Command Summary

This appendix contains brief descriptions of all the commands acceptable to the CP, CMS and RSCS components of the VM/370 system.

Figure 7 contains an alphabetical list of the CP commands, the privilege classes which may execute the command, and a brief statement about the use of each command.

Command	Privilege Class	Usage
*	any	Annotate the console sheet.
#CP	any	Execute a CP command while remaining in the virtual machine environment.
ACNT	A	Create accounting records for logged on users and reset accounting data.
ADSTOP	G	Halt execution at a specific virtual machine instruction address.
ATTACH	B B B	Attach a real device to a virtual machine. Attach a DASD device for CP control. Dedicate all devices on a particular channel to a virtual machine.
ATTN	G	Make an attention interruption pending for the virtual machine console.
AUTOLOG	A,B	Automatically log on a virtual machine and have it operate in disconnect mode.
BACKSPAC	D	Restart or reposition the output of a unit record spooling device.
BEGIN	G	Continue or resume execution of the virtual machine at either a specific storage location or at the address in the current PSW.
CHANGE	D,G	Alter one or more attributes of a closed spool file.
CLOSE	G	Terminate spooling operations on a virtual card reader, punch, printer, or console.
COUPLE	G	Connect channel-to-channel adapters.
CP	any	Execute a CP command while remaining in the CMS virtual machine environment.
DCP	C,E	Display real storage at terminal.
DEFINE	G	Reconfigure your virtual machine.

Figure 7. CP Command Summary (Part 1 of 4)

Command	Privilege Class	Usage
DETACH	B B B G G	Disconnect a real device from a virtual machine. Detach a DASD device from CP. Detach a channel from a specific user. Detach a virtual device from a virtual machine. Detach a channel from your virtual machine.
DIAL	any	Connect a terminal or display device to the virtual machine's virtual communication line.
DISABLE	A,B	Disable 2701/2702/2703, 3704/3705 in EP mode, and 3270 local communication lines.
DISCONN	any	Disconnect your terminal from your virtual machine.
DISPLAY	G	Display virtual storage on your terminal.
DMCP	C,E	Dump the specified real storage location on your virtual printer.
DRAIN	D	Halt operations of specified spool devices upon completion of current operation.
DUMP	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	Test terminal hardware by redisplaying data entered at the terminal.
ENABLE	A,B	Enable communication lines.
EXTERNAL	G	Simulate an external interruption for a virtual machine and return control to that machine.
FLUSH	D	Cancel the current file being printed or punched on a specific real unit record device.
FORCE	A	Cause logoff of a specific user.
FREE	D	Remove spool HOLD status.
HALT	A	Terminate the active channel program on specified real device.
HOLD	D	Defer real spooled output of a particular user.
INDICATE	E,G	Indicate resource utilization and contention.
IPL	G	Simulate IPL for a virtual machine.
LINK	G	Provide access to a specific DASD device by a virtual machine.
LOADBUF	D	Load real UCS/UCSB or FCB printer buffers.

Figure 7. CP Command Summary (Part 2 of 4)

Command	Privilege Class	Usage
LOADVFCB	G	Load virtual forms control buffer for a virtual 3211 printer.
LOCATE	C,E	Find CP control blocks.
LOCK	A	Bring virtual pages into real storage and lock them; thus excluding them from future paging.
LOGOFF	any	Disable access to CP.
LOGON	any	Provide access to CP.
MESSAGE	A,B,any	Transmit messages to other users.
MONITOR	A,E	Trace events of the real machine and record system performance data.
NETWORK	A,B,F	Load, dump, trace and control the operation of the 3704/3705 control program. Control the operation of 3270 remote devices.
NOTREADY	G	Simulate "not ready" for a device to a virtual machine.
ORDER	D,G	Rearrange closed spool files in a specific order.
PURGE	D,G	Remove closed spool file from system.
QUERY	A,B,C,D, E,F,G	Request information about machine configuration and system status.
READY	G	Simulate device end interruption for a virtual device.
REPEAT	D	Repeat (a specified number of times) printing or punching of a specific real spool output file.
REQUEST	G	Make an attention interruption pending for the virtual machine console.
RESET	G	Clear and reset all pending interruptions for a specified virtual device and reset all error conditions.
REWIND	G	Rewind (to load point) a tape and ready a tape unit.
SAVESYS	E	Save virtual machine storage contents, registers, and PSW.
SET	A,B,F,G	Operator--establish system parameters. User--control various functions within the virtual machine.

Figure 7. CP Command Summary (Part 3 of 4)

Command	Privilege Class	Usage
SHUTDOWN	A	Terminate all VM/370 functions and checkpoint CP system for warm start.
SLEEP	any	Place virtual machine in dormant state.
SPACE	D	Force single spacing on printer.
SPOOL	G	Alter spooling control options; direct a file to another virtual machine or to a remote location via the RSCS virtual machine.
START	D	Start spooling device after draining or changing output classes.
STCP	C	Change the contents of real storage.
STORE	G	Alter specified virtual storage locations and registers.
SYSTEM	G	Simulate RESET, CLEAR STORAGE and RESTART buttons on a real system console.
TAG	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	Define or redefine the input and attention handling characteristics of your virtual console.
TRACE	G	Trace specified virtual machine activity at your terminal, spooled printer, or both.
TRANSFER	D,G	Transfer input files to or reclaim input files from a specified user's virtual card reader.
UNLOCK	A	Unlock previously locked page frames.
VARY	B	Mark a device unavailable or available.
WARNING	A,B	Transmit a high priority message to a specified user or to all users.

Figure 7. CP Command Summary (Part 4 of 4)

Figure 8 contains an alphabetical list of the CMS commands with a brief statement about the use of each command. Unless otherwise noted under Ref. Code, the CMS commands are described in the VM/370: CMS Command and Macro Reference.

Use the following chart to determine which publication contains reference information about the CMS commands that are not described in the above publication.

Ref
Code

SYSGEN Described in the VM/370: Planning and System Generation Guide.

OS PP Invokes an OS Program Product, available from IBM for a license fee.

EREP Described in the VM/370: Environmental Recording, Editing, and Printing (EREP) Program.

Op Gd Described in the VM/370: Operator's Guide.

SCRIPT Invokes a text processor that is an IBM Installed User Program, available from IBM for a license fee.

DOS PP Invokes a DOS Program Product available from IBM for a license fee.

SPG Described in the VM/370: System Programmer's Guide.

IPCS Described in the VM/370: Interactive Problem Control System (IPCS) User's Guide.

Command	Code	Usage
ACCESS		Identify direct access space to a CMS virtual machine, create extensions and relate the disk space to a logical directory.
AMSERV		Invoke Access Method Services utility functions to create, alter, list, copy, delete, import, or export VSAM catalogs and data sets.
ASM3705	SYSGEN	Assemble 3704/3705 source code.
ASSEMBLE		Assemble Assembler Language source code.
ASSGN		Assign or unassign a CMS/DOS system or programmer logical unit for a virtual I/O device.
CMSBATCH		Invoke the CMS Batch Facility.
COBOL	OS PP	Compile OS ANS Version 4 or OS/VS COBOL source code.
COMPARE		Compare records in CMS disk files.
CONVERT	OS PP	Convert free form FORTRAN statements to fixed form.
COPYFILE		Copy CMS disk files according to specifications.
CP		Enter CP commands from the CMS environment.
CPEREP	EREP	Dump error information which was recorded by VM/370 error recording routines.
DDR	Op Gd, SYSGEN	Perform backup, restore, and copy operations for disks.
DEBUG		Enter DEBUG subcommand environment, debug mode.
DIRECT	Op Gd, SYSGEN	Set up VM/370 directory entries.
DISK		Perform disk-to-card and card-to-disk operations for CMS files.
DLBL		Define a DOS filename or VSAM ddname and relate that name to a disk file.
DOSGEN	SYSGEN	Load and save the CMSDOS shared segment.
DOSLIB		Delete, compact, or list information about the phases of a CMS/DOS phase library.
DOSLKED		Link-edit CMS text decks or object modules from a DOS/VS relocatable library and place them in executable form in a CMS/DOS phase library.
DOSPLI	DOS PP	Compile DOS PL/I source code under CMS/DOS.
DSERV		Display information contained in the DOS/VS core image, relocatable, source, procedure, and transient directories.

Figure 8. CMS Command Summary (Part 1 of 4)

Command	Code	Usage
DUMPSCAN	IPCS	Provide interactive analysis of CP abend dumps.
EDIT		Invoke the CMS Editor to create or modify a disk file.
ERASE		Delete CMS disk files.
ESERV		Display, punch or print an edited (compressed) macro from a DOS/VS source statement library (E sublibrary).
EXEC		Execute special procedures made up of frequently used sequences of commands.
FCOBOL	DOS PP	Compile DOS/VS COBOL source code under CMS/DOS.
FETCH		Fetch a CMS/DOS executable phase.
FILEDEF		Define an OS ddname and relate that ddname to any device supported by CMS.
FORMAT		Prepare disks in CMS 800-byte block format.
FORTGI	OS PP	Compile FORTRAN source code using the G1 compiler.
FORTHX	OS PP	Compile FORTRAN source code using the H-extended compiler.
GEN3705	SYSGEN	Generate an EXEC file that assembles and link-edits the 3704/3705 control program.
GENDIRT		Fill in auxiliary module directories.
GENMOD		Generate non-relocatable CMS files (MODULE files).
GLOBAL		Identify specific CMS libraries to be searched for macros, copy files, missing subroutines, or DOS executable phases.
GOFCRT	OS PP	Compile FORTRAN source code and execute the program using the FORTRAN Code and Go compiler.
INCLUDE		Bring additional TEXT files into storage and establish linkages.
LISTDS		List information about data sets and space allocation on OS, DOS, and VSAM disks.
LISTFILE		List information about CMS disk files.
LISTIO		Display information concerning CMS/DOS system and programmer logical units.
LKED	SYSGEN	Link-edit the 3704/3705 control program.
LOAD		Bring TEXT files into storage for execution.
LOADMOD		Bring a single MODULE file into storage.
MACLIB		Create or modify CMS macro libraries.

Figure 8. CMS Command Summary (Part 2 of 4)

Command	Code	Usage
MODMAP		Display the load map of a MODULE file.
MOVEFILE		Move data from one device to another device of the same or a different type.
NCPDUMP	Op Gd, SYSGEN, SPG	Process CP spool reader files created by 3704/3705 dumping operations.
OPTION		Change the DOS COBOL compiler (FCOBOL) options that are in effect for the current terminal session.
PLIC	OS PP	Compile and execute PL/I source code using the PL/I Checkout Compiler.
PLICR	OS PP	Execute the PL/I object code generated by the OS PL/I Checkout Compiler.
PLIOPT	OS PP	Compile PL/I source code using the OS PL/I Optimizing Compiler.
PRB	IPCS	Update IPCS problem status.
PRINT		Spool a specified CMS file to the virtual printer.
PROB	IPCS	Enter a problem report in IPCS.
PSERV		Copy a procedure from the DOS/VS procedure library onto a CMS disk, display the procedure at the terminal, or spool the procedure to the virtual punch or printer.
PUNCH		Spool a copy of a CMS file to the virtual punch.
QUERY		Request information about a CMS virtual machine.
READCARD		Read data from spooled card input device.
RELEASE		Make a disk and its directory inaccessible to a CMS virtual machine.
RENAME		Change the name of a CMS file or files.
RSERV		Copy a DOS/VS relocatable module onto a CMS disk, display it at the terminal, or spool a copy to the virtual punch or printer.
RUN		Initiate series of functions to be performed on a source, MODULE, TEXT, or EXEC file.
SAVENCPC	SYSGEN, SPG	Read 3704/3705 control program load into virtual storage and save an image on a CP-owned disk.
SCRIPT	SCRIPT	Format and print documents according to embedded SCRIPT control words in the document file.
SET		Establish, set, or reset CMS virtual machine characteristics.

Figure 8. CMS Command Summary (Part 3 of 4)

Command	Code	Usage
SETKEY	SPG	Assign storage protect keys to storage assigned to named systems.
SORT		Arrange a specified file in ascending order according to sort fields in the data records.
SSERV		Copy a DOS/VS source statement book onto a CMS disk, display it at the terminal, or spool a copy to the virtual punch or printer.
START		Begin execution of programs previously loaded (OS and CMS) or fetched (CMS/DOS).
STAT	IPCS	Display the status of reported system problems.
STATE		Verify the existence of a CMS disk file.
STATEW		Verify the existence of a file on a read/write CMS disk.
SVCTRACE		Record information about supervisor calls.
SYNONYM		Invoke a table containing synonyms you have created for CMS and user-written commands.
TAPE		Perform tape-to-disk and disk-to-tape operations for CMS files, and position tapes.
TAPEMAC		Create CMS MACLIB libraries directly from an IEHMOVE-created partitioned data set on tape.
TAPPDS		Load OS partitioned data set (PDS) files or card image files from tape to disk.
TESTCOB	OS PP	Invoke the OS COBOL Interactive Debug Program.
TESTFORT	OS PP	Invoke the FORTRAN Interactive Debug Program.
TXTLIB		Generate and modify text libraries.
TYPE		Display all or part of a CMS file at the terminal.
UPDATE		Make changes in a program source file as defined by control cards in a control file.
VMFDUMP	Op Gd IPCS	Format and print system abend dumps; under IPCS, create a problem report.
VMFLOAD	SYSGEN	Generate a new CP, CMS or RSCS module.
VSAMGEN	SYSGEN	Load and save the VSAM shared segment.
VS BASIC	OS PP	Compile and execute VS BASIC programs under CMS.
VS BUTIL	OS PP	Convert BASIC 1.2 data files to the format required by VS BASIC.
ZAP	Op Gd, SYSGEN, SPG	Modify or dump LOADLIB, TXTLIB, or MODULE files.

Figure 8. CMS Command Summary (Part 4 of 4)

Figure 9 contains an alphabetical list of the RSCS commands with a brief statement about the use of each command. RSCS commands are described in the VM/370: Remote Spooling Communications Subsystem (RSCS) User's Guide.

Command Name	Function
BACKSPAC	Restart or reposition, in a backward direction, the file currently being transmitted.
CHANGE	Alters one or more attributes of a file owned by RSCS.
CMD	Control certain functions performed by a remote system, or control the logging of I/O activity on a specified link.
DEFINE	Temporarily add a new link definition to the RSCS link table or temporarily redefine an existing link.
DELETE	Temporarily delete a link definition from the RSCS link table.
DISCONN	Place RSCS in disconnect mode and optionally direct output to another virtual machine.
DRAIN	Deactivate an active communication link.
FLUSH	Discontinue processing the current file on the specified link.
FREE	Resume transmission on a communication link previously in HOLD status.
FWDSPACE	Reposition, in a forward direction, the file currently being transmitted.
HOLD	Suspend file transmission on an active link without deactivating the line.
MSG	Send a message to a local or remote station.
ORDER	Reorder files enqueued on a specific link.
PURGE	Remove all or specified files from a link.
QUERY	Request system information for a link, a file, or for the system in general.
START	Activate a specified communication link.
TRACE	Monitor line activity on a specified link.

Figure 9. RSCS Command Summary

* command
 described 34
 summary 143

#CP command
 described 35
 summary 143

A
 abbreviations, controlling 19
 ACCESS command, summary 148
 accounting, information, displaying 110
 ACNT
 command, summary 143
 operand, of SET command (CP) 110
 ADSTOP command
 described 37
 OFF operand 37
 summary 143
 ALL operand
 of CHANGE command (CP) 41
 of PURGE command (CP) 91
 of QUERY command (CP) 95
 of TRACE command (CP) 135
 of TRANSFER command 140
 altering
 attributes of
 virtual printer file 41
 virtual punch file 41
 virtual reader file 41
 logical editing symbols 132
 virtual machine environment 49
 AMSERV command, summary 148
 annotating your console sheet 34
 Any user, described 22
 APL
 control of translation tables 132
 operand, of TERMINAL command 133
 ASM3705 command, summary 148
 ASSEMBLE command, summary 148
 ASSGN command, summary 148
 ASSIST operand, of SET command (CP) 113
 ATTACH command, summary 143
 attention
 handling, terminal control of 132
 interrupt
 effect on virtual console in CP mode 16
 effect on virtual console in VM mode 15
 entering from your virtual console 39
 for a virtual machine 107
 Attention (ATTN) key
 how to use 13
 used to interrupt execution of a command 13
 used to switch command environments 13

ATTN
 command
 described 39
 summary 143
 operand, of TERMINAL command 134
 attributes
 spool file, changing 41
 virtual device, modifying 118

B
 BACKSPAC
 command (CP), summary 143
 command (RSCS), summary 152
 BEGIN command
 described 40
 summary 143
 used with the DISCONN command (CP) 59
 blanks, as delimiters 17
 BOTH operand, of TRACE command (CP) 136
 BRANCH operand, of TRACE command (CP) 135
 branches, tracing 135

C
 CAW (Channel Address Word), displaying 63
 CAW operand, of DISPLAY command 63
 CCW operand, of TRACE command (CP) 135
 CCW translation, controlling 112
 CHANGE
 command (CP)
 ALL operand 41
 CLASS operand 41
 COPY operand 41
 described 41
 DIST operand 42
 HOLD operand 41,43
 NAME operand 42
 NOHOLD operand 41
 PRINTER operand 41
 PUNCH operand 41
 READER operand 41
 summary 143
 command (RSCS), summary 152
 changing
 extended control registers in your virtual machine 125
 filename of a spool file 42
 floating-point registers in your virtual machine 125
 general registers in your virtual machine 125
 Program Status Word (PSW) in your virtual machine 125
 spool class of a file 41
 spool file attributes 41
 spoolid of a file 41
 virtual storage locations 125
 CHANNELS operand
 of DEFINE command (CP) 50

- of QUERY command (CP) 94
- channel-to-channel adapters, virtual, connecting 46
- character
 - set 18
 - translation tables, APL, controlling 132
- CHARDEL operand, of TERMINAL command 132
- class
 - privilege, for CP commands 21
 - spool, changing for a file 41
 - virtual device spool, modifying 118
- CLASS operand
 - of CHANGE command (CP) 41
 - of PURGE command (CP) 91
 - of QUERY command (CP) 96
 - of SPOOL command 119
 - of TRANSFER command 140
- CLEAR operand
 - of IPL command 75
 - of SYSTEM command 128
- clearing
 - interrupts from a virtual machine 108
 - storage to zeros 75,128
- CLOSE
 - command
 - CONSOLE operand 43
 - described 43
 - DIST operand 44
 - NAME operand 44
 - NOHOLD operand 43
 - PRINTER operand 43
 - PUNCH operand 43
 - PURGE operand 44
 - READER operand 43
 - summary 143
 - operand, of SPOOL command 121
- closing
 - files 43
 - virtual devices 118
- CMD command, summary 152
- CMS commands, summary of 148
- CMSBATCH command, summary 148
- COBOL command, summary 148
- command
 - name 17
 - operands 17
 - options 17
- command environment
 - CP 13
 - defined 13
 - switching 14
 - VM 13
- commands
 - abbreviation of 19,29
 - CMS, summarized 148
 - CP, summarized 143
 - minimum truncation for 18,29
 - notational conventions for 29
 - RSCS, summarized 152
 - summary of use 143
 - truncation of 18
 - used to control a terminal session, summary of 143
- comments, how to write 17
- COMPARE command, summary 148
- connecting
 - remote terminals to virtual machines 57

- virtual channel-to-channel adapters 46
- console
 - input/output, terminating 118
 - sheet, annotation of 34
 - spool file, modifying 118
 - spooling, controlling 118
- CONSOLE operand
 - of CLOSE command 43
 - of DEFINE command (CP) 49
 - of QUERY command (CP) 94
 - of SPOOL command 119
- CONT operand, of SPOOL command 119
- continuous reading of an entire file 118
- control, functions, setting for your virtual machine 110
- Control Program (see CP (Control Program))
- controlling, a terminal session, summary of commands used 143
- CONVERT command, summary 148
- COPY operand
 - of CHANGE command (CP) 41
 - of SPOOL command 121
- COPYFILE command, summary 148
- COUPLE command
 - described 46
 - summary 143
- CP (Control Program), basic description 11
- CP command
 - described 48
 - summary 148,143
- CP command environment 13
- CP commands
 - entering from the VM environment 48
 - privilege classes, described 21
 - privilege classes for 21
- CPEREP command, summary 148
- CSW (Channel Status Word), displaying 63
- CSW operand
 - of DISPLAY command 63
 - of TRACE command (CP) 135
- CTCA operand, of DEFINE command (CP) 50

D

- DASD operand, of QUERY command (CP) 94
- DCP command, summary 143
- DDR command, summary 148
- DEBUG command, summary 148
- dedicated channel, detaching 56
- DEFINE
 - command (CP)
 - CHANNELS operand 50
 - CONSOLE operand 49
 - CTCA operand 50
 - described 49
 - GRAF operand 51
 - LINE operand 51
 - PRINTER operand 49
 - PUNCH operand 49
 - READER operand 49
 - STORAGE operand 51
 - summary 143
 - TIMER operand 50
 - T2305 operand 51
 - T2314 operand 51
 - T2319 operand 51

T3330 operand 51
 T3340 operand 51
 T3350 operand 51
 1403 operand 50
 3211 operand 50
 command (RSCS), summary 152
 defining, virtual devices 49
 DELETE command, summary 152
 delimiters, in command lines 17
 DETACH CHANNEL command, described 56
 DETACH command
 described 54
 summary 144
 detaching
 dedicated channels 56
 virtual devices 54
 device-end interrupt pending for a virtual
 device, specifying 106
 devices
 virtual
 linking to 77
 purging from your system 91
 simulating not ready status for 88
 DIAL command
 described 57
 summary 144
 DIRECT command, summary 148
 DISABLE command, summary 144
 DISCONN
 command (CP)
 described 59
 HOLD operand 59
 summary 144
 command (RSCS), summary 152
 disconnecting
 virtual console 59
 virtual machine 82
 DISK command, summary 148
 DISPLAY command
 CAW operand 63
 CSW operand 63
 described 61
 PSW operand 63
 summary 144
 displaying
 accounting information 110
 Channel Address Word (CAW) 63
 Channel Status Word (CSW) 63
 extended control registers 62
 floating-point registers 62
 general registers 62
 Program Status Word (PSW) 63
 storage keys 61
 virtual storage locations 61
 DIST operand
 of CHANGE command (CP) 42
 of CLOSE command 44
 distribution code, for an output file 42
 DLBL command, summary 148
 DMCP command, summary 144
 DOSGEN command, summary 148
 DOSLIB command, summary 148
 DOSLKED command, summary 148
 DOSPLI command, summary 148
 DRAIN
 command (CP), summary 144
 command (RSCS), summary 152
 DSERV command, summary 148
 DUMP command
 described 67
 summary 144
 DUMPSCAN command, summary 148

 E
 ECHO command
 described 70
 summary 144
 ECMODE operand, of SET command (CP) 111
 EDIT command, summary 149
 editing symbols
 controlling 132
 logical, controlling use of 111
 EMSG operand, of SET command (CP) 112
 ENABLE command, summary 144
 END operand, of TRACE command (CP) 136
 entering
 a CP command from the VM environment
 35,48
 CP and CMS commands 17
 EOF operand, of SPOOL command 120
 ERASE command, summary 149
 error message handling, controlling 112
 ESCAPE operand, of TERMINAL command 133
 ESERV command, summary 149
 EXEC command, summary 149
 execution
 halting at an instruction address 37
 of a virtual machine, resuming 40
 extended control mode, setting 111
 extended control registers
 changing your virtual machine 125
 displaying 62
 printing 67
 EXTERNAL
 command
 described 71
 summary 144
 operand, of TRACE command (CP) 135
 external interrupt, simulated 71

 F
 FCB operand, of LOADVFCB command 81
 FCBOLE command, summary 149
 FETCH command, summary 149
 FILEDEF command, summary 149
 files
 changing the spool class of 41
 changing the spoolid for 41
 closing 43
 distribution code for 42
 holding before output processing 41
 holding from further processing 43,118
 purging 43,118
 reclaiming 140
 reordering closed spool files 89
 spool, changing filename for 42
 terminating processing of 43
 transferring to another user 118
 transmitting to a virtual reader 140
 FILES operand, of QUERY command (CP) 94
 floating-point registers
 changing your virtual machine 125
 displaying 62
 printing 67

FLUSH
 command (CP), summary 144
 command (RSCS), summary 152
 FOR operand, of SPOOL command 120
 FORCE command, summary 144
 FORMAT command, summary 149
 forms controls for a 3211 printer,
 specifying 81
 PORTGI command, summary 149
 FORTHX command, summary 149
 FREE command, summary 152,144
 FROM operand, of TRANSFER command 140
 FWDSPACE command, summary 152

G
 GENDIRT command, summary 149
 general registers
 changing in your virtual machine 125
 displaying 62
 printing 67
 general user, described 22
 GENMOD command, summary 149
 GEN3705 command, summary 149
 GLOBAL command, summary 149
 GOFORT command, summary 149
 GRAF operand
 of DEFINE command (CP) 51
 of QUERY command (CP) 94

H
 HALT command, summary 144
 halting, execution at an instruction
 address 37
 handshaking, control of VM/VS 112
 hiding your password 84
 HOLD
 command (CP), summary 144
 command (RSCS), summary 152
 operand
 of CHANGE command (CP) 41,43
 of DISCONN command (CP) 59
 of LOGOFF command 82
 of SPOOL command 119
 holding
 a file before output processing 41
 files from further processing 43,118

I
 IMMSG operand, of SET command (CP) 111
 INCLUDE command, summary 149
 INDEX operand, for LOADVFCB command 81
 INDICATE command
 described 72
 LOAD operand 72
 USER operand 72
 initial program load (IPL)
 automatic, suppression of 84
 of a virtual machine operating system
 75
 stopping 75
 input/output operations, tracing 135

INSTRUCT operand, of TRACE command (CP)
 135
 interrupts
 clearing from a virtual machine 108
 device-end, specifying virtual 106
 external, simulating 71
 for a virtual machine 107
 tracing 135
 I/O devices, virtual, defining for your
 virtual machine 50
 I/O operand, of TRACE command (CP) 135
 IPL command
 CLEAR operand 75
 described 75
 for named systems 75
 NOCLEAR operand 75
 PARM operand 76
 STOP operand 75
 summary 144
 ISAM operand, of SET command (CP) 111

K
 keys
 program function, controlling 113
 storage
 displaying 61
 printing 67

L
 line editing symbols, controlling 111
 line length, controlling at your terminal
 134
 LINE operand, of DEFINE command (CP) 51
 LINEDEL operand, of TERMINAL command 133
 LINEDIT operand, of SET command (CP) 111
 LINEND operand, of TERMINAL command 133
 LINES operand, of QUERY command (CP) 95
 LINESIZE operand, of TERMINAL command 134
 LINK command
 described 77
 passwords 78
 summary 144
 linking, a device to your virtual machine
 77
 LINKS operand, of QUERY command (CP) 95
 LISTDS command, summary 149
 LISTFILE command, summary 149
 LISTIO command, summary 149
 LKED command, summary 149
 LOAD
 command, summary 149
 operand, of INDICATE command 72
 load key, simulation of via IPL command 75
 LOADBUF command, summary 144
 LOADMOD command, summary 149
 LOADVFCB command
 described 81
 FCB operand 81
 INDEX operand 81
 summary 145
 LOCATE command, summary 145
 LOCK command, summary 145
 logging off of your virtual machine 82
 logging on your virtual machine 84

logical editing symbols
 altering 132
 controlling 132
 controlling use of 111
 logically connected terminal 57
 LOGMSG operand, of QUERY command (CP) 104
 LOGOFF command
 described 82
 HOLD operand 82
 summary 145
 LOGON command
 described 84
 MASK operand 84
 NOIPL operand 84
 summary 145
 logon procedure, hiding your password
 during 84

M
 MACLIB command, summary 149
 MASK operand
 of LOGON command 84
 of TERMINAL command 133
 masking
 of passwords, controlling 132
 password during logon 84
 MESSAGE
 command
 described 86
 OPERATOR operand 86
 summary 145
 message handling, error, handling 112
 messages
 controlling transmission of 111
 sending to other users 86
 minimum truncation of commands 18
 MODE operand, of TERMINAL command 14,134
 mode setting of virtual console 13
 modifying
 spooling control options 118
 storage, using ADSTOP command 37
 virtual device attributes 118
 MODMAP command, summary 150
 MONITOR command, summary 145
 MOVEFILE command, summary 150
 MSG
 command, summary 152
 operand, of SET command (CP) 110

N
 NAME operand
 of CHANGE command (CP) 42
 of CLOSE command 44
 named systems, IPL command for 75
 NAMES operand, of QUERY command (CP) 104
 NCPDUMP command, summary 150
 NETWORK command, summary 145
 NOCLEAR operand, of IPL command 75
 NOCONT operand, of SPOOL command 119
 NOEOF operand, of SPOOL command 120
 NOHOLD operand
 of CHANGE command (CP) 41
 of CLOSE command 43
 of SPOOL command 120

NOIPL operand, of LOGON command 84
 NORUN operand, of TRACE command (CP) 136
 notational conventions 29
 NOTERM operand, of SPOOL command 122
 NOTRANS operand, of SET command (CP) 112
 NOTREADY command
 described 88
 summary 145

O
 OFF operand
 of SPOOL command 121
 of the ADSTOP command 37
 of TRACE command (CP) 136
 operands, command 17
 operating system
 for a virtual machine, passing
 parameters to 76
 initial program load for 75
 operator, sending messages to 86
 OPERATOR operand, of MESSAGE command 86
 OPTION command, summary 150
 options, command 17
 ORDER
 command (CP)
 described 89
 summary 145
 command (RSCS), summary 152
 OS Indexed Sequential Access Method (ISAM)
 111

P
 PAGEX operand, of SET command (CP) 112
 parameters, passing to a virtual machine
 operating system 76
 PARM operand, of IPL command 76
 passing parameters to your virtual machine
 operating system 76
 password
 controlling masking of 132
 hiding during logon procedure 84
 with the LINK command 78
 PFnn operand
 of QUERY command (CP) 96
 of SET command (CP) 113
 PLIC command, summary 150
 PLICR command, summary 150
 PLIOPT command, summary 150
 PRB command, summary 150
 PRINT command, summary 150
 printer (3211), virtual, specifying forms
 controls for 81
 printer files, virtual, altering the
 attributes of 41
 PRINTER operand
 of CHANGE command (CP) 41
 of CLOSE command 43
 of DEFINE command (CP) 49
 of PURGE command (CP) 91
 of QUERY command (CP) 95
 of SPOOL command 118
 of TRACE command (CP) 136
 printing
 contents of virtual machine components
 67

Program Status Word (PSW) 67
 virtual storage keys 67
 virtual storage locations 67
 PRIV operand, of TRACE command (CP) 135
 privilege classes
 for CP commands
 defined 21
 summarized 21
 privileged instructions, tracing 135
 PROB command, summary 150
 program function keys, controlling 113
 PROGRAM operand, of TRACE command (CP) 135
 Program Status Word (PSW)
 changing your virtual machine 125
 displaying 63
 printing 67
 PSERV command, summary 150
 PSW operand, of DISPLAY command 63
 PUNCH
 command, summary 150
 operand
 of CHANGE command (CP) 41
 of CLOSE command 43
 of DEFINE command (CP) 49
 of PURGE command (CP) 91
 of QUERY command (CP) 95
 of SPOOL command 118
 punch files, virtual, altering the
 attributes of 41
 PURGE
 command (CP)
 ALL operand 91
 CLASS operand 91
 described 91
 PRINTER operand 91
 PUNCH operand 91
 READER operand 91
 summary 145
 command (RSCS), summary 152
 operand
 of CLOSE command 44
 of SPOOL command 121
 purging
 files 43,118
 virtual devices from your system 91

Q

QUERY
 command (CMS), summary 150
 command (CP)
 ALL operand 95
 CHANNELS operand 94
 CLASS operand 96
 CONSOLE operand 94
 DASD operand 94
 described 93
 FILES operand 94
 GRAF operand 94
 LINES operand 95
 LINKS operand 95
 LOGMSG operand 104
 NAMES operand 104
 PFnn operand 96
 PRINTER operand 95
 PUNCH operand 95
 READER operand 95
 SET operand 94

STORAGE operand 95
 summary 145
 TAPES operand 95
 TERMINAL operand 94
 TIME operand 94
 UR operand 95
 USERS operand 104
 VIRTUAL operand 94
 command (RSCS), summary 152

R
 READCARD command, summary 150
 reader files, virtual, altering the
 attributes of 41
 READER operand
 of CHANGE command (CP) 41
 of CLOSE command 43
 of DEFINE command (CP) 49
 of PURGE command (CP) 91
 of QUERY command (CP) 95
 of SPOOL command 118
 reading, an entire file continuously 118
 read/write access, with LINK command 78
 READY command
 described 106
 summary 145
 real computer
 reset button, simulating 128
 restart button, simulating 128
 reclaiming transmitted files 140
 registers
 extended control
 displaying 62
 printing 67
 floating-point
 displaying 62
 printing 67
 general
 displaying 62
 printing 67
 RELEASE command, summary 150
 Remote Spooling Communications Subsystem
 receiving files 123
 TAG command 129
 transmitting files 123
 remote terminal, connecting to a virtual
 machine 57
 removing a virtual device from your virtual
 machine 54
 RENAME command, summary 150
 REPEAT command, summary 145
 REQUEST command
 described 107
 summary 145
 RESET
 command
 described 108
 summary 145
 operand, of SYSTEM command 128
 reset button, simulating 128
 restart button, simulating 128
 RESTART operand, of SYSTEM command 128
 resume
 execution of your virtual machine 40
 use of your virtual machine after
 disconnecting it 59

REWIND command
 described 109
 summary 145
 rewinding a real tape 109
 RSCS (see Remote Spooling Communications Subsystem)
 command summary 152
 RSERV command, summary 150
 RUN
 command, summary 150
 operand
 of SET command (CP) 111
 of TRACE command (CP) 136

S
 saved systems, IPL command for 75
 SAVENCP command, summary 150
 SAVESYS command, summary 145
 saving, virtual machine data 125
 SCRIPT command, summary 150
 sending messages
 to other users 86
 to the operator 86
 service representative, described 22
 SET
 command (CMS), summary 151
 command (CP)
 ACNT operand 110
 ASSIST operand 113
 described 110
 ECMODE operand 111
 EMSG operand 112
 IMSG operand 111
 ISAM operand 111
 LINEDIT operand 111
 MSG operand 110
 NOTRANS operand 112
 PAGEX operand 112
 PFnn operand 113
 RUN operand 111
 summary 145
 TIMER operand 112
 WNG operand 111
 operand, of QUERY command (CP) 94
 SETKEY command, summary 151
 SHUTDOWN command, summary 146
 simulating
 not ready status for a virtual device 88
 the reset button on a real computer 128
 the restart button on a real computer 128
 SIO operand, of TRACE command (CP) 135
 SLEEP command
 described 117
 summary 146
 SORT command, summary 151
 SPACE command, summary 146
 specifying, device-end interrupt for a virtual device 106
 spool class
 for a file, changing 41
 modifying 118
 SPOOL command
 CLASS operand 119
 CLOSE operand 121
 CONSOLE operand 119
 CONT operand 119
 COPY operand 121
 described 118
 EOF operand 120
 FOR operand 120
 HOLD operand 119
 NOCONT operand 119
 NOEOF operand 120
 NOHOLD operand 120
 NOTERM operand 122
 OFF operand 121
 PRINTER operand 118
 PUNCH operand 118
 PURGE operand 121
 READER operand 118
 START operand 121
 STOP operand 121
 summary 146
 SYSTEM operand 120
 TERM operand 121
 TO operand 120
 use with TAG command 123
 spool command options, set for a virtual device 43
 spool file
 altering TAG information 129
 changing filename for 42
 changing the attributes of 41
 reordering 89
 spoolid 122
 TAG command 129
 transmitting to remote location 123,129
 spoolid number 122
 spooling
 control, options, modifying 118
 operator, described 22
 SSERV command, summary 151
 START
 command (CMS), summary 151
 command (CP), summary 146
 command (RSCS), summary 152
 operand, of SPOOL command 121
 STAT command, summary 151
 STATE command, summary 151
 STATEW command, summary 151
 status
 virtual machine, determining 93
 words, tracing 135
 STCP command, summary 146
 STOP operand
 of IPL command 75
 of SPOOL command 121
 stopping, initial program load (IPL) procedure 75
 storage
 clearing to zeros 75
 keys, printing 67
 locations, virtual, displaying 61
 modification of 37
 storage keys, displaying 61
 storage locations
 virtual machine
 altering 125
 displaying 61
 printing 67

STORAGE operand
of DEFINE command (CP) 51
of QUERY command (CP) 95

STORE command
described 125
summary 146

storing
CPU status for virtual machine 125
virtual machine data 125

summary
of CMS commands 148
of CP commands 143
of RSCS commands 152

suppressing, automatic IPL of a virtual machine operating system 84

SVC operand, of TRACE command (CP) 135

SVCTRACE command, summary 151

switching command environments 14
using the Attention (ATTN) key 13

SYNONYM command, summary 151

SYSTEM
command
CLEAR operand 128
described 128
RESET operand 128
RESTART operand 128
summary 146
operand, of SPOOL command 120

system analyst, described 22

system operator, primary, described 22

system programmer, described 22

system resource operator, described 22

system resources, indication of 72

T

TAG command
altering TAG information 129
described 129
summary 146
transmitting files to remote location 129
use with SPOOL command 123

TAPE command, summary 151

TAPEMAC command, summary 151

tapes, rewinding 109

TAPES operand, of QUERY command (CP) 95

TAPPDS command, summary 151

TERM operand, of SPOOL command 121

TERMINAL
command
APL operand 133
ATTN operand 134
CHARDEL operand 132
described 132
ESCAPE operand 133
LINEDEL operand 133
LINEND operand 133
LINESIZE operand 134
MASK operand 133
MODE operand 134
summary 146
used to set the mode of your virtual console 14
operand
of QUERY command (CP) 94
of TRACE command (CP) 136

terminals
console, disconnecting from your virtual machine 59
control of attention handling 132
input/output processing, controlling 132
line length, controlling 132
remote, connecting to a virtual machine 57
session
commands used to control, summary of 143
determining the length of time of 93
testing using the ECHO command 70

terminating
console input/output 118
processing of files 43

TESTCOB command, summary 151

TESTFORT command, summary 151

testing, terminal 70

time, determining length for a terminal session 93

TIME operand, of QUERY command (CP) 94

timer, virtual, controlling 112

TIMER operand
of DEFINE command (CP) 50
of SET command (CP) 112

TO operand
of SPOOL command 120
of TRANSFER command 140

TRACE
command (CP)
ALL operand 135
BOTH operand 136
BRANCH operand 135
CCW operand 135
CSW operand 135
described 135
END operand 136
EXTERNAL operand 135
INSTRUCT operand 135
I/O operand 135
NORUN operand 136
OFF operand 136
PRINTER operand 136
PRIV operand 135
PROGRAM operand 135
RUN operand 136
SIO operand 135
summary 146
SVC operand 135
TERMINAL operand 136
command (RSCS), summary 152

tracing, virtual machine activity 135

TRANSFER command
ALL operand 140
CLASS operand 140
described 140
FROM operand 140
summary 146
TO operand 140

transferring
files
to a virtual reader 140
to another user 118

translating, CCWs, controlling 112

truncating, commands 18,29

TXTLIB command, summary 151

TYPE command, summary 151
 T2305 operand, of DEFINE command (CP) 51
 T2314 operand, of DEFINE command (CP) 51
 T2319 operand, of DEFINE command (CP) 51
 T3330 operand, of DEFINE command (CP) 51
 T3340 operand, of DEFINE command (CP) 51
 T3350 operand, of DEFINE command (CP) 51

U
 UNLOCK command, summary 146
 UPDATE command, summary 151
 UR operand, of QUERY command (CP) 95
 USER operand
 of INDICATE command 72
 of QUERY command (CP) 104

V
 VARY command, summary 146
 virtual console
 disconnecting from your virtual machine 59
 mode setting 13
 spooling 119
 virtual devices
 closing 118
 defining 49
 detaching from your virtual machine 54
 determining number of for your virtual machine 93
 determining the status of 93
 I/O, defining for your virtual machine 50
 purging from your system 91
 removing from your virtual machine 54
 simulating not ready status for 88
 specifying a device-end interrupt for 106
 virtual machine
 activity, tracing 135
 configuration, altering 49
 determining status of 93
 device, linking to 77
 disconnecting 82
 execution, resuming 40
 logging off of 82
 logging on 84
 operating system, passing parameters to 76
 placing in a dormant state 117
 setting control functions for 110
 storing information from 125
 Virtual Machine (VM) environment, entering CP commands from 35
 Virtual Machine Facility/370, basic description of 11
 VIRTUAL operand, of QUERY command (CP) 94
 virtual printer (3211), specifying forms controls for 81
 virtual storage locations
 changing 125
 printing 67
 virtual timer, controlling 112
 VM environment 13
 entering a CP command from 48
 VMFDUMP command, summary 151
 VMFLOAD command, summary 151
 VM/VS handshaking, control of 112
 VM/370 (*see* Virtual Machine Facility/370)
 VSANGEN command, summary 151
 VSBASIC command, summary 151
 VSBUTIL command, summary 151

W
 WARNING command, summary 146
 WNG operand, of SET command (CP) 111
 writing, comments in VM/370 17

Z
 ZAP command, summary 151
 zeros, clearing storage to 75,128

1
 1403 operand, of DEFINE command (CP) 50

3
 3211 operand, of DEFINE command (CP) 50
 3211 printer, virtual, specifying forms controls for 81

**READER'S
COMMENT
FORM**

Title: IBM Virtual Machine Facility/370:
CP Command Reference for
General Users

Order No. GC20-1820-0

Please check or fill in the items; adding explanations/comments in the space provided.

Which of the following terms best describes your job?

- | | | | |
|--|--|---|--|
| <input type="checkbox"/> Customer Engineer | <input type="checkbox"/> Manager | <input type="checkbox"/> Programmer | <input type="checkbox"/> Systems Analyst |
| <input type="checkbox"/> Engineer | <input type="checkbox"/> Mathematician | <input type="checkbox"/> Sales Representative | <input type="checkbox"/> Systems Engineer |
| <input type="checkbox"/> Instructor | <input type="checkbox"/> Operator | <input type="checkbox"/> Student/Trainee | <input type="checkbox"/> Other (explain below) |

How did you use this publication?

- | | | | |
|--|---|-----------------------------------|--|
| <input type="checkbox"/> Introductory text | <input type="checkbox"/> Reference manual | <input type="checkbox"/> Student/ | <input type="checkbox"/> Instructor text |
| <input type="checkbox"/> Other (explain) _____ | | | |

Did you find the material easy to read and understand? Yes No (explain below)

Did you find the material organized for convenient use? Yes No (explain below)

Specific criticisms (explain below)

- Clarifications on pages _____
- Additions on pages _____
- Deletions on pages _____
- Errors on pages _____

Explanations and other comments:

Trim Along This Line

Thank you for your cooperation. No postage necessary if mailed in the U.S.A.

YOUR COMMENTS PLEASE . . .

Your views about this publication may help improve its usefulness; this form will be sent to the author's department for appropriate action. Using this form to request system assistance and/or additional publications or to suggest programming changes will delay response, however. For more direct handling of such requests, please contact your IBM representative or the IBM Branch Office serving your locality. Your comments will be carefully reviewed by the person or persons responsible for writing and publishing this material. All comments or suggestions become the property of IBM.

FOLD

FOLD

FIRST CLASS
PERMIT NO. 172
BURLINGTON, MASS.

BUSINESS REPLY MAIL
NO POSTAGE STAMP NECESSARY IF MAILED IN U.S.A.



POSTAGE WILL BE PAID BY

IBM CORPORATION
VM/370 PUBLICATIONS
24 NEW ENGLAND EXECUTIVE PARK
BURLINGTON, MASS. 01803

FOLD

FOLD



International Business Machines Corporation
Data Processing Division
1133 Westchester Avenue, White Plains, New York 10604
(U.S.A. only)

IBM World Trade Corporation
821 United Nations Plaza, New York, New York 10017
(International)