

File No. S370-36
Order No. SC19-6211-0

Program Product

**IBM Virtual
Machine/System Product:
CP Command Reference
for General Users**

IBM

Program Product

IBM Virtual Machine/System Product: CP Command Reference for General Users

Program Number 5664-167

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

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

PREREQUISITE PUBLICATIONS

IBM Virtual Machine/System Product:

Introduction, Order No. GC19-6200

Terminal User's Guide, Order No. GC19-6206

COREQUISITE PUBLICATIONS

IBM Virtual Machine/System Product:

System Messages and Codes, Order No. SC19-6204



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

First Edition (September 1980)

This edition (SC19-6211) applies to the IBM Virtual Machine/System Product unless otherwise indicated in new editions or Technical Newsletters. Changes are continually made to the information contained herein; before using this publication in connection with the operation of IBM systems, consult the IBM System/370 and 4300 Processors Bibliography, GC20-0001, for the editions that are applicable and current.

It is possible that this material may contain reference to, or information about, IBM products (machines and programs), programming, or services that are not announced in your country. Such references or information must not be construed to mean that IBM intends to announce such IBM products, programming, or services in your country.

Publications are not stocked at the address given below; 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 Programming Publications, Dept. G60, P.O. Box 6, Endicott, New York, U.S.A. 13760. IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation whatever. You may, of course, continue to use the information you supply.

Preface

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

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

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

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

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

Part 1 has three sections:

"Section 1. The CP Command Syntax" describes the VM/SP command environments and the general structure of the CP command language.

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

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

Part 2 has two sections:

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

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

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

In this publication, the following terminology is used:

- The term 'cylinder' is used to describe DASD space on Direct Access Storage Devices (count-key-data devices) supported by the VM/SP System Control Program. The term 'block' is used to describe DASD space on FB-512 devices. Specifically, FB-512 devices are the IBM 3310 and 3370 Direct Access Devices employing fixed block mode. The term 'DASD space' may appear in text when there is no need to differentiate between count-key-data devices and FB-512 devices.
- "2305" refers to the IBM 2305 Fixed Head Storage, Models 1 and 2.
- The term, "3270", is used in this publication to refer to a series of display devices, namely, the IBM 3275, 3276, 3277, 3278, and 3279 Display Stations. A specific device type is used only when a distinction is required between device types. Information about display terminal usage also applies to the IBM 3138, 3148, 3158, 3033, and 3036 Display Consoles when used in display mode, unless otherwise noted.

- Any information pertaining to the IBM 3284 or 3286 printer also pertains to the IBM 3262, 3287, 3288 and 3289 printers unless otherwise noted.
 - "3330" refers to the IBM 3330 Disk Storage Models 1, 2, 11; the IBM 3333 Disk Storage and Control Models 1 and 11; and the 3350 Direct Access Storage operating in 3330/3333 Model 1 or 3330/3333 Model 2 compatibility mode.
 - "3340" refers to the IBM 3340 Disk Storage, Models A2, B1, and B2; and the 3344 Direct Access Storage, Model B2.
 - "3350" refers to the IBM 3350 Direct Access Storage, Models A2 and B2, in native mode.
 - "3380" refers to IBM 3380 Direct Access Storage. Information on the 3380 is for planning purposes only until the availability of that device.
 - "VM/370 hardware assist" refers to both the virtual machine assist function and VM/370 Extended Control-Program Support. Some form of the VM/370 hardware assist is available on all VM/370 supported System/370 systems except the 155 II, 165-III, and 4331. Refer to IBM Virtual Machine/System Product Planning and System Generation Guide, Order No. SC19-6201, for further device information on the VM/370 hardware assist and ECPS features.
 - Any information pertaining to the IBM 2741 terminal also applies to the IBM 3767 terminal, Model 1, operating as a 2741, unless otherwise specified.
 - The term "processor" used in this publication means the main processor which is the processor with I/O handling capabilities. The term "attached processor" means a processor that has no I/O capabilities; an attached processor is always linked to a main processor for utilization. VM/SP also supports real multiprocessor (MP) configurations on the 158MP, 168MP, and 3033MP processors. In a MP system, the term "IPL processor" refers to the processor that is initialized first during system generation. The term "nonIPL processor" refers to the processor that is initialized after the IPL processor. Both the IPL and nonIPL processors in a real MP configuration have I/O capabilities.
 - "3800" refers to the IBM 3800 Printing Subsystem.
 - "3850" refers to the IBM 3850 Mass Storage System.
 - Unless otherwise noted, the term "VSE" refers to the combination of the DOS/VSE system control program and the VSE/Advanced Functions program product.
- In certain cases, the term DOS is still used as a generic term. For example, the disk packs initialized for use with VSE or any predecessor DOS or DOS/VS system may be referred to as DOS disks.
- For a glossary of VM/SP terms, see the IBM Virtual Machine/System Product Glossary and Master Index, Order No. GC19-6207.

PREREQUISITE PUBLICATIONS

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

The physical operation of the virtual machine console is covered in the IBM Virtual Machine/System Product Terminal User's Guide, Order No. GC19-6206.

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

ASSOCIATED PUBLICATIONS

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

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

IBM Virtual Machine/System Product

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

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

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

IBM Virtual Machine Facility/370:

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

Interactive Problem Control System (IPCS) User's Guide, Order No. GC20-1823

In addition, VM/SP users can have enhanced networking facilities provided in the RSCS Networking program product (5748-XP1). Enhanced interactive problem control is provided by the VM/IPCS Extension program product (5748-SA1). The following publications contain information about the command languages of the RSCS Networking program product and the VM/IPCS Extension program product:

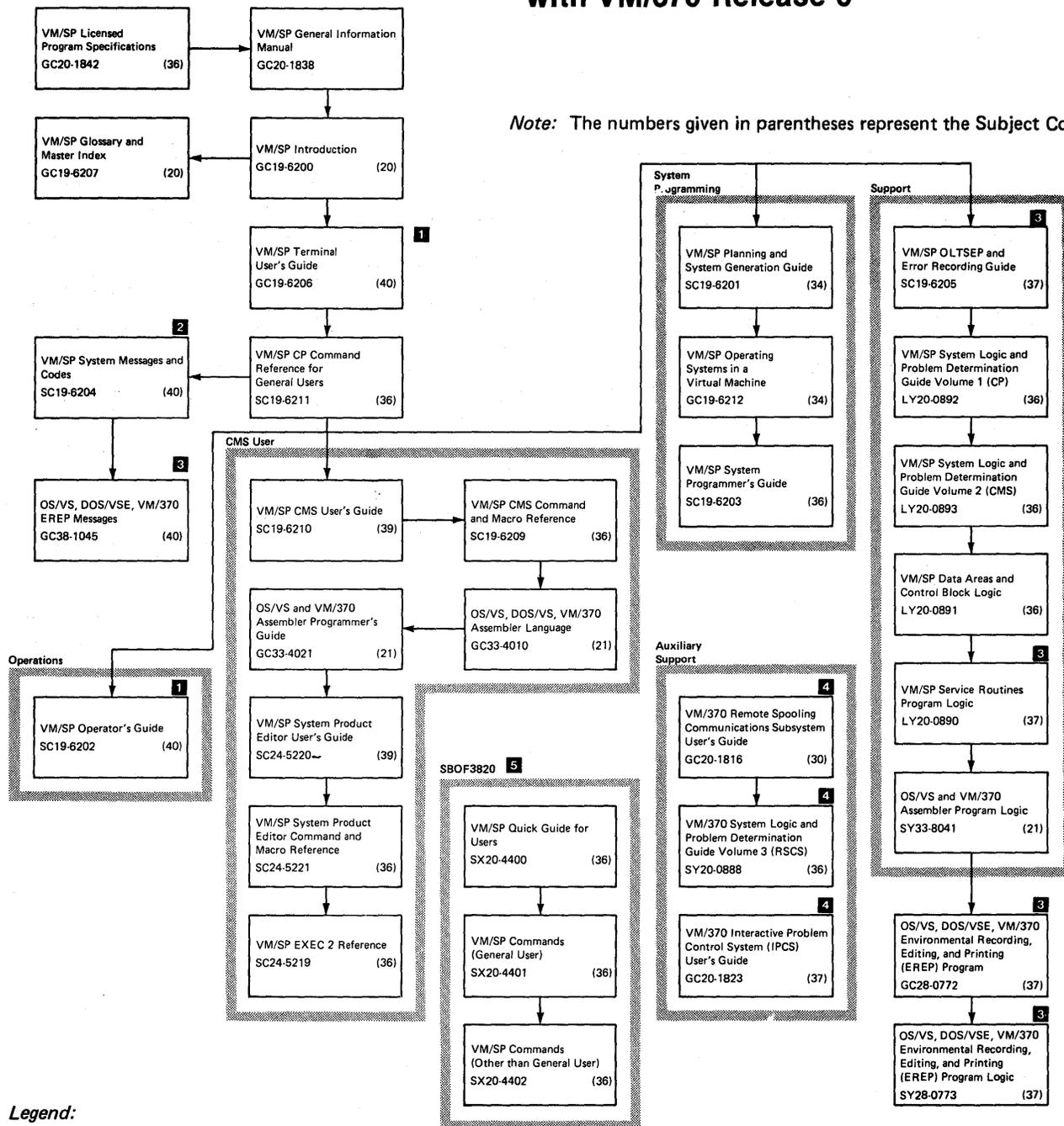
VM/370 RSCS Networking Program Reference and Operations Guide, Order No. SH24-5005

VM/IPCS Extension User's Guide and Reference, Order No. SC34-2020

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

Publications that support VM/SP as used in conjunction with VM/370 Release 6

Note: The numbers given in parentheses represent the Subject Code.



Legend:

- 1** For SNA terminal users, the prerequisite publication is: *VM/VCNA Installation, Operations, and Terminal Use*, Order No. SC27-0502.
- 2** All users of virtual machines must use the *VM/SP System Messages and Codes* publication.
- 3** Contains information on VM/SP EREP support.
- 4** VM/370 Release 6 components. However, the IPCS Extension Program Product (5748-SA1) and the RSCS Networking Program Product (5748-XP1) are recommended for use with VM/SP.
- 5** If you want all three of the Reference Summary publications, use the SBOF number when ordering.

Figure 1. The Virtual Machine/System Product Library

Contents

| | | | |
|--|----|--|-----|
| PART 1. GUIDANCE INFORMATION | 1 | *. | 40 |
| SECTION 1. THE CP COMMAND SYNTAX | 3 | #CP. | 41 |
| VM/SP Command Environments | 3 | ADSTOP | 43 |
| VM/SP CP Command Structure | 3 | ATTN | 45 |
| The Command Name | 4 | BEGIN. | 46 |
| The Command Operands | 4 | CHANGE | 47 |
| Comments in the CP Command Language. | 4 | CLOSE. | 51 |
| Character Set Usage. | 4 | COUPLE | 55 |
| SECTION 2. THE CP COMMAND LANGUAGE | 7 | CP | 57 |
| Privilege Classes for CP Commands. | 7 | DEFINE | 58 |
| The Class G Commands | 9 | DETACH | 66 |
| The Class Any Commands | 9 | DETACH CHANNEL | 69 |
| SECTION 3. CP COMMAND USAGE. | 11 | DIAL | 70 |
| Controlling Input and Output Functions | 12 | DISCONN. | 72 |
| Virtual Disks. | 13 | DISPLAY. | 74 |
| Permanent Virtual Disks. | 14 | DUMP | 80 |
| Temporary Virtual Disks. | 14 | ECHO | 83 |
| Sharing Virtual Disks. | 14 | EXTERNAL | 84 |
| Virtual Unit Record Devices. | 15 | INDICATE | 85 |
| Virtual Unit Record Spooling | 15 | IPL. | 88 |
| Spool File Characteristics | 16 | LINK | 92 |
| Virtual Console Spooling | 18 | LOADVFCB | 96 |
| Reordering and Purging Spool Files | 18 | LOGOFF | 98 |
| Dedicated Devices. | 19 | LOGON. | 100 |
| Dedicated Channels | 20 | MESSAGE. | 102 |
| Controlling the Virtual Machine. | 21 | NOTREADY | 104 |
| Loading an Operating System. | 21 | ORDER. | 105 |
| Simulating Interrupts. | 22 | PURGE. | 107 |
| Entering CP Commands | 23 | QUERY. | 109 |
| Entering CP Commands when the Virtual | 23 | QUERY Command for Class G Users. | 110 |
| Entering CP Commands from the Virtual | 23 | QUERY Command for All Classes of Users | 127 |
| Console Read Environment. | 24 | (Except Class Any) | 127 |
| Entering CP Commands from CP Console | 25 | READY. | 130 |
| Function Mode | 25 | REQUEST. | 131 |
| Reconfiguring the Virtual Machine. | 25 | RESET. | 132 |
| Testing and Debugging of Programs. | 26 | REWIND | 133 |
| Stopping Execution of Your Virtual | 26 | SCREEN | 134 |
| Machine | 26 | SEND | 137 |
| Displaying Virtual Storage | 27 | SET. | 138 |
| Terminal Output. | 27 | SLEEP. | 151 |
| Byte Alignment on Terminal Output. | 28 | SMSG | 152 |
| Printer Output | 29 | SPOOL. | 153 |
| Altering Virtual Storage | 30 | STORE. | 163 |
| Tracing Virtual Machine Activity | 31 | SYSTEM | 166 |
| PART 2. REFERENCE INFORMATION. | 33 | TAG. | 167 |
| SECTION 4. NOTATIONAL CONVENTIONS. | 35 | TERMINAL | 170 |
| SECTION 5. FORMAT OF CP COMMANDS | 39 | TRACE. | 176 |
| | | TRANSFER | 181 |
| | | VMDUMP | 184 |
| | | APPENDIX: CP COMMAND SUMMARY | 189 |
| | | INDEX. | 195 |

FIGURES

| | | | |
|-----------|--|------------|--|
| Figure 1. | The Virtual Machine/System Product Library.....vi | Figure 6. | Action Taken on CLOSE for a Virtual Input Device.....53 |
| Figure 2. | Character Sets and Their Contents.....5 | Figure 7. | Action Taken on CLOSE for a Virtual Output Device.....53 |
| Figure 3. | CP Privilege Class Descriptions.....8 | Figure 8. | Available Form Width Codes for 3800 Spooling.....64 |
| Figure 4. | Controlling the Virtual Machine During a Terminal Session.....11 | Figure 9. | Password Requirements on the LINK Command.....94 |
| Figure 5. | System Action to #CP Command.....42 | Figure 10. | 3270 High Intensity Features.....175 |
| | | Figure 11. | CP Command Summary.....189 |

Part 1. Guidance Information

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

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

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

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

Section 1. CP Command Syntax

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

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

VM/SP Command Environments

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

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

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

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

VM/SP CP Command Structure

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

```
[ command name | [ operand... ] ]
```

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

THE COMMAND NAME

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

THE COMMAND OPERANDS

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

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

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

COMMENTS IN THE CP COMMAND LANGUAGE

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

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

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

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

CHARACTER SET USAGE

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

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

Figure 2. Character Sets and Their Contents

Section 2. The CP Command Language

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

Privilege Classes for CP Commands

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

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

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

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

This publication is a reference manual for only the class G and class 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/SP system. Class A is assigned to the user at the VM/SP system console during IPL. The primary system operator is responsible for the availability of the VM/SP system and its communication lines and resources. In addition, the class A user controls system accounting, broadcast messages, virtual machine performance options, and other command operands that affect the overall performance of VM/SP. The system operator controls operation of the real machine using the system control panel and console device.</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 | <p><u>System Resource Operator</u>: The class B user controls allocation and deallocation of all the real resources of the VM/SP system, except those controlled by the primary system operator and spooling operator.</p> |
| C ¹ | <p><u>System Programmer</u>: The class C user updates certain functions of the VM/SP system. The system programmer can modify real storage in the real machine.</p> |
| D ¹ | <p><u>Spooling Operator</u>: The class D user controls spool data files and specific functions of the system's unit record equipment.</p> |
| E ¹ | <p><u>System Analyst</u>: The class E user displays the contents of real storage, performs the functions required to generate saved systems and discontinuous saved segments, and controls the collection and recording of performance measurement data. This class of user can display the contents of specified real storage areas on the virtual operator's console or on a spooled virtual printer, but cannot modify real storage.</p> |
| F ¹ | <p><u>Service Representative</u>: The class F user obtains, and examines, in detail, certain data about input and output devices connected to the VM/SP system. The service representative can establish intensive recording mode for one I/O device at a time and can cause the recording of repressable machine check errors to be initiated or resumed.</p> |
| G ² | <p><u>General User</u>: The class G user controls functions associated with the execution of his virtual machine. A general user cannot display or modify real storage.</p> |
| Any ² | <p>The Any classification is given to certain CP commands that are available to any user. These are primarily for the purpose of gaining and relinquishing access to the VM/SP system.</p> |
| H | <p>Reserved for IBM use.</p> |

¹Described in the VM/SP Operator's Guide.
²Described in this publication.

Figure 3. CP Privilege Class Descriptions

THE CLASS G COMMANDS

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

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

THE CLASS ANY COMMANDS

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

Section 3. CP Command Usage

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

| STEP ACTION | FUNCTION | COMMAND DESCRIPTION |
|---|--|---|
| 1. User must fill out request provided by by installation ¹ | Establish userid, user classes, passwords, storage requirements, disk space, and I/O devices | None |
| 2. Installation builds VM/SP directory entry for system user from information ¹ | Directory entry provides CP with information necessary to create a virtual machine | CMS DIRECT command |
| 3. User can activate terminal and begin work session ² | Enter CP commands to perform the function required | Available class G commands (Figure 11) |
| | a. activate terminal | LOGON command (Figure 11) |
| | b. transfer and store data | LINK, SPOOL, STORE, and TRANSFER commands (Figure 11) |
| | c. record and control data | CHANGE, CLOSE, DISPLAY, DUMP, ORDER, PURGE, QUERY, and SPOOL commands (Figure 11) |
| ¹ Unless the installation or the users change system requirements, steps 1 and 2 do not have to be repeated every time a user activates a terminal after the initial session. | | |
| ² Throughout the rest of this publication, the text is interspersed with examples of VM/SP commands and system responses. All user input is shown in lowercase, and all system responses are shown in uppercase. | | |

Figure 4. Controlling the Virtual Machine During a Terminal Session (Part 1 of 2)

| STEP ACTION | FUNCTION | COMMAND DESCRIPTION |
|-------------|---|--|
| | d. reconfigure the virtual machine | DEFINE and DETACH commands (Figure 11) |
| | e. communicating with the virtual machine | #CP, ADSTOP, ATTN, BEGIN, EXTERNAL, IPL, REQUEST, SET commands (Figure 11) |
| | f. handling errors | TRACE, DISPLAY, and DUMP commands (Figure 11) |
| | g. end session | LOGOFF command (Figure 11) |

Figure 4. Controlling the Virtual Machine During a Terminal Session
(Part 2 of 2)

Controlling Input and Output Functions

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

```

(1)    USER VMUSER 123456 512K 2M G
        ACCOUNT 87321 BIN14           (2)
(3)    OPTION ECMODE
        CONSOLE 01F 3215              (4)
        SPOOL 00C 2540 R
(5)    SPOOL 00D 2540 P
        SPOOL 00E 1403
(6)    MDISK 130 2314 050 050 VDISK1 WR RWPASSWD
        MDISK 151 3330 001 020 VDISK2 WR RWPASSWD
        LINK CMSSYS 440 190 RR

```

where:

- (1) The userid of this virtual machine is VMUSER, and the password is 123456. The virtual machine's storage size is defined as 512K; however, this can be redefined up to 2 megabytes if, during a operating session, the need for additional storage arises (see the topic on "Reconfiguring the Virtual Machine"). The user privilege class is G.
- (2) VM/SP generates accounting data for processor time and I/O device usage. This data is charged to a virtual machine by account number, which in this case is 87321. BIN14 is a distribution code that is printed or punched on spooled printer and punch output. It usually denotes a location where the user can pick up spooled output of a terminal session.
- (3) Option ECMODE allows the virtual machine to use the complete set of virtual System/370 control registers and the dynamic address translation feature of the System/370.

- (4) The virtual address of the virtual machine console is 01F, and your operating system corresponds with the terminal as if it were a 3215/3210.
- (5) The virtual unit record devices (reader, punch, and printer) are to be spooled and their addresses, as far as the virtual machine is concerned, are 00C, 00D, and 00E, respectively.
- (6) In this configuration, a 50 cylinder read/write minidisk with a virtual address of 130 is located on cylinders 50 through 99 of a real 2314 volume labeled VSDISK1. Similarly, a 20 cylinder read/write minidisk with a virtual address of 151 is located on cylinders 1 through 20 of a real 3330 volume labeled VDISK2. The last entry provides a link to a sharable CMS system volume so the user can use such CMS functions as the CMS editor or the EXEC processor. The LINK entry makes the virtual disk at virtual address 440 available to your virtual machine in read-only mode at virtual address 190.

Notes:

1. The read password entry (RR) on the LINK command means that anyone with the proper password can share the use of the minidisk in read-only mode.
2. The write access mode (WR) on the LINK command means that you can write to the disk as long as no one else has a link to it; if someone else links to it before you log on, you still have read-only access.
3. For detailed information on directory entries, see the section "Directory Control Statements" in the VM/SP Planning and System Generation Guide.

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

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

VIRTUAL DISKS

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

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

PERMANENT VIRTUAL DISKS

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

TEMPORARY VIRTUAL DISKS

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

```
define t2319 as 133 cyl 15
```

In the previous example, a virtual 2319 disk comprising 15 cylinders is allocated to your virtual machine at virtual address 133. You can then notify your operating system of the additional storage space using the appropriate control statements or commands.

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

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

```
detach 133
```

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

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

SHARING VIRTUAL DISKS

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

```
link to smith 330 as 134 rr rpasswd
```

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

Notes:

1. For further information on virtual disks, consult VM/SP CMS User's Guide.
2. If your installation is using the password suppression facility, you cannot specify the password on the same command line. For further information, refer to discussion of the LINK command in Section 5 of this publication.

VIRTUAL UNIT RECORD DEVICES

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

VIRTUAL UNIT RECORD SPOOLING

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

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

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

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

SPOOL FILE CHARACTERISTICS

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

```
spool 00e class g
```

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

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

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

For instance, you can issue:

```
change printer form fedtax form statetax nohold
```

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

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

```
change printer 1234 class m
```

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

```
query printer all
```

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

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

```
spool printer hold
```

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

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

```
close printer nohold
```

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

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

```
query printer
```

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

```
change printer 0246 nohold
```

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

```
spool printer nohold
```

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

```
spool reader hold
```

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

```
spool reader nohold
```

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

```
purge reader all
```

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

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

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

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

VIRTUAL CONSOLE SPOOLING

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

```
spool console start
```

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

```
spool console start noterm
```

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

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

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

```
spool console stop
```

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

```
close console
```

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

REORDERING AND PURGING SPOOL FILES

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

```
order printer 2468 1357 class c
```

resequences your closed printer spool files as follows:

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

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

```
purge 1234 class b
```

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

```
purge punch all
```

removes all punch files.

```
purge all
```

removes all files from all devices.

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

DEDICATED DEVICES

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

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

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

```
DEDICATE 495 295
```

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

```
detach 495
```

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

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

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

```
msg operator pls attach 281 to smith as 181
```

The operator issues the command:

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

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

```
TAPE 181 ATTACHED
```

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

```
detach 181
```

You receive the acknowledgment:

```
TAPE 181 DETACHED
```

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

DEDICATED CHANNELS

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

```
msg operator pls attach channel 2 to smith
```

If the channel is available, the operator issues:

```
attach channel 2 to smith
```

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

```
CHANNEL 2 ATTACHED
```

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

```
detach channel 2
```

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

```
CHANNEL 2 DETACHED userid
```

Controlling the Virtual Machine

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

LOADING AN OPERATING SYSTEM

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

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

```
logon userid noipl
```

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

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

```
ipl 190 clear
```

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

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

```
ipl 190 clear stop
```

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

```
begin
```

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

```
ipl dosvs
```

If you do load by name, the options to clear storage or to stop before loading the initial PSW are not available.

Whether you IPL by device address or by name, you can pass up to 64 bytes of data (including embedded and trailing blanks) to your operating system via the IPL command. The data is entered following the keyword PARM:

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

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

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

where xx... denotes no change to previous contents.

If you had entered three blank characters following the word "passed" in the command line, the registers would contain:

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

For further information on the PARM operand, refer to the "Usage Notes" discussion of the IPL command in Section 5.

SIMULATING INTERRUPTS

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

```
external
```

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

```
external A8
```

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

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

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

```
attn
-- or --
request
```

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

ENTERING CP COMMANDS

There are several ways that you can interrupt the running of your virtual machine in order to execute CP commands. You can stop the virtual machine and place it in a virtual console read environment; you can go from a virtual console read environment to the CP environment; and, you can stop a virtual machine and place it directly into the CP environment.

ENTERING CP COMMANDS WHEN THE VIRTUAL MACHINE IS RUNNING

If your terminal mode is set to VM, pressing the attention key (or its equivalent) once stops the virtual machine and places it into the virtual console read environment. You can now execute CP commands by entering one or more command lines as operands of the #CP command; for example:

```
#cp query time
-- or --
#cp query time#query users
```

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

Note: The pound sign (#) in these examples represents the logical line end character currently in effect.

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

```
#cp attn
-- or --
#cp request
```

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

```
#cp query time
```

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

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

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

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

To restart the virtual machine, enter:

```
begin
```

ENTERING CP COMMANDS FROM THE VIRTUAL CONSOLE READ ENVIRONMENT

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

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

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

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

```
#cp begin
```

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

```
query time
```

```
-- or --
```

```
query users
```

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

Entering the command:

```
begin
```

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

ENTERING CP COMMANDS FROM CP CONSOLE FUNCTION MODE

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

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

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

```
begin
```

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

RECONFIGURING THE VIRTUAL MACHINE

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

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

```
define reader 00b
```

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

```
define t3330 as 291 cyl 125
```

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

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

You can enter:

```
define storage as 768k
```

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

Note: When defining virtual devices, you must be careful to specify virtual addresses that do not result in conflict or contention in the virtual control unit interface.

For example, when using the SPOOL, DEDICATE and SPECIAL directory control statements to define virtual devices, specify virtual addresses that do not conflict or contend with the virtual control unit interface. This conflict or contention occurs because devices can require special

I/O interface protocol from control units such as shared and nonshared subchannel operations. Putting devices that require different real control units on the same virtual control unit can result in a hung or busy condition. To avoid this problem, users must define (and separate) device within their own control unit range. For example, if the directory entry specifies:

```
SPOOL 102 3211
SPECIAL 103 3270
```

the control unit 0 on channel 1 controls both a nonshared device (the 3211 printer) and a shared device (the 3270 display unit). Processing of channel programs involving these two devices can result in a hung or busy condition.

Testing and Debugging of Programs

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

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

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

STOPPING EXECUTION OF YOUR VIRTUAL MACHINE

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

```
#cp adstop 3000
```

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

```
ADSTOP AT 3000
```

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

```
begin
```

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

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

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

DISPLAYING VIRTUAL STORAGE

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

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

TERMINAL OUTPUT

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

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

For the first format, enter the DISPLAY command as:

```
display 1026-102c
```

you receive the response:

```
001024 xxxxxxxx xxxxxxxx xxxxxxxx
```

For the second format, enter the command as:

```
display t1026-102c
```

and the response is:

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

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

```
display 1024.12
```

The response displays 20 bytes as follows:

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

BYTE ALIGNMENT ON TERMINAL OUTPUT

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

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

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

If you enter:

```
display k1024-3200
```

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

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

To display all storage keys, enter:

```
display k
```

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

```
display X1 4 A
```

and receive the response:

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

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

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

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

PRINTER OUTPUT

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

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

```
dump 0
```

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

```
dump 1064-10ff
```

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

```
dump 1064.9b
```

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

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

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

```
close printer
```

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

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

When you enter at the terminal

```
vmdump 150-200
```

or

```
vmdump 400:500
```

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

If you enter

```
vmdump 150.50
```

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

ALTERING VIRTUAL STORAGE

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

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

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

```
store 1024 46a2
```

```
-- or --
```

```
store l1024 46a2
```

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

```
store 1024 46 a2
```

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

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

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

```
store s1026 d1d6c5
```

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

```
store s1026 d1d6c
```

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

You can store data into one or multiple consecutive registers.

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

```
store g4 123456
```

loads general register 4 with 00123456.

```
store g4 12 34 56
```

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

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

```
store y2 00123456789
```

loads floating-point register 2 with the value 0012345678900000.

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

```
store status
```

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

```
set ecmode on
```

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

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

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

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

TRACING VIRTUAL MACHINE ACTIVITY

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

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

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

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

VM/SP stops tracing an activity when a virtual-machine-extended-facility assist instruction results in a branch. (Virtual-machine-extended-facility assist instructions are described in the publication IBM System/370 Extended Facility, GA22-7022.) Tracing resumes when these two conditions are met:

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

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

```
trace all
```

```
-- or --
```

```
trace svc program branch
```

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

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

```
trace program branch off
```

and tracing is now confined to SVCs only.

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

```
trace all printer
```

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

```
trace end  
close printer
```

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

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

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/SP is:

- Truncations and Abbreviations of Commands and Operands

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

The following examples show the format specifications as used:

Display INDicate DISConn

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

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

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

Reader

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

READER
RDR

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

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

underscore
braces { }
brackets []
ellipsis ...

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

```

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

```

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

```

A
B
C

```

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

Example

The representation

```

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.

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

Example:

Nesting levels are represented as follows

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

Where the highest level of nesting is the operand that is enclosed in only one pair of brackets and the lowest level of nesting is the operand that is enclosed by the maximum number of brackets. As noted, our rule states: the higher level of nesting must be selected in order to select the next level (lower level) operand.

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

Example

The representation

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

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

Section 5. Format of CP Commands

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

The command formats are presented in the following order:

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

Note: The execution of certain commands may produce system messages. Refer to VM/SP System Messages and Codes for information on system message formats and user action.

*

*

Privilege Class: Any

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

```
| *          | anycomment
```

Responses

None.

#CP

Privilege Class: Any

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

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

where:

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

Usage Notes

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

5. If you issue the `TERMINAL APL ON` command on a keyboard equipped with APL characters, and then issue a `#CP` command during your APL session, CP does not translate the `#CP` command. When APL is ON, CP uses a different set of translation tables and does not recognize the pound sign (#) as a logical line end symbol.

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

Figure 5. System Action to #CP Command

Responses:

If you enter `#CP` without a 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. Only one ADSTOP command can be in effect at a given time. If a second ADSTOP command is issued, the previous ADSTOP setting is canceled. The format of the ADSTOP command is:

```
ADSTOP | { hexloc }
        | { OFF    }
```

where:

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

OFF cancels any previous ADSTOP setting.

Usage Notes

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

ADSTOP

3. Address stop (ADSTOP hexloc) is specified as a virtual machine real address. For OS/VS or DOS/VS, the specified address must be in fixed storage. Movement of the page containing the ADSTOP can cause erroneous processing because the resulting SVC X'B3' is presented to the virtual machine for handling.
4. If the SVC handling portion of the virtual machine assist feature is enabled on your virtual machine, CP turns it off when an ADSTOP is set. When the address stop is removed, CP returns the assist feature SVC handling to its previous status.
5. Issue the SET RUN OFF command before the ADSTOP command.

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.
3. CP ignores the ATTN and REQUEST commands if an I/O operation is in progress at the console or if other interrupts are pending for your virtual machine. This condition may result if you issue the ATTN or REQUEST command by means of DIAGNOSE X'8'.

Responses

None.

BEGIN

BEGIN

Privilege Class: G

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

```
Begin | [hexloc]
```

where:

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

Usage Notes

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

Responses

None. The virtual machine begins execution.

CHANGE

DIST dist

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

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

FLASH name nnn

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

MODIFY name

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

CHARS name

specifies the character arrangement table used when printing a file. The name must be from one to four characters.

FCB name

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

FORM form2

changes the spool form name of the file to form2.

NAME fn [ft]

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

NAME dsname

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

Usage Notes

1. Issue the QUERY command to determine the current attributes of the file.
2. If you want to change an output file, the file must have been closed but not yet selected for printing or punching.
3. You can change an input (READER) file at any time before it is opened; that is, before CP issues the first read for the file.
4. Use the QUERY command to verify the new attributes of the file or files.
5. Your system programmer can define certain spool classes to automatically print certain classification titles (for, example, TOP PRIORITY or CONFIDENTIAL). The title prints on the separator page, and, optionally, at the bottom of each printed page. Ask your system programmer if he has defined certain classes for this function, or request that classes be defined. For example, you may wish class P to print TOP PRIORITY on the separator page, and class X to print CONFIDENTIAL on the separator page and also on the bottom of each page of output.
6. In order to print on the 3800, the name you specify in the FLASH option must match the FLASH name the VM/SP operator specifies in the START command, unless the FLASH number nnn is zero. If you specify a zero for the FLASH number or you omit the FLASH option, a file still prints on the 3800 as long as the spool file class matches the print class designated by the installation for the 3800. If you fail to meet these requirements, CP submits the spool file to another printer and the special 3800 print options are ignored.
7. When a spool file containing special 3800 options is spooled to a printer other than the 3800, the options are ignored and printing occurs normally.
8. You can nullify any previous or existing settings of the CHARS, MODIFY, or FCB options by specifying NULL in the name field of these operands.
9. An FCB name of 6, 8, or 12 avoids an FCB forms mismatch. These special FCB specifications print the entire spool file at 6, 8, or 12 lines per inch regardless of the paper size. VM/SP calculates the proper FCB form to load.
10. When the operator activates the 3800 printer, the operator specifies the character arrangement table and the FCB value that prints the separator page for every output file. If you omit the CHARS option on a file spooled to the 3800, the printing defaults to the character arrangement table previously defined by the operator. This is also the case for the FCB option.
11. VM/SP issues a message to you whenever a file spooled to the 3800 printer contains invalid or undefined option names. If the operator has not automatically purged the file, you can correct the situation using the CHANGE command or spool the file to another printer.

CHANGE

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 INSG OFF command line.

CLOSE

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

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

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

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

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

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

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

Usage Notes

1. If the file is an input reader file, the file being processed is purged unless SPOOL READER HOLD was previously specified (see the SPOOL command). The effect of HOLD or NOHOLD for a particular file established by the SPOOL command can be overridden by specifying NOHOLD or HOLD, respectively, in the CLOSE command.
2. If an input spool file is read but the read is not completed (that is, the virtual machine does not get a last-card indication), you must issue a CLOSE READER command in order to be able to read that file again (or to read any other file). Unless you specify HOLD when you close a reader file, the file is purged.

3. If the file is an output file on a printer, punch, or console, the file is either queued for output on a real unit record device, or, if the virtual output device is transferred (by use of the "SPOOL vaddr TO userid" command), the file is queued for input to the receiving user.
4. You can specify a filename and filetype and an optional distribution code to aid in later identification of the file and its contents.
5. Figure 6 shows what happens to an input file in your virtual reader when the CLOSE command is issued with the HOLD or NOHOLD operand in effect. The CLOSE command can result in an input file being purged or saved.

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

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

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

6. Figure 7 shows what happens to a file in your virtual output device when the CLOSE command is issued with the HOLD, NOHOLD, CONT, and NOCONT operands in effect.

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

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

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

CLOSE

Responses

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

Privilege Class: G

Use the COUPLE command to connect your virtual (nondedicated) 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.

COUPLE

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

CTCA vaddr1 DROP FROM userid vaddr2

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

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

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  
        | 3203  
        | 3211  
        | 3262  
        | 3289E  
        | 3505  
        | 3525  
  
        | 3800 [As] vaddr [Size ww ll] [4Wcgm] [CFS] [NODatck]  
        | [2Wcgm] [BTS] [Datck]  
  
        | CHANnels [As] {SEL}  
        | {BMX}  
  
        | Line [As] vaddr [IBM[1]]  
        | [TELE[2]]  
  
        | GRAF cuu [ 3033 ]  
        | [ 3036 ]  
        | [ 3138 ]  
        | [ 3148 ]  
        | [ 3158 ]  
        | [ 3270 ]  
  
        | vaddr1 [As] vaddr2  
  
        | TFB-512  
        | T3310 [As] vaddr [BLK] [nnnnnn]  
        | T3370  
  
        | T2305  
        | T2314  
        | T2319 [As] vaddr [CYL] [nnn]  
        | T3330  
        | T3340  
        | T3350  
        | T3380  
  
        | STORage [As] {nnnnnK}  
        | {nnM}
```

where:

READER [AS] vaddr
RDR
adds a spooling 2540 card reader with the address specified by vaddr to the virtual machine configuration.

PRINTER [AS] vaddr
PRT
adds a spooling 1403 printer with the address specified by vaddr to the virtual machine configuration.

PUNCH [AS] vaddr
PCH
adds a spooling 2540 card punch with the address specified by vaddr to the virtual machine configuration.

CONSOLE [AS] vaddr
adds a virtual system console to the virtual machine at the address specified by vaddr.

CTCA [AS] vaddr
adds a virtual channel-to-channel adapter with the address specified by vaddr to the virtual machine configuration. 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.

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

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

DEFINE

3262 [AS] vaddr

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

3289E [AS] vaddr

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

3505 [AS] vaddr

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

3525 [AS] vaddr

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

3800 [As] vaddr [Size ww ll] [2WCGM] [BTS] [DATck] [4WCGM] [CFS] [NODatck]

adds a spooling 3800 printer to the virtual machine configuration; vaddr specifies the address to the virtual machine configuration. The SIZE keyword specifies the size of the forms that CP loads into the virtual 3800. The ww specifies the width of the form as a hexadecimal code (see Usage Note 6. for the available form width codes.) The ll specifies the length of the paper in half inches expressed in decimal notation. For example, if you want 8 1/2" by 11" paper loaded into your virtual 3800, specify:

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

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

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

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

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

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

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 [3033]
 [3036]
 [3138]
 [3148]
 [3158]
 [3270]
 [

defines a temporary 3033, 3036, 3270, 3138, 3148, or 3158 (the console used on the IBM System/370 Model 138, 148, or 158) for the virtual machine. The cuu is the hexadecimal virtual address for the device. After you define the 3033, 3036, 3270, 3138, 3148, or 3158, you must issue the CP DIAL command from the device in order to use it. The device must be supported by the virtual machine's operating system.

DEFINE

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

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

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

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

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

Example:

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

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

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

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

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 for count-key-data devices. Specify T3350 if a 3350 is used in native mode; specify T3330 if a 3350 is used in 3330 compatibility mode. Specify T3340 if a 3344 is used.

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

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

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

STORAGE [AS] { nnnnnK }
 { nnM }

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

Usage Notes

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

DEFINE

2. If either storage or channel operating mode is altered, the virtual machine is immediately reset and the IPL procedure must be performed again.
3. If storage is redefined, any pages that have been locked via the LOCK command become unlocked.
4. If your virtual machine is operating in the basic control mode (ECMODE OFF), virtual addresses specified in vaddr, vaddr1, and vaddr2 can be no higher than 5FF (channels 0 through 5). If it is operating in extended control mode (ECMODE ON), all addresses through FFF are valid.
5. Use caution in defining the hexadecimal addresses of virtual devices (cu) in DEFINE statements in order to avoid a usage conflict caused by control unit I/O interface protocol. The following is an example of a virtual machine's DEFINE statements that can cause operational conflict:

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

The virtual addresses of both the 3211 printer and the 3270 terminal indicate the use of the same channel (1) and control unit (0). By definition, the devices are virtual and therefore share one virtual control unit in CP. A real 3211 printer operates on a nonshared subchannel, and the real 3270 console is designed for shared subchannel operations. Both of these real devices are mapped to the same virtual control unit. Thus, subsequent processing of a channel program involving these devices can result in a hung or busy condition (caused by a conflict in real-to-virtual I/O processing). Therefore, when defining devices, make sure the devices are defined (and separated) within their own control unit range and not shared with other devices.

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

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

Figure 8. Available Form Width Codes for 3800 Spooling

Responses

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

type vaddr DEFINED
where the possible values for type are:

| <u>Type</u> | <u>Meaning</u> |
|-------------|--|
| DASD | Direct access storage device/3330V (virtual volumes) |
| TAPE | Magnetic tape |
| LINE | Communication line |
| RDR | Card reader |
| PRT | Line printer |
| PUN | Card punch |
| GRAF | Graphics device |
| CONS | Console |
| CTCA | Channel-to-channel adapter |
| MSC | Mass Storage Control |

CHANNELS = {SEL}
 {BMX}

is the channel mode of operation for the virtual machine. This response applies to all channels except channel 0 (always a byte multiplexer channel) and any channel that has a virtual or real channel-to-channel adapter (always a selector channel).

STORAGE = {nnnnnK}
 {nnM}

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

DETACHPrivilege Class: G

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

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

where:

vaddr [vaddr...]

vaddr-vaddr

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

Usage Notes

1. You can detach a previously attached device even if the device is currently in use. You can also detach devices that were attached because of VM/SP directory entries or during CP system initialization.
2. When you detach a virtual device, it becomes inaccessible to your virtual machine. If the device was previously attached to your virtual machine by an ATTACH command, it is released and becomes available for attachment to your virtual machine, another user, or the CP system. Tape devices are automatically rewound and unloaded when detached.
3. If you detach a device that was previously attached to your virtual machine by the operator, a message is sent to the operator informing him that the device is free.
4. If the device being detached is a dedicated 2305, the virtual address you specify must be a base address (xx0 or xx8).
5. If a detach operation is taking place for a 2305 device and a range is being used, processing for all 2305 exposures is performed when the base address is encountered. It is possible that the range operand may not include all the 2305 exposures. This is not an error situation; all exposures are detached. When multiple real device addresses are specified on the command line, only the base address for the 2305 should be used. Any device address other than the base exposure is considered an error condition.

6. Command processing does not terminate if an error (other than a syntax error) is encountered when processing a range of addresses or multiple addresses. Instead, an error message is issued for each device (address) that cannot be detached. In addition, an informational message identifies those devices that were successfully detached. Informational messages may appear several times in one command execution when a range of addresses or multiple addresses are being processed. Multiple informational messages appear only when range is being processed.
7. It is possible for a class B user to DETACH real devices from another user or other users.

Responses

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

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

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

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

```
{ type vaddr } DETACHED
 { vaddr... }
 { vaddr-vaddr }
```

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

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

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

DETACH

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

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

| <u>Type</u> | <u>Meaning</u> |
|-------------|--|
| DASD | Direct access storage device/3330V (virtual) volumes |
| TAPE | Magnetic tape |
| LINE | Communication line |
| RDR | Card reader |
| PRT | Line printer |
| PUN | Card punch |
| GRAF | Graphics device |
| CONS | Console |
| CTCA | Channel-to-channel adapter |
| DEV | Any other device |
| MSC | Mass Storage Control unit |

DETACH CHANNEL

Privilege Class: G

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

```
| DETach | CHANnel c |
```

where:

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

Usage Note

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

Response

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

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/SP and the multiple-access virtual machine. See the VM/SP Terminal User's Guide for details on running and gaining access to multiple-access machines.
2. The DIAL command is not supported for the 3066 system console, for the 3210/3215 system console, or for terminals that are using NCP lines in a 3704/3705 control unit, but is supported for 3138, 3148, and 3158 display consoles.
3. Once the connection is made, your terminal operates entirely under the control of that virtual machine. The DIAL command matches your terminal to an equivalent type defined in the multiple-access virtual machine. If no matching terminal type exists, the connection cannot be made and an error message is issued.
4. The DIAL command, though not supported for remote 3270 terminals, is supported for locally attached 3270 display devices. If the DIAL command is issued from a local 3270 terminal, the virtual system user must issue the CP command RESET to drop the dialed connection.
5. The DIAL command is not supported for SNA terminals logically connected to the host VM/SP system using the VTAM Communications Network Application (VM/VCNA) program product.

Responses

DIALED TO userid vaddr
is the message sent to the user indicating that a logical connection has been made.

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

is the response to the primary system operator. It indicates a successful connection to the virtual machine (userid) and the total number of VM/SP lines (nnn) currently connected to other virtual machines. DEV resid indicates the resource identification of a 3704/3705 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/SP system while the virtual machine continues operation. If you have a secondary userid {secuser} specified in your directory entry, you can now use the console of that logged on secondary user as your virtual console after you disconnect.

The format of the DISCONN command is:

```
DISConn | [Hold]
```

where:

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

Usage Notes

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

Responses

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

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

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

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

DISPLAYPrivilege Class: G

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

- Virtual storage locations (second-level virtual storage only; see "Usage Notes")
- Storage keys
- General registers
- Floating-point registers
- Control registers
- Program status word (PSW)
- Channel address word (CAW)
- Channel status word (CSW)

Note: Use the NETWORK DISPLAY command to display the content of 3704/3705 storage.

The format of the DISPLAY command is:

| | |
|---------|--|
| Display | <pre> [hexloc1] [{-} [hexloc2]] [Khexloc1] [{ : } [END]] [Lhexloc1] [[]] [Thexloc1] [{ . } [bytecount]] [[] [[] [END]]] Greg1 [{-} [reg2]] Yreg1 [{ : } [END]] Xreg1 [[]] [{ . } [regcount]] [[] [END]] Psw CAW CSW </pre> |
|---------|--|

where:

hexloc1
Lhexloc1
Thexloc1
Khexloc1

is the first, or only, hexadecimal storage location that is to be displayed at the terminal. If L or no letter prefix is specified, the storage contents are displayed in hexadecimal. If T is specified, the storage contents are displayed in hexadecimal, with EBCDIC translation. If K is specified, the storage keys are displayed in hexadecimal.

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

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

{-}hexloc2
{:}END

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

{.}bytecount
END

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

DISPLAY

{-}reg2
{:}END

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

{.}regcount
END

is a decimal number from 1 to 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. First level storage is real main storage. Only second-level storage (storage that is real to the virtual machine) can be displayed. Operating systems running in a virtual machine such as DOS/VS and OS/VS have virtual storage of their own. This third-level virtual storage cannot be displayed directly. The user or the virtual operating system is responsible for converting any third-level storage locations to second-level storage locations before issuing the command.
2. If a command line with an invalid operand is entered, the DISPLAY command terminates when it encounters the invalid operand; however, any previous valid operands are processed before termination occurs. Multiple storage locations, registers, and control words can be displayed using a single command line.

3. When multiple operands are entered on a line for location or register displays, the default display type is the same as the previous explicit display type. The explicit specification of a display type defines the default for subsequent operands for the current display function. Blanks are used to separate operands or sets of operands if more than one operand is entered on the same command line. If a dash or a colon is omitted between the hexloc1 and hexloc2 operands, and the two operands are separated by one or more blanks, CP only displays the storage contents at those two specific addresses. Blanks must not be used to the right or left of the range or length delimiters (: or - or .), unless the user wants the default value of the missing operand defined by the blank. For example:

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

displays the following, respectively:

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

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

Responses

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

DISPLAY

Displaying Storage Locations

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

This is the response you receive when you display storage locations; xxxxxx is the hexadecimal storage location of word1. Word1 is displayed (word-aligned) for a single location specification. Up to four words are displayed on a line, followed, optionally, by an EBCDIC translation of those four words. Periods represent nonprintable characters. Multiple lines are used (if required) for a range of locations. If translation to EBCDIC is requested (Theoloc), 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 .xxxxxxxxxxxxxxxxxxxx E xx
```

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

Displaying Control Registers

ECR n = ctlreg1 ctlreg2 ctlreg3 ctlreg4

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

Displaying the PSW

PSW = xxxxxxxx xxxxxxxx

The contents of the PSW are displayed in hexadecimal.

Displaying the CAW

CAW = xxxxxxxx

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

Displaying the CSW

CSW = xxxxxxxx xxxxxxxx

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

DUMP

Privilege Class: G

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

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

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

The format of the DUMP command is:

```

DUMP | [Lhexloc1] | {(-) [hexloc2] | [*dumpid]
      | [Thexloc1] | { : [END] |
      | [hexloc1] | [.] [bytecount] |
      | | [END] |
  
```

where:

Lhexloc1
Thexloc1
hexloc1

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

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

```

{-} hexloc2
{:} END
  
```

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

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

{.}bytecount
END

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

*dumpid

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

Usage Notes

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

DUMP 0

5. The output format for the virtual storage locations is eight words per line with the EBCDIC translation on the right. Each fullword consists of eight hexadecimal characters. All the rest of the information (PSW, general and floating-point registers, and storage keys) is printed in hexadecimal. If you have the ECMODE option in your VM/SP directory entry, the control registers are also printed.

DUMP

To print the dump on the real printer, a CLOSE command must be issued for the spooled virtual printer.

6. Normally, you should define beginning and ending dump locations in the following manner:

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

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

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

The following produces three full dumps:

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

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

Responses

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

DUMPING LOC hexloc

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

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

COMMAND COMPLETE

is the response indicating normal completion of the dump function.

ECHO

Privilege Class: G

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

The format of the ECHO command is:

```

ECho | [nn]
      | [1]
      | ]

```

where:

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

Usage Notes

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

Responses

ECHO ENTERED; TO TERMINATE TEST, TYPE END
 is displayed after the ECHO command is invoked to indicate that the echo environment has been entered.

ENTER LINE

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

EXTERNAL

EXTERNAL

Privilege Class: G

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

The format of the EXTERNAL command is:

```
EXTERNAL | [code] |  
         | [ 40 ] |
```

where:

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

Usage Note

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

Responses

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

INDICATE

Privilege Class: G

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

```

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

where:

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

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

Usage Notes

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

Response for INDICATE LOAD

```

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

where:

CPU-nnn%

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

APU-nnn%

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

INDICATE

PROC xx-nn%

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

PROC yy-nn%

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

Q1-nn Q2-nn

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

STORAGE-nnn%

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

EXPAN-nnn

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

PAGING-nnn/SEC

indicates the average number of page reads and writes performed per second.

STEAL-nnn%

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

LOAD-nnn%

is an artificial value. It attempts to measure the elapsed time as a percentage of the system time devoted to paging that results from real storage contention. This calculation includes the estimated processor time involved in paging, the amount of time spent in page waits, and the percentage of pages stolen (STEAL).

Response for INDICATE USER

PAGES: RES-nnnn WS-nnnn READS=nnnnnn WRITES=nnnnnn MH-nnnn FH-nnnn
 VTIME=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.

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

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

VTIME=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 processor time (virtual plus overhead) that you have used since you logged on or since the last ACNT command was issued for your virtual machine.

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

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

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.

IPLPrivilege Class: G

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

```

Ipl {
  vaddr [cylno] [CLEAR] [STOP] [ATTN] [PARAM {p1 p2...}]
  [nnnnnn] [NOCLEAR]
  systemname
}

```

where:

```

vaddr [cylno] [CLEAR] [STOP]
      [nnnnnn] [NOCLEAR]

```

simulates the IPL function when loading by device address.

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

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

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

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

NOCLEAR

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

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

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

systemname

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

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

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

PARM {p1 p2... }

passes up to 64 bytes of data (excluding the first blank character after the keyword, PARM, but including all other embedded blanks) to your virtual machine's general registers (4 bytes per register), starting with the high order byte of general register 0. Since the registers are not cleared before use, the PARM value should be some multiple of 4 bytes to avoid having extraneous characters present in the low order positions of the register. For example, you could key in

PARM CMSØ

where Ø represents a blank to ensure that the low order positions of register 0 contain a hexadecimal 40. 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 system console. The specified virtual address is accessed and the required input/output operations are performed to retrieve the IPL data.
2. Optionally, the IPL procedure can be stopped just before loading the virtual PSW except when loading a named system. Also, parameters can be passed to the virtual machine's general registers. When the simulated load function is complete, CP initiates execution of the virtual machine by loading the IPL PSW which was stored during the simulation process.
3. Care must be used when passing parameters to a named system (systemname). Named systems expect certain registers to be initialized when they are given control. Indiscriminate use of the PARM operand could overlay a previously initialized register causing unpredictable results.

4. The IPL simulator program occupies one page (4096 bytes) of storage in your virtual machine. Therefore, if the system being loaded utilizes data remaining in storage by the previous system, card should be taken as to where that data is located. The starting address of the overload virtual storage is:

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

For example:

Storage size=200K then the IPL Simulator is loaded at 100K.

5. To load a VMSAVE system, a number of conditions must be met:
- The user must be specified in module DMKSNT as either the owner (USERID=) or the receiver (RCVRID=).
 - The area must contain a valid saved system.
 - The system must have been saved under the same system name expressed in the IPL command.
 - The area cannot be the target area for other users currently enabled for VMSAVE.

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

6. Any logical lines following the IPL command are ignored since the execution of the IPL command transfers control from the CP environment to the virtual machine environment.
7. Some CMS routines and programs are stored on disks and loaded into storage as needed. Since this storage area is not contiguous with your virtual storage, the saved segments that are loaded in this area are called discontinuous segments (DCSS). However, the only discontinuous saved segment a CMS user can replace is the CMSSEG segment. The user can indicate an alternate segment to be loaded by issuing an IPL command with this format:

```
IPL {cuu          } PARM SEG=segmentname
    {systemname }
```

where:

SEG=segmentname

indicates the name of the saved segment to be loaded whenever the CMS Editor, EXEC processor, or OS simulation routines are needed. Eight characters must be entered for segmentname; either assign an 8-character segment name when you code the NAMESYS macro for your installation, or be sure that the operator enters trailing blanks if segmentname is less than eight characters long.

The CMS batch facility loads whatever segment is specified on the first IPL command issued for the batch virtual machine. Thus, if the first IPL command for a CMS batch facility machine is:

```
IPL CMS PARM SEG=CMSSEG02
```

all subsequent IPL commands issued by the CMS batch facility specify the same segment name (CMSSEG02).

8. The BATCH option in the PARM operand indicates that the CMS initial program load is being performed in a batch instead of an interactive virtual machine. This option causes the CMS nucleus to issue the CMSBATCH command to load the batch processor. Specifying the BATCH option eliminates the need for operator intervention between jobs. Consult VM/SP CMS Command and Macro Reference for information on the CMSBATCH facility.
9. CMS users can use the PARM operand to pass the AUTOOCR (automatic carriage return) option to the CMS initialization routine. AUTOOCR simulates the pressing of the ENTER key as input to the virtual machine. This results in automatic execution of the user's PROFILE EXEC, if it exists on virtual minidisk 191.

Note: For a disconnected CMS service virtual machine, this mechanism can prove to be very useful. The name of a CMS restart routine may be specified in the CMS nucleus. In the event of an ABEND, the specified restart routine receives control instead of the service virtual machine entering a disabled wait state. If this routine issues an IPL CMS PARM AUTOOCR command, the PROFILE EXEC executes initial setup procedures that allow the service virtual machine to reestablish itself.
10. If the user issues an 'IPL vaddr ATTN' command to a virtual machine other than OS/VS1 with the FASTNIP feature, results are unpredictable. The results depend on how the object operating system handles ATTN interrupts during IPL.

Responses

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

LINK

LINK

Privilege Class: G

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

```
LINK | [To] userid vaddr1 [As] vaddr2 [mode] [[PASS=] password1]
```

¹If your installation uses the password suppression facility, the password for a DASD unit cannot be entered on the LINK command line. See the full description of the password operand for the steps needed to link to DASD with the password suppression facility in effect.

where:

[TO] userid

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

vaddr1

is the virtual device address (cuu) in the VM/SP directory for that user1.

[AS] vaddr2

is the virtual address (cuu) that is to be assigned to the device for your virtual machine. If the keyword AS is omitted, vaddr may not be "AS" or "A". If your virtual machine has the ECMODE option, any address up to X'FFF' is valid; otherwise, any address up to X'5FF' is valid.

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 (R/O) 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 (R/O) 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.

Mode Meaning

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 (R or RR) link to a device that is being updated by a user who has the device in write/read status.

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

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/SP 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/SP directory for the user (userid) specified. The password should be specified only when the LINK is executed by a virtual machine (for example, from CMS), since printing of the password is not suppressed when included with the LINK command. The access mode password cannot be the same as any of the access modes (R, RR, W, WR, M, MR, or MW) if the default mode is to be used.

Notes:

1. The access mode password should not be confused with a user password.
2. If your installation is using the password suppression facility, an INVALID FORMAT message is issued when a user attempts to enter the password for a DASD unit on the LINK command line. The user must wait for the ENTER PASSWORD response before typing in the password. This facility improves system security because the password is automatically masked.

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. If a LINK command is issued from a CMS batch job, a password must be provided, even if the device password is ALL.
2. When linking to a 2305 device, you gain access only to the virtual address specified and not to all eight addresses associated with the device.
3. 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/SP directory for the disk being linked. The third row indicates whether the disk is already being

LINK

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.

4. Virtual reserve/release processing cannot be requested by appending a V to the mode operand of the LINK command line. Virtual reserve/release processing can only be requested by means of the MDISK statement of the directory program.

| | | | |
|------------------------------|-----------|-------------|-------------|
| 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 |

where: N=no or none; R=read; W=write; M=multiple; Y=yes

Figure 9. Password Requirements on the LINK command.

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 the LOGON/AUTOLOG/LINK journaling facility is activated, unsuccessful link attempts, due to invalid passwords, are recorded. When a predefined system threshold value for attempts is exceeded, an installation userid receives a message. There is another system threshold value that disables the LINK command when a user continues to enter trial passwords to obtain a link to a minidisk. The default is ten incorrect passwords before the command is disallowed for the remainder of the terminal session.

DASD vaddr2 LINKED R/O

This response indicates that a read-only link to the given disk is established for a LINK request with a mode of R or RR, and that no other users are linked to the same disk in read/write mode.

DASD vaddr2 LINKED R/W

This response indicates that a read/write link to the given disk is established for a LINK request with a mode of W, WR, M, MR, or MW, and that no other users are linked to the same disk.

DASD vaddr2 LINKED R/O; R/W BY {nnn USERS} [; R/O BY {nnn USERS}]
 {userid} [{userid}]

This response indicates that a read-only link to the given disk is established for a LINK request with a mode of RR, but warns that the disk is in read/write use by some users and possibly in read use by some users. If only one user has access, the number of users (nnn USERS) is replaced by userid.

DASD vaddr2 LINKED R/W; R/O BY {nnn USERS}
 {userid}

This response indicates that a read/write link to the given disk is established for a LINK request with a mode of M, MR, or MW, and informs you that the disk is also in read-only use by userid or by nnn users. (No other users have a read/write link to the disk.)

DASD vaddr2 LINKED R/W; R/W BY {nnn USERS} [; R/O BY {nnn USERS}]
 {userid} [{userid}]

This response indicates that a read/write link to the given disk is established for a LINK request with a mode of MW, but warns you that the disk is also in read/write use by some users and possibly in read use by some users. If only one user has access, the number of users (nnn USERS) is replaced by userid.

LOADVFCB

LOADVFCB

Privilege Class: G

Use the LOADVFCB command to specify the forms control buffer image for a virtual spooled 3203, 3211, 3262, or 3289E printer. The format of the LOADVFCB command is:

```
LOADVFCB | vaddr   Fcb  name [Index [nn]]
```

where:

vaddr is the virtual device address (cuu) of the virtual spooled 3203, 3211, 3262, or 3289E printer.

FCB is a required reserved keyword meaning Forms Control Buffer.

name is a system-defined name for the 3203, 3211, 3262, or 3289E FCB image which is to be the controlling virtual FCB image.

INDEX [nn] for the 3211 only, is the number of the print position that is the first print position. The value, nn, must be a number from 1 through 31; a leading zero need not be specified. If the keyword INDEX is specified without a value, the index defaults to the value specified in the FCB macro. See the VM/SP System Programmer's Guide for a discussion of the FCB macro and forms control images.

The 3203, 3262, and 3289E do not support the index feature and CP rejects any attempt to issue the INDEX operand.

Usage Notes

- 1. The LOADVFCB command can be used with installations that do not have a 3203, 3211, 3262, or 3289E printer. The virtual machine's VM/SP directory entry must indicate a 3203, 3211, 3262, or 3289E, even though the program and operating system have a 1403 printer defined. Then the LOADVFCB command can be used to specify a virtual FCB image for 1403 printers so that programs that use printer overflow sensing may be spooled to disk.
2. VM/SP provides one FCB image (FCB1) with the following format:

Table with 3 rows: Line spacing = 6 lines/inch, Length of page = 66 lines, and Channel skip grid with 13 columns and 2 rows of data.

Your system programmer may provide you with additional FCB images.

3. The user must load a virtual 3203, 3211, 3262, or 3289E printer with an FCB that is compatible with an FCB used in its read counterpart. Failure to do so may produce unpredictable results.
4. CP stores the virtual FCB image in all spool files created on the virtual spooled printer. If a real 3203, 3211, 3262, or 3289E printer prints the spool file, the virtual FCB image is sent to the real printer. The spool file is then printed under the control of the FCB, with the channel skip commands skipping to the proper line on the page.

If the spool file is printed on a real 1403 printer, the virtual FCB image is ignored and channel skipping is controlled by the carriage tape.

5. If the virtual machine operating system issues a LOAD FCB channel command word to the virtual spooled printer, the LOAD FCB CCW image replaces the virtual FCB image provided on the LOADVFCB command line. Note that the CMS PRINT command issues a LOAD FCB CCW and thereby replaces the virtual FCB image.
6. The following table shows the default virtual FCB image for a spooled 3203, 3211, 3262, or 3289E printer:

| | | | | | | | | | | | | |
|-----------------------------|---|---|---|---|---|---|---|---|---|----|----|----|
| Line spacing = 6 lines/inch | | | | | | | | | | | | |
| Length of page = 66 lines | | | | | | | | | | | | |
| Page line | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 60 |
| Channel skip | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |

Responses

None.

LOGOFFPrivilege Class: Any

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

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

where:

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

Usage Notes

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

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

- Typewriter Terminal and Remote 3270 Display Terminal -- Logoff takes place after a 15-minute interval has elapsed. This occurs if no attempt is made to turn terminal power on and re-establish communications with the still logged-on virtual machine during this 15-minute period.
- Local 3270 Display Terminal -- Logoff only takes place 15 minutes after VM/SP discovers that the terminal has been turned off (that is, VM/SP attempts to send a message to the terminal, but gets back an error code indicating that the terminal is turned off). Because many hours may pass before VM/SP 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= mmm:ss.hs
the virtual CPU time used in the current terminal session in
minutes:seconds.hundredths of seconds.

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

These times are either the elapsed time for the entire terminal
session or the elapsed time since the ACNT command was entered for
this user.

LOGOFF AT hh:mm:ss zone weekday mm/dd/yy
is the response for a logoff.

{ 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

Privilege Class: Any

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

| | | | | |
|-------|--|--------|------------|---------|
| Logon | | userid | [password] | [Noipl] |
| Logon | | | | |

where:

userid is the identifier assigned to you in the VM/SP system.

password is your password. Specify this field if no protection (that is, masking characters) is desired.

Note: If your installation is using the password suppression facility, an INVALID FORMAT message is issued when a user attempts to enter the system LOGON password on the command line. The user must wait for the ENTER PASSWORD response before typing in the password. This facility improves system security because the password is automatically masked.

NOIPL specifies that the IPL device or name in the VM/SP directory should not be used for an automatic IPL.

Usage Notes

1. Upon successful logon, VM/SP creates a virtual machine configuration from information in that user's directory entry. The LOGON command name may not be entered using any line-editing symbols, but the operands may use these symbols. See the VM/SP Terminal 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.
3. If you previously issued the DISCONN command to run your virtual machine without a console and then issue the LOGON command to reconnect your console, certain SET and TERMINAL command operands are forced off even if they were on prior to issuing the DISCONN command. For example, after a reconnect, CP resets the line delete character specified in the directory to the system default value. Issue the QUERY SET and QUERY TERMINAL command lines to check which functions are currently active.
4. During the virtual machine logon process on a VM/SP system generated for attached processing operation but operating in uniprocessor mode, a message is sent to the user with attached processor AFFINITY set (via the user directory's OPTION statement)

that the AFFINITY setting of his virtual machine is nullified by the system's uniprocessor mode of operation.

5. If the LOGON/AUTOLOG/LINK journaling facility is activated, accounting records are produced for unsuccessful logon attempts with incorrect passwords. If system defined values are exceeded attempting to logon with trial passwords, a message is issued to an installation userid and any further LOGON attempts by the user are rejected.
6. CP automatically masks your password on typewriter terminals to insure installation security. For installations migrating from a VM/370 operating system to VM/SP, and for compatibility purposes, CP still accepts the MASK operand on the LOGON command as valid. However, CP automatically masks out your password should you forget to issue a LOGON command without the MASK operand.

Responses

ENTER PASSWORD:

indicates that the userid has been accepted. You should now type in the password.

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

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

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

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

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

-- or --

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

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

```
(luname
  GRAF raddr ) LOGON AS userid    USERS = nnn
  LINE raddr )
  (DEV resid )
```

-- or --

```
(luname
  GRAF raddr ) RECONNECT userid  USERS = nnn
  LINE raddr )
  (DEV resid )
```

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

MESSAGE

Privilege Class: Any

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

| | | | |
|---------|--|-------------|---------|
| Message | | {userid } | msgtext |
| MSG | | {* } | |
| | | {Operator } | |

where:

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

* specifies that you are sending a message to yourself.

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

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

Usage Notes

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

Responses

hh:mm:ss

MSG FROM OPERATOR: msgtext

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

hh:mm:ss

MSG FROM {LOGONxxx}: msgtext

userid

{LOGNxxxx}

is the format of the message sent to another user or to the system operator, where userid is that of the sender. If the user sending the message is not logged on to VM/SP, LOGON or LOGN and the line number are displayed instead of userid.

hh:mm:ss

is the time in hours:minutes:seconds when the message was sent to the user.

NOTREADY

NOTREADY

Privilege Class: G

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

```
| NOTReady | vaddr
```

where:

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

Usage Notes

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

Response

None.

ORDER

Privilege Class: G

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

| | | | | | |
|-------|--|--------------------------------|--|--|----------------|
| ORDER | | { Reader Printer PUnch } | | { Class c1 Class c2... FORM form1 FORM form2... spoolid1 spoolid2... } | } ¹ |
|-------|--|--------------------------------|--|--|----------------|

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

where:

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

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

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

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

FORM form1 FORM form2... are the forms you want rearranged and the sequence in which you want CP to process them. The form1 and form2 specifications are one-to-eight character form names defined during system generation.

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

Usage Notes

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

order printer class A 1963 class C form CATALOG4

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

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 | | FORM form1 FORM form2... | | |
| | | PUnch | | spoolid1 spoolid2... | | |
| | | ALL | | ALL | | |
| | } | | [| |] | |

¹A combination of CLASS, FORM 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.

FORM form1 FORM form2...
specifies that CP should purge all files with the specified form names defined during system generation. The form1 and form2 options are one-to-eight character form numbers.

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

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

PURGE

Usage Notes

1. You can purge any closed spool file you own, regardless of its status, as long as CP has not selected it for processing.
2. You can purge files using a combination of CLASS, FORM, and spoolid specifications. For example:

```
purge printer class A 1932 class D 619 form INVOICE
```

specifies that CP should purge all class A and class D printer files, printer files with spoolids of 1932 and 619, and all printer files with form name INVOICE.

3. You cannot purge any file you spool to the reader of another user it is unless you first reclaim it using the TRANSFER command.

Response

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

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

QUERY

Privilege Class: G and all classes except class Any

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

- The time you have used during a terminal session.
- The number of closed input and output spool files associated with your virtual machine. For information on how to alter spool file attributes, see the CHANGE command.
- The current settings of the color and/or extended highlight values in effect for your virtual machine console.
- The current settings of the SET command functions.
- The current settings of the TERMINAL command functions.
- The status of all the devices on your virtual machine.
- The channel operating mode of your virtual machine, whether block-multiplexer or selector.
- A listing of all users who are linked to a given virtual address, together with their device addresses and access modes.
- Display of the secondary user (secuser) that is specified in the CONSOLE directory statement.
- Identification and attributes associated with your virtual printer, punch, and reader spool files.
- The identification of your virtual processor.
- The mode of processor operation of your VM/SP installation: uniprocessor mode (UP), attached processor mode (AP), or multiprocessor mode (MP).

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

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

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

QUERY

QUERY COMMAND FOR CLASS G USERS

The format of the Class G QUERY command is:

| | |
|---|--|
| Query | <pre> Time Set TERMINal Files [Class c] [FORM form] { NOHold }¹ { Hold } { USERhold } { SYShold } [Virtual] { CHANnels } GRAF } CONsole } DASd } TAPes } LINES } UR } STORage } ALL } vaddr[-vaddr] } Links vaddr CPUid SECuser Reader { Printer { [Class c] [FORM form] { NOHold } Punch { spoolid { Hold } { USERhold } { SYShold } } } [ALL] [TBL]¹ PF[nn] SCREEn PROCEss VMSAVE S370E SPMODE </pre> |
| <p>¹The options may be specified in any order.</p> | |

where:

- TIME** displays the current time, time zone, weekday, date, connect and processor time for the current terminal session.
- SET** displays the status of the SET command functions.
- TERMINAL** displays the current options in effect for your virtual console environment.
- CPUID** displays the 16-digit processor identification that is in use by the virtual machine. Values within the CPUID can be set by the directory OPTION statement or modified by the SET CPUID.

FILES displays the number of spooled input and output files for your virtual machine. Files currently being processed are not included in the totals. The CLASS, FORM, and HOLD, NOHOLD, SYSHOLD, or USERHOLD operands restrict counting to certain spool files.

VIRTUAL displays the status of all virtual devices.

CHANNELS displays the channel mode of operation for the virtual machine.

GRAF displays the status of all 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.

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

LINKS vaddr displays the userid, device address, and access mode at the terminal for all users linked to the specified virtual address (vaddr). A range of virtual addresses is not supported by the LINKS operand.

SECUSER displays the userid of the secondary user that is specified in the CONSOLE directory statement.

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
- Form name

One line of information is displayed for each spool file. spool file specified by spoolid).

QUERY

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

- Date and time the file was created
- Filename and filetype of file (if any); if your file was assigned a dsname and you later issue **QUERY**, only the first 18 characters of the 24-character field are displayed.
- Distribution code of the file

Note: The form name is not displayed when the **ALL** option is used.

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

TBL displays additional information about spool files pertaining to the 3800 printer. This option must follow the **READER**, **PRINTER**, or **PUNCH** operand. In addition to the information normally displayed, the following is also displayed:

- Character arrangement table
- Vertical format (FCB)
- Copy modification module
- Forms overlay frame
- Number of copies to be printed with forms overlay frame in place
- The placement of 3800 **LOAD** channel command words (CCWs) within a 3800 spool file

Note: The form name is not displayed when the **TBL** option is used.

One line of information is displayed for each print file specified.

spoolid selects the specified spool file for display (**READER/PRINTER/PUNCH** option)

CLASS c selects spool files of class 'c' for counting (**FILES** option) or display (**READER/PRINTER/PUNCH** option)

FORM form selects spool files with a particular form name for counting (**FILES** option) or display (**READER/PRINTER/PUNCH** option)

NOHOLD selects spool files that are not held (either by system hold or user hold) for counting (**FILES** option) or display (**READER/PRINTER/PUNCH** option)

HOLD selects spool files that are held (either by system hold or user hold) for counting (**FILES** option) or display (**READER/PRINTER/PUNCH** option)

USERHOLD selects spool files that are in user hold for counting (**FILES** option) or display (**READER/PRINTER/PUNCH** option)

SYSHOLD selects spool files that are in system hold for counting (**FILES** option) or display (**READER/PRINTER/PUNCH** option)

PF[nn] displays the program function key number specified for a 3270 Information Display device, along with its associated command lines. If you do not specify nn, CP displays all 24 program function keys and their associated data lines. The value, nn, is a number from 1 (or 01) to 24. See the CP SET command for an explanation of how to define and use program function keys.

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

SCREEN displays the color and extended highlight values currently in effect for your virtual console. These values are set either by the SCREEN statement in your directory entry or when you issue class G SCREEN command(s).

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

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

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

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

QUERY SET

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

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

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

MSG {ON } {OFF }, SMSG {ON } {OFF }, AFFINITY {NONE } {nn }, NOTRAN {ON } {OFF }, VMSAVE {ON } {OFF }, 370E {ON } {OFF }

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

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

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

QUERY TERMINAL

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

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

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

SCRNSAVE {ON } {OFF }

The settings of all functions that are controlled by the TERMINAL command are displayed. Refer to the discussion of the TERMINAL command for explanations of the functions. If LINEDIT is turned off, the logical editing symbols displayed are those that were in effect before line editing was turned off.

Note: If the user has no physical console, only the first line is issued. Also, the responses after the HILIGHT message do not appear when there is no virtual console defined.

QUERY CPUID

CPUID= aabbbbbbbccccdddd

where:

- aabbbbbbbccccdddd is the 16-digit processor identification associated with the virtual machine.
- aa is the version code; these two digits are forced to X'FF' to identify that the virtual machine is running under VM/SP.
- bbbbbb is the processor identification number; this field contains six hexadecimal digits. This is the only part of the CPUID that can be modified by means of the SET CPUID command or set by the VM/SP directory's OPTION control statement.
- cccc is the model number of the processor you are running on; this field is padded to the left with zeroes when you are not running on a 303X or 4300 processor. This field defaults to the model number of the real machine. In MP mode, this value is the model number stored in the PSA (prefix storage area) of the IPL processor. If the IPL processor is not online at the time of the QUERY, CP displays the model number of the processor that is online at the time the QUERY command is issued.
- dddd is the machine check extended logout; this field is forced to X'0000' since CP does not reflect machine checks back to the virtual machine.

Note: If the CPUID is not supplied by the directory OPTION statement or the SET CPUID command, then the CPUID associated with virtual machine(s) is as follows:

```

                                aabbbbbbbccccdddd
                                A A      A  A
FF(virtual mach)-----J |      |  |
real CPUID-----J      |  |  |
real S/370 Model No.-----J  |  |
all zeros-----J-----J
    
```

QUERY FILES

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

CP displays the total number of spool files in your system. If you specify the CLASS, FORM and/or NOHold, HOld, USERhold, or SYShold options with QUERY FILES, CP displays only the totals for the options you specify rather than for all files on your system.

QUERY

QUERY VIRTUAL CHANNELS

CHANNELS= {SEL}
 {BMX}

The operating mode of the virtual machine channels is displayed. This response applies to all of the virtual machine channels except channel 0, which is always a byte multiplexer channel, and any channels with virtual or real channel-to-channel adapters, which are always selector channels.

QUERY VIRTUAL GRAF

GRAF vaddr {ON DEV raddr}
 {NOT READY }

The status of all locally attached virtual display devices defined to your virtual machine is displayed.

where:

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

raddr is the real address of the device.

NOT READY shows the status of a virtual display device that has not been attached via the DIAL command.

QUERY VIRTUAL CONSOLE

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

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

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

vaddr FLASH ovly CHAR ctab MDYF cmod FCB pcp

For virtual machine consoles and SNA terminals, a four-line response is displayed. The first line shows the console status and options and the next three lines show the virtual console spooling status.

where:

vaddr is the virtual address of the virtual machine console.

raddr is the real address of the terminal associated with the virtual console.

c is the spooling class of the console.

nnn is the number of copies spooled.

userid is the user identification.

distcode is the distribution code.

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

ctab is the name of the character arrangement table used to generate output.

cmmod is the name of the copy modification module used to alter output text.

pcpi is the name of the FCB module used in the vertical formatting of a page.

ccc is the flash count for the file.

luname is the logical unit name for the SNA terminal simulating a virtual console. The response to the QUERY VIRTUAL CONSOLE command displays the luname rather than the real address (raddr) and the GRAF/LINE option. Luname is a one-to-eight character identifier assigned to a SNA terminal user and use by CP instead of raddr. It is valid for SNA terminals connected to the host VM/SP system by the VTAM Communications Network Application (VM/VCNA) program product.

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

The default settings for a virtual console are:

```
CONS vaddr ON DEV raddr TERM STOP
      vaddr CL T NOCONT NOHOLD COPY 001 READY
      vaddr FOR userid DIST distcode FLASHC 000
      vaddr FLASH CHAR MDY FCB
```

The default settings for a SNA terminal simulating a virtual console are:

```
CONS vaddr luname TERM STOP
      vaddr CL T NOCONT NOHOLD COPY 001 READY
      vaddr FOR userid DIST distcode FLASHC 000
      vaddr FLASH CHAR MDY FCB
```

QUERY

QUERY VIRTUAL DASD

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

The status of each virtual minidisk defined for your system is displayed. where:

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

type is an IBM direct access device.

Notes:

1. A display of 3340 represents either a 3340 or 3344 device.
2. For a dedicated DASD unit, the response is:

DASD vaddr ON DASD raddr

where:

vaddr is the virtual address to which the FB-512 device is attached.

raddr is the real address of the FB-512 device.

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 for count-key-data devices.

nnnnnn is the number of blocks on the virtual disk for FB-512 devices.

QUERY VIRTUAL TAPES

TAPE vaddr ON DEV raddr

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

where:

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

raddr is the real address of the tape.

QUERY VIRTUAL LINES

LINE vaddr ON DEV raddr

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

where:

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

raddr is the real address of the line.

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

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

vaddr { TO }
 { FOR } userid DIST distcode

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

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

vaddr FLASH ovly CHAR ctab MDYF cmod FCB pcpi

vaddr SIZE ww ll { 2WCGM } { BTS } { DATCK }
 { 4WCGM } { CFS } { NODATCK }

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

QUERY

Note: The last line of the response is displayed only for a virtual 3800 printer.

where:

vaddr is the virtual device address of the virtual printer or punch.

c is the output class assigned to spool files produced from the device.

[*]nnn is the number of copies from 1 to 255, of each output file to be produced. When the optional asterisk (*) appears, the 3800 performs the duplication internally, provided the spool file contains sufficient information to be spooled to the 3800 printer. If there is insufficient information, the asterisk is ignored and printing occurs normally on the printer assigned in the spool file class.

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

TO userid indicates that the output from the device, when closed, becomes a reader input spool file for the indicated userid.

FOR userid indicates the userid identification (spool file owner) assigned to spool files produced from the device.

distcode is the distribution code assigned to each spool file produced from the device.

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

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

ctab is the name of the character arrangement table used to generate output.

cmo is the name of the copy modification module used to alter output text.

pcpi is the name of the FCB module used in the vertical formatting of a page.

ww indicates the form width code (in hexadecimal) for a virtual 3800 printer.

ll indicates in decimal notation the length of the paper in half-inches loaded into a 3800 printer.

Note: The distcode in this case indicates the FOR userid; however, the distcode produced on the output files when the file is closed is the distcode assigned to the FOR userid as specified in the VM/SP directory.

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

The default settings are:

```
PUN vaddr CL A NOCONT NOHOLD COPY 001 READY FORM form
    vaddr FOR userid DIST distcode

PRT vaddr CL A NOCONT NOHOLD COPY 001 READY FORM form
    vaddr FOR userid DIST distcode FLASHC 000
    vaddr FLASH CHAR MDY FCB
    vaddr SIZE OF 11 4WCGM CFS NODATCK
```

where:

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

QUERY VIRTUAL STORAGE

STORAGE = nnnnnK

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

QUERY VIRTUAL ALL

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

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

QUERY VIRTUAL vaddr

QUERY VIRTUAL vaddr-vaddr

The response is in the same form as QUERY VIRTUAL DASD, TAPES, LINES, or UR, depending on virtual device type. When a range of addresses is specified a response is used for each device within that range.

If vaddr corresponds to a 3330V (Mass Storage System's virtual 3330 volume) address, the following response is issued:

```
DASD vaddr ON DASD vaddr
```

If vaddr on the command line is a Mass Storage Control 3851 port address, the following response is issued:

```
MSC vaddr ON DEV vaddr
```

QUERY

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 SCREEN

The response you receive is:

| | | | | | |
|----------|-------|-----------|--------|-------|-----------|
| CPOUT | color | exthilght | VMOUT | color | exthilght |
| INREDISP | color | exthilght | INAREA | color | exthilght |
| STATAREA | color | exthilght | | | |

where:

color is the color value currently in effect for your virtual console.

exthilght is the extended highlight value currently in effect for your virtual console.

Note: If you do not have a SCREEN directory statement in your directory entry and you have not issued the SCREEN command during the current terminal session, the response to the QUERY SCREEN command is DEFAULT for the color value and NONE for the exthilght value.

QUERY SECUSER

The response is one of the following:

SECONDARY USER IS UNDEFINED

is the response if no secondary user appears in the user's directory entry. If the primary user disconnects, all output issued during that period is lost.

SECONDARY USER XXXXXXXX IS LOGGED ON

is the response if the secondary user is logged on. If the primary user disconnects, the secondary user receives all output issued until the primary user reconnects.

SECONDARY USER XXXXXXXX IS NOT LOGGED ON
 is the response if the secondary user is not logged on. If the primary user disconnects, all output issued is lost until the primary user reconnects or the secondary user logs on.

SECONDARY USER XXXXXXXX IS DISCONNECTED
 is the response if the secondary user is disconnected. If the primary user disconnects, all output issued is lost until either the primary user reconnects or the secondary user reconnects.

QUERY READER, QUERY PRINTER, QUERY PUNCH

| Basic Information | | | | | | Additional Information | | | | | |
|-------------------|------|-------|---------|----------|------|------------------------|----------|------|------|-------|-------|
| ORIGINID | FILE | CLASS | RECORDS | CPY HOLD | FORM | DATE | TIME | NAME | TYPE | DIST | |
| userid | file | a typ | norecs | nnn stat | form | mm/dd | hh:mm:ss | name | type | code | |
| Table Information | | | | | | FLASH | CHARS | FCB | MDFY | FLSHC | LOAD |
| | | | | | | ovly | ctab | plpi | cmof | ccc | {NO} |
| | | | | | | | | | | | {BEG} |
| | | | | | | | | | | | {ANY} |

where:

- userid is the user who originally created the file.
- spoolid is a unique, system-assigned number which is used by VM/SP to identify the file.
- c is the spool file class.
- typ is the originating device type (PRT, PUN, CON, DMP, or RDR).
- norecs is the number of logical records contained in the file.
- [*]nnn is the number of copies assigned to the file (it has no effect for virtual reader files). Appearance of the asterisk indicates that spool file duplication is handled by the 3800 printer. The device copies a spool file internally one page at a time (collated output produced).
- stat is the file hold status: NONE (no hold), USER (user hold), SYS (system hold), or USYS (system and user hold).
- form is the form name of the spool file.
- mm/dd is the date the file was created in month/day.
- hh:mm:ss is the time the file was opened for creation in hours:minutes:seconds. For some files, such as spooled console files, this time may be a span of hours before the time at which the file is closed.
- fn is the filename assigned to the file (if any). If the file has a 24-character data set name (dsname), only 18 characters are displayed. These characters extend from the "name" field through the "type" field.

QUERY

ft is the filetype assigned to the file (if any).

distcode is the distribution code assigned to the file.

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

ctab is the name of the character arrangement table used to generate output.

pcpi is the name of the FCB module used in the vertical formatting of a page.

cmod is the name of the copy modification module used to alter output text.

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

NO indicates that no 3800 LOAD channel command words (CCWs) are positioned within the 3800 spool file.

BEG indicates that 3800 LOAD channel command words (CCWs) are positioned at the beginning of the 3800 spool file.

ANY indicates that 3800 LOAD channel command words (CCWs) are interspersed throughout the 3800 spool file.

When you issue QUERY READER, QUERY PRINTER, or QUERY PUNCH commands, CP responds by listing (in the form described) all the closed spool files associated with your virtual reader, printer, or punch.

The information listing DATE, TIME, NAME, TYPE, and DIST (date of file creation, time of file creation, filename, filetype and file distribution code) is displayed only when you specify the ALL operand. However, if you specify the TBL operand, the information listing FLASH, CHARS, FCB, MDPY, PLSHC, and LOAD (3800 overlay form, 3800 character arrangement table, 3800 vertical formatting, 3800 copy modifications, the number of copies to be printed by 3800 with overlay form, and the placement of 3800 LOAD CCW's within a 3800 spool file) is displayed.

Once a printer file starts printing, it does not appear in the response to a QUERY PRINTER command.

If a reader file appears to be lost, due to the virtual machine abnormally terminating without closing the reader, you can issue the command CLOSE READER HOLD to try to recover the reader file.

To terminate the displaying of the response to a QUERY READER, QUERY PRINTER, or QUERY PUNCH command, press the Attention key (or its equivalent).

Note: For further information on altering spool file attributes, refer to the CHANGE command in this section.

QUERY PFnn

```

PFnn {IMMED} {tab... }
      {DELAY} {copy... }
          {pfdata1...}

```

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

PFnn UNDEFINED

Notes:

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

QUERY VMSAVE

```

USER  RECEIVER SYSNAME PRTY IPLD ENBLD IN USE DATE/TIME VOLID
USERF  USERA   AREA 6  0015 NO  NO    NO      N/A  TND198
0001-0002 0006

```

where:

- USER is the userid of the user who can store into the area or IPL the area (the owner of the area).
- RECEIVER is the userid of the user who can IPL the area.
- SYSNAME is the name of the area specified on a SET VMSAVE or IPL command.
- PRTY is the priority number of this area.
- IPLD=YES indicates that either the owner or the receiver is currently loaded from the area.
- ENBLD=YES indicates that the owner has issued a SET VMSAVE for this area.
- IN USE=YES indicates that a VMSAVE area that overlaps this one is in use by some user (the area has been loaded, VMSAVE has enabled, or a system is already saved in the area).
- DATE/TIME=NONE indicates that no system is currently stored in this area.
- DATE/TIME=N/A indicates one of the following:
 - The DASD volume is not mounted or is not CP owned.
 - An overlapping area is enabled for VMSAVE or contains a valid system.

QUERY

VOLID is the volume serial number of the target volume.

The second and subsequent lines indicate the page range associated with this area. A page range is indicated by the first and last page numbers of the range separated by a hyphen. A single page is simply indicated by the page number. For a saved system, these pages are the ones that were stored, and the ones that are loaded by the IPL command; for a virtual machine with VMSAVE enabled, this is the list of pages that are saved when and if the virtual machine is terminated by VM/SP or when VM/SP itself terminates. See the discussion on the VMSAVE option in the VM/SP System Programmer's Guide for more information regarding the specification and use of the VMSAVE option.

QUERY S370E

```
S370E {ON } {PROC xx, {ON } PROC yy}
      {OFF} | {OFF} |
```

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

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

QUERY PROCESSOR

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

where:

mm is the processor ID number of the main processor if VM/SP is running in UP or AP modes; in MP mode, it is the processor ID number of the IPL processor.

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

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

QUERY SPMODE

SPMODE {ON }
 {OFF}

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

Note: When VM/SP is AP or MP generated and SPMODE is off, the user can reset the affinity setting of the virtual machine after the attached processor or non-IPL processor has been varied online -- VM/SP has resumed AP or MP mode.

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

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

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

The format for this QUERY command is:

```

Query | { 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, the abbreviation DSC is printed instead of the line address.

USERS

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

USERS userid

userid

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

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

QUERY

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 } }, ...

VSM      - {userid}, ...

userid - {luname}, ...

Lnnn     - vaddr
```

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. The userid - luname response displays the userids and logical unit names (lunames) of all users accessing the VM/SP system in a SNA network through the VTAM Service Machine (VSM). The VSM - userid displays the userids of all users connected through that particular VSM to the host VM/SP system. If your installation has the VM/Pass-through program product, Lnnn displays the logical device name of the user accessing the host VM/SP system through the Pass-through virtual machine; vaddr is logical address of that user.

QUERY USERS

nnn USERS, mmm DIALED, xxx NET

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

where:

nnn is the total number of logged-on users.

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

xxx is the total number of users logged on through a VSM in a SNA network.

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

QUERY userid or QUERY USERS userid

userid - { raddr }
 { resid }
 { luname }

The real address (raddr) or the resource identification of a 3704/3705 line (resid) to which the specified user is connected is displayed. The luname is displayed instead of the raddr or resid if that user is logged on through a SNA network using the VTAM Service Machine.

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 interrupt pending at your virtual console. The format of the REQUEST command is:

```
Request |
```

Usage Notes

1. The REQUEST command performs the same functions as ATTN and the two commands can be used interchangeably.
2. CP ignores the REQUEST command if any I/O operation is in progress at the console or if other interrupts are pending. This condition may result if the user issues the REQUEST command by means of DIAGNOSE X'8'.

Responses

None.

RESET

RESET

Privilege Class: G

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

```
RESET | vaddr
```

where:

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

Usage Notes

1. In addition to clearing all pending interrupts, all error conditions occurring as a result of unit checks and virtual sense bytes are reset.
2. When the DIAL command is issued from a local 3270 terminal, the virtual system user must use the CP command RESET to drop the dialed connection.

Responses

DEVICE RESET

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

REWIND

Privilege Class: G

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

```
REWIND | vaddr
```

where:

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

Usage Note

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

Responses

REWIND COMPLETE

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

REWIND NOT PERFORMED

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

color specifies the color attribute you want to assign to an 'area' of your screen. Color can be:

BLUe -- blue
RED -- red
GREen -- green
YELlow -- yellow
TURquois -- turquoise
PINK -- pink
WHItE -- white
DEFault -- monochrome is the default.

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

BLInk -- blinking
REVvideo -- reverse video
UNDerlin -- underlining
NONE -- no extended highlighting

Usage Notes

1. If you issue the SCREEN command from a terminal that does not support color and/or extended highlighting, CP still sets the values you specify for your virtual machine console. Then, if you disconnect from that terminal and logon at a terminal that does support color and/or extended highlighting, the values you set are implemented for your virtual machine after the reconnection. The color and extended highlighting values remain in effect for the duration of the current terminal session unless you alter them with other SCREEN commands.
2. You cannot specify a particular 'area' operand more than once on the same SCREEN command line.
3. If you enter the SCREEN command and you fail to enter any operands (or enter invalid operands), command processing terminates, you receive an error message, and the color and extended highlighting values in effect for your terminal remain unchanged by the invalid command.

SCREEN

Example:

If you enter:

SCREEN OUTAREA RED NONE INAREA GREEN BLINK STATAREA PINK UNDERLIN

CP sets the following values for your terminal:

| <u>AREA</u> | | <u>COLOR</u> | <u>EXTHIGHLIGHT</u> |
|-------------|-----------|--------------|---------------------|
| cpout | } outarea | red | none |
| vmout | | red | none |
| inredisp | | red | none |
| inarea | | green | blinking |
| statarea | | pink | underlining |

Response

None.

SEND

Privilege Class: G

Use the SEND command to pass commands and message replies from the secondary user's console to disconnected virtual machines for execution.

The format of the SEND command is:

```

| SEND | [CP] userid [text] |

```

where:

CP indicates that CP executes the text in behalf of the receiving virtual machine. If the first three bytes of text are '#CP' or if the receiving virtual machine is already in CP READ state, the user need not specify the CP operand when sending a CP command. If the SEND command does not include text, CP puts the receiving virtual machine into console function mode. This operand is equivalent to the #CP command.

userid represents the disconnected virtual machine that has a specified secondary user (secuser) in its directory entry.

text can be any command allowed in the virtual machine command environment or a reply to a message issued by the disconnected virtual machine. CP passes the text to the disconnected virtual machine for processing. If the user wants to send multiple lines in the text, he must use the logical line end (#) immediately preceded by the logical escape character ("); in addition, the user must have line editing enabled in his virtual machine.

Usage Notes:

1. The issuer of the SEND command must be authorized by the secondary user (secuser) by means of the secuser operand of the CONSOLE statement in the directory entry of the receiving virtual machine.
2. The receiver of the SEND command input must be running disconnected.

Responses

None.

SET

Privilege Class: G

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

```

SET {
  ACNT
  AUTOPoll
  MSG
  SMsg
  WNG
  MSG { ON
  RUN { OFF
  LINEDit
  ECmode
  ISAM
  NOTRans
  PAGEX

  EMSG { ON
        { OFF
        { CODE
        { TEXT

  TIMER { ON
        { OFF
        { REAL

  AFFinity { [
            { ON
            { OFF
            ]

  CPUid bbbbbb

  ASsist { [ ON ] [ SVC ] [ TMR ]
        { [ ] [ NOSVC ] [ NOTMR ]
        { OFF

  PFnn [ IMMed ] [ pfdatan ]
       [ DELayed ]

  PFnn [ TAB n1 n2 ... ]
  PFnn [ COPY [resid] ]
  PFnn [ COPY [cuu] ]
  PFnn [ COPY [luname] ]
}

```

(Part 1 of 2)

| | | | | | |
|----------|----------|--------|--------------|----------|-----|
| VMSAVE | { ON } | | | | |
| | { OFF } | | | | |
| | { name } | | | | |
| STBypass | { | nnnnnK | { NOVERIFY } | { } | |
| | { | nnM | { } | { } | |
| | { | VR | | | |
| | { OFF } | | | | |
| STMulti | { | n | { USEG xx } | CSEG yyy | { } |
| | { | ON | { } | | { } |
| | { | OFF | | | |
| 370E | { ON } | | | | |
| | { OFF } | | | | |

(Part 2 of 2)

where:

ACNT { ON }

{ OFF }

controls whether accounting information is displayed at the terminal or not (ON and OFF, respectively) when the operator issues the CP ACNT command. When you log on VM/SP, ACNT is set on.

AUTOPOLL { ON }

{ OFF }

controls whether or not CP tests each BTAM autopoll CCW to see if it has been dynamically modified. ON indicates that (1) CP bypasses testing, and (2) the VS1 operating system is to notify CP, via the DIAGNOSE instruction code X'0028', after an autopoll CCW has been modified. ON substantially reduces the overhead required by CP to service BTAM autopoll channel programs. OFF causes CP to check each autopoll CCW after each execution of the channel program and is the default status when the user logs on.

Note: If you SET AUTOPOLL ON and the virtual operating system does not have the appropriate Diagnose interface, CP does not detect changes to BTAM channel programs and unpredictable results can occur. OS/VS 1 Release 6.0, however, does have this interface.

MSG { ON }

{ OFF }

controls whether messages sent by the MSG command from other users are to be received at the terminal. If ON is specified, the messages are displayed. If OFF is specified, no messages are received. In addition to controlling messages generated by the MESSAGE command, spooling messages generated by users sending punch, printer or reader files to another virtual machine are also suppressed if OFF is specified. When you log on VM/SP, MSG is set on.

SET

SMSG {ON }
{OFF }

controls whether a virtual machine is receiving special messages or not. If OFF is specified, the virtual machine cannot receive special messages.

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/SP, WNG is set on.

IMSG {ON }
{OFF }

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

LINEDIT {ON }
{OFF }

controls the line editing functions. ON specifies that the line editing functions and the symbols of the VM/SP system are to be used to edit virtual processor console input requests. This establishes line editing features in systems that do not normally provide them. OFF specifies that no character or line editing is to be used for the virtual machine operating system. When you log on VM/SP, LINEDIT is set on. The LINEDIT operand is unaffected by the setting of the APL/TEXT operand of the TERMINAL command.

ECMODE {ON }
{OFF }

controls whether the virtual machine operating system may use System/370 extended control mode and control registers 1 through 15. Control register 0 may be used with ECMODE either ON or OFF. When you log on VM/SP, ECMODE is set according to the user's directory option; ON if ECMODE was specified and OFF if not.

Note: Execution of the SET ECMODE {ON|OFF} command always causes a virtual system reset.

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

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

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

Any of the above conditions forces CCW translation.

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

PAGEX { ON }
 { OFF }
 controls the pseudo page fault portion of the VM/VS handshaking feature. PAGEX ON or OFF should only be issued for a virtual machine that has the VM/VS handshaking feature active. It can only be specified for a virtual machine that has the extended control mode (ECMODE) option. PAGEX ON sets on the pseudo page fault portion of handshaking; PAGEX OFF sets it off. When you log on to VM/SP, PAGEX is set OFF. Also, each time you IPL your virtual machine, PAGEX is set off. If you want to use the pseudo page fault handling portion of handshaking, you must issue SET PAGEX ON after you load your operating system.

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

SET

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. Also, data written by means of type in, such as a list of names following a message, are not controlled by the EMSG operand.

TIMER { ON
OFF
REAL }

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

AFFINITY [OFF]
[ON]

controls whether or not CP dispatches a virtual machine and execute its programs on either the main processor, the attached processor of an attached processor system or the nonIPL processor of a multiprocessor system. Normal mode is AFFINITY OFF, allowing virtual machine execution on either processor. AFFINITY ON causes the directory to be interrogated for an affinity setting; if there is none in the directory, the user receives an informational message. In uniprocessor mode, affinity is meaningless because all virtual machines execute on the main processor.

When affinity is in effect for a virtual machine, only the specified processor runs the virtual machine. This is particularly useful for virtual machines that require features or performance enhancements that are only available on one processor in an attached processor or multiprocessor system. CP instructions (such as privileged operation instructions) executed on behalf of a virtual machine are not restricted to a processor by an affinity setting.

CPUID bbbbbb

CPUID is the keyword that provides a means of setting or changing the CPU identification for a virtual machine. The variable value bbbbbb represents a string of six hexadecimal digits that permit unique identification of the CPU. This command allows the user to provide a unique CPU identification to be stored in response to the 'STIDP' instruction. It is necessary to associate a unique CPUID with each virtual machine that is attached to an MSC port since solicited/unsolicited messages are directed to the host system by means of the CPUID. There is no checking by VM/SP to ensure that all virtual machine using the SET CPUID command have specified unique CPU serials. The hexadecimal field 'bbbbbb' is the CPU identification number (CPU serial).

Note: The CPU identification number (serial) is only a portion of the complete CPUID. The CPUID identification stored in response to a STIDP instruction is a string of 16 hexadecimal digits as follows:

aabbbbttbbccccdddd

where

- aa - version code; these two digits are forced to X'FF' to identify that the virtual machine is running under VM/SP.
- bbbbbb - CPU identification number; this field contains six hexadecimal digits. This is the only part of the CPUID that can be set by the DIRECTORY OPTION control statement and modified by means of the SET CPUID command.
- cccc - model number; this field contains a high order 0 digit followed by the three digits of the model number (0-9). This field defaults to the model number of the real machine.
- dddd - machine check extended logout; this field is forced to X'0000' since CP does not reflect machine checks back to the virtual machine.

ASSIST { [ON] [SVC] [TMR]
 [] [NOSVC] [NOTMR]
 [] [] [] }

OFF

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

All, some, or none of the function performed by VM/SP hardware assist may be available on the processor, attached processor, or nonIPL processor that the virtual machine is logged on. (For the extent of VM/SP hardware assist available on supported VM/SP processors, refer to the VM/SP Operator's Guide.) The extent of VM/SP hardware assist available to your virtual machine can be determined by invoking QUERY SET and by the responses to attempts to invoke the various operands of SET ASSIST. The hardware assist functions can only be activated for an individual virtual machine provided the VM/SP system operator permits the associated hardware assist function to be active for the entire system (or in attached processor or multiprocessor systems, the appropriate processor). Overall system control of hardware assist is controlled by SET SASSIST and SET CPASSIST commands; see the VM/SP Operator's Guide for details on these commands.

If VM/SP hardware assist is active for the entire system, then hardware assist is active and used during virtual machine operations immediately after logon. Hardware assist as it pertains to a virtual machine can then be turned off and on again by SET ASSIST OFF and SET ASSIST ON respectively.

SET

The SVC handling portion of the hardware assist is invoked when you log on unless your VM/SP directory entry has the SVCOFF option. Issue the QUERY SET command line to see if the hardware assist is activated and whether the hardware assist or VM/SP is handling SVC interruptions. All SVC 76 requests are passed to CP for handling, regardless of the SVC and NOSVC operands. If you issue the SET ASSIST command line and specify SVC or NOSVC while the hardware assist is turned off, the appropriate bits are set. Later, if hardware assist is turned on again, the operand you specified while it was off becomes effective. ON sets the hardware assist on for the virtual machine; OFF turns it off. SVC specifies that the hardware assist handles all SVC interruptions except SVC 76 for the virtual machine; NOSVC means VM/SP handles all the SVC interruptions.

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

The TMR and NOTMR operands control the method of updating the virtual interval timer (virtual address X'50'). If TMR is specified, the virtual interval timer assist hardware updates the virtual interval timer in the same way that the real interval timer is updated, that is, 300 times per second. If NOTMR is specified, the virtual interval timer is updated only when CP is given control. This can be as seldom as once per time slice interval.

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

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

defines a program function for a program function key on a 3270 Information Display station and indicates when that function is to be executed. If PFnn is specified with no operands, that program function key is considered "undefined." See the VM/SP Terminal User's Guide for a description of how to use the 3270 program function keys.

In PFnn, nn is a number from 1 (or 01) to 24 that corresponds to a key on a 3270 display station. The program function is a programming capability you create by defining a series of VM/SP commands or data you want executed. This series of commands is executed when you press the appropriate program function key.

IMMED specifies that the program function is executed immediately after you press the program function key.

DELAYED specifies that execution of the program function is delayed for a display terminal. When the program function is entered, it is displayed in the input area and not executed until you press the Enter key. DELAYED is the default value for display terminals.

pfdata1#pfdata2#...pfdata_n
 defines the VM/SP command or data lines that constitute the program function. If more than one command line is to be entered, the pound sign (#) must separate the lines. If you use the pound sign (#) to separate commands that you want executed with the designated PF key, you must precede the command line with #CP, turn line editing off, or precede each pound sign with the logical escape character ("). For further explanation, see the "Usage Notes" section that follows. If no command lines are entered following the PF_{nn} designation, then the program function is marked undefined. Program functions cannot be embedded within one another.

PF_{nn} [TAB n1 n2 ...]
 specifies a program function key number to be associated with tab settings on a terminal. The number of the PF key, nn, can be a value from 1 (or 01) to 24. If the TAB keyword and tab settings are not entered, that program function key is considered "undefined." For examples of how this feature is used, see the VM/SP CMS User's Guide. TAB is a keyword identifying the tab function. The tab settings (n1 n2 ...) may be entered in any sequence.

Note: When defining the TAB function, the existence of blanks preceding the keyword TAB or the lack of trailing tab settings forces the PF key to be treated as pfdata.

PF_{nn} [COPY [resid]]
 specifies that the program function key numbered nn performs a COPY function for a remote 3270 Information Display terminal. The number, nn, must be a value from 1 (or 01) to 24. The COPY function produces a printed output of the entire screen display at the time the PF key is actuated. The output is printed on an IBM printer connected to the same control unit as your display terminal. The PF key COPY function is not available for dedicated 3284, 3286, 3287, 3288, and 3289 printers. The resource identification, resid, may be specified if more than one printer is connected to the same control unit as your display terminal. It is a 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 designated program function key, PFnn, performs a COPY function for a local 3270 Information Display terminal. The number of the PF key, nn, must be a value from 1 (or 01) to 24. When the PF key is pressed, the COPY function produces a printed output of the entire local screen display except for the status field, which is replaced with blanks. The cuu represents the real hardware address of the display printer, and may specify a printer that is on a different control unit than the one to which your 3270 is attached. If you do not specify cuu, the printer with the lowest cuu that is available on the same control unit as your 3270 is selected.

PFnn [COPY [luname]]

specifies that the program function key numbered nn performs a COPY function for a SNA terminal. The number, nn, must be a value between 1 (or 01) to 24. The COPY function produces a printed output of the entire screen display at the time the PF key is pressed. The output is printed on an IBM printer connected to the same control unit as your display terminal. CP does not know the real address (raddr) of the logical unit (LU device), so you must specify a one to eight character luname (logical unit name).

Notes: For both remote and local COPY functions:

1. You receive a NOT ACCEPTED message, displayed in the screen status field of your 3270, if any of the following situations occur:
 - The printer is already busy, or all printers are busy.
 - The printer is turned off.
 - The printer is operational but has not been varied online by the system operator.
 - The printer is out of paper or is in any other intervention-required condition.
 - The designated printer does not support the copy functions. See VM/SP Planning and System Generation Guide for a list of 3270 Information Display printers that support the COPY function.
 - The SET PFnn COPY command format is invalid.

You may include your own identification on the printed output by entering the data into the user input area of the screen before you press the PF key. The identification appears on the last two lines of the printed copy.
2. To insure the definition of a PF key for the COPY function, you can either start with one leading blank before the keyword COPY or follow the keyword with at least five blanks.
3. The existence of more than one data field (cuu or resid) following the keyword COPY identifies the string as pfd data rather than a hardcopy request.

VMSAVE { ON
 OFF
 name }

controls the status of the VMSAVE option. ON enables the option for the user with exactly one DASD area. OFF disables the option for the user. The name operand allows a user with one or more DASD areas to designate which DASD area is to be enabled. When the name operand is used, the name must be identical to the name used in the SYSNAME parameter of the NAMESYS macro in the DMKSNT module. The DEFINE STORAGE command turns VMSAVE off. See the VM/SP System Programmer's Guide for more information. Note, when specifying the VMSAVE option, the virtual machine size cannot exceed eight megabytes.

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

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

nnnnnK defines in thousands of bytes the highest contiguous storage address starting from zero where the virtual and real addresses are equal for a virtual machine running in a V=V area. The value is rounded down to the nearest full page boundary.

nnM defines in millions of bytes the highest contiguous storage address starting from zero where the virtual and real addresses are equal for a virtual machine running in a V=V area.

NOVERIFY prevents the scanning of a virtual machine's page and segment tables.

VR eliminates shadow tables for a virtual machine running in the VM/SP V=R area.

OFF terminates the shadow table bypass function.

STMULTI { [n [USEG xx] CSEG yyy]
 |ON|
 |
 OFF }

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

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

ON establishes the system default of 3 for the number of shadow tables.

USEG xx defines the size of the contiguous preallocated pool of shadow page tables. xx is the number of shadow page tables in each pool (one pool per set of shadow tables) and ranges from 0 through 99. The USEG operand is only valid when STMULTI n or ON was previously specified.

SET

CSEG yyy defines the number of full segments at the high end of the address space that are shared by all address spaces. The value of yyy ranges between 0 and 128 and represents the number of shadow page tables to be allocated for CSEG.

OFF sets the number of shadow tables to 1.

370E {ON }
{OFF }

provided the System/370 Extended (hardware) special feature has been installed and the VM/SP system operator has enabled it, this operand allows you to execute the OS/VS2 MVS/System Extensions Program Product (program number 5740-XE1) or any other operating systems capable of using the 370E feature.

To determine whether the product has been installed and enabled, issue the QUERY S370E command. To determine the status of the feature for your virtual machine, issue the QUERY SET command.

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

Usage Notes

1. System default values for SET operands or the values set by previously invoked SET command operands can be determined by the use of the QUERY command.
2. SET TIMER REAL must be specified if an interrupt-driven system such as Customer Information Control System (CICS) is to be run in a virtual machine.
3. Both SET PFnn TAB and SET PFnn COPY are executed immediately upon pressing the appropriate program function key. If you insert the keywords DELAYED or IMMED after the PFnn operand, the command is accepted, but the keywords are ignored. The program function still executes immediately.
4. In the case of systems with attached processors or multiprocessors, the VM/SP hardware assists may be installed on none, one, or both processors. Similarly, hardware assist, if equipped on both processors of an attached processor or multiprocessor system, can be activated for none, either, or both systems by the system operator. Thus, with the AFFINITY setting of a virtual machine, the user can receive no response indicating that VM/SP hardware assist is installed and active on his processor. In attached processor or multiprocessor setups where the user has no processor preference (AFFINITY=NONE) upon entering SET ASSIST he too receives no response if either processor has VM/SP hardware assist active. Any error message is indicative that VM/SP hardware assist is not available or active for user. If the user receives a message indicating that VM/SP hardware assist is not available on a particular processor it is also implying that it is available on the other processor. Therefore, the user, to take advantage of VM/SP hardware, need only change his affinity setting, then initiate the SET ASSIST command.

5. If you plan to issue an ADSTOP command, you must first issue a SET RUN OFF command. This command allows the virtual machine to halt execution at the address specified on the ADSTOP command line.
6. If you use the SET PFnn command to set up a series of concatenated commands, you should be careful of line-editing characters. The following examples demonstrate two methods for entering a series of commands:

Example 1

Enter the following command while in CMS mode:

```
set linedit on
```

and press the enter key.

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

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

```
-- or --
```

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

```
-- or --
```

```
set pf02 immed q rdr"#q prt"#q pun
```

and press the enter key.

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

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:

```

[ [SEC]
|nn |MIN|
[ [HRS]
  
```

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

Usage Notes

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

Responses

None.

SMSG

Privilege Class: G

Use the SMSG command to send a special message to a virtual machine programmed to accept and process the message. The format of the SMSG command is:

```

| Smsg      |      userid msgtext
|-----|-----|

```

where:

userid is the receiving virtual machine's logon ID; if you specify userid as "*", it indicates the message is to be sent to yourself.

msgtext is the text of the message that is to be transmitted. Because the input buffers on various console devices differ, the length of the message is limited by the number of characters remaining on the input line after entering the command and the userid. For example: A 3270 display device contains a 139 character input buffer. If you enter:

| | |
|---------------------|------------------------|
| <u>smg pd58gh1c</u> | <u>special message</u> |
| 14 characters used | 125 characters left |
| <u>sm u</u> | <u>special message</u> |
| 5 characters used | 134 characters left |

Note: Although the 3270 display station contains 139 characters in its input area, CP only reads the first 134 characters. When the SMSG command is issued by DIAGNOSE X'8', CP reads only the first 132 characters.

Usage Notes:

1. Special messages do not appear on the specially programmed virtual machine's screen because the messages are designed to be instructions that are passed as data parameters. The special messages are stored in the prespecified VMCF data receiving area as they are received, and the instructions are processed by the programmed virtual machine.
2. Virtual machines can only receive and process special messages under these two conditions: (1) the virtual machine has issued the SET SMSG ON command, and (2) the user wishing to receive special messages has issued an AUTHORIZE with the Virtual Machine Communication Facility (VMCF) of VM/SP.

Note: Refer to VM/SP System Programmer's Guide for more information on the AUTHORIZE function of VMCF.

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 default values for the SPOOL command appear under the "Usage Notes" following the SPOOL operand descriptions. The format of the SPOOL command is:

| | | | |
|-------|-----------|---|--|
| SPOOL | {Reader} | {Class {c} CONT HOLD EOF | } ¹ |
| | {vaddr} | {* NOCont NOHold NOEOF | |
| | | | |
| | {Printer} | {To {userid} HOLD CONT | } ¹ |
| | {PUnch} | {For * NOHold NOCont [Class c] [Copy[*]nnn] | |
| | {CONsole} | {SYSTEM | |
| | {vaddr} | {OFF | |
| | | {CLOSE Flash name nnn FORM {form} Dist {distcode} | } ¹ |
| | | {PURGE MODify name {OFF} {OFF} | |
| | | {Chars name | |
| | | {FCB name | |
| | | | {START} ² {TERM} ² |
| | | | {STOP} {NOTERM} |

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

²These options apply only to a virtual spooled console.

where:

READER modifies the options for all reader spool devices.
RDR

PRINTER modifies the options for all printer spool devices.
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.

SPOOL

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 alphanumeric field whose values can be A through Z, or 0 through 9.
{*}

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

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

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

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

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

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

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

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

[TO] {userid}
 *
 (SYSTEM)

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

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

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

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

```
spool pun off
```

```
-- or --
```

```
spool pun for usera
```

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

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

FOR {userid}
 *
 (SYSTEM)

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

- OFF** resets the transferred spool option.

SPOOL

COPY[*]nnn

is the number of copies that are to be printed or punched when the file is spooled to the real unit record device. This operand is valid only for output files; the number of copies, nnn must be between 1 and 255 (leading zeros need not be specified). When the optional asterisk (*) is specified, output files spooled to the 3800 printer (see "Usage Notes") are duplicated inside the device one page at a time. If a file is not spooled to the 3800 printer, the function performed by specifying the asterisk (*) is ignored and duplication occurs normally.

CLOSE

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

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

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

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

PURGE

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

FLASH name nnn

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

MODIFY name

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

CHARS name

specifies the name of the character arrangement table used when printing a file. The name must be from one to four characters.

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

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

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

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

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

- it is the result of CP commands entered from CP mode.
- it is a message or warning.
- it is the result of a CP command entered by means of the #CP command.
- it is the result of Commands entered on a 3270 in CMS EDIT mode.
- the output line immediately precedes a virtual machine read to the console. (This output line prints twice in the spooled console file.)

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

FORM form

specifies the form name for spool files created on this virtual device. 'form' is a one to eight characters name defined during system generation. This form can be overridden by the **FORM** option of the **CLOSE** command.

FORM OFF

resets the form for the virtual device to the system default as specified by the installation. When a virtual device is defined (at logon or with the **DEFINE** command), its form is set to this default.

SPOOL

DIST distcode

specifies the distribution code for spool files created on this virtual device. 'distcode' is one to eight characters. This distcode can be overridden by the DIST option of the CLOSE command.

DIST OFF

resets the distribution code for the virtual device to the default specified in the directory for this virtual machine. When a virtual device is defined (at logon or with the DEFINE command), its distcode is set to this default.

Usage Notes

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

| Spool File | Default Options |
|------------|--|
| Reader | NOHOLD NOCONT CLASS * ¹ EOF |
| Punch | NOHOLD NOCONT CLASS A ¹ OFF COPY 001 |
| Printer | NOHOLD NOCONT CLASS A ¹ OFF COPY 001 FLASHC 000 |
| Console | NOHOLD NOCONT CLASS T ¹ OFF COPY 001 TERM |

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

Spool Class Considerations

Your system programmer can define certain spool classes to automatically print certain classification titles (for, example, TOP PRIORITY or CONFIDENTIAL). The title prints on the separator page, and, optionally, at the bottom of each printed page. Ask your system programmer if he has defined certain classes for this function, or request that classes be defined. For example, you may wish class P to print TOP PRIORITY on the separator page, and class X to print CONFIDENTIAL on the separator page and also on the bottom of each page of output.

Spooling Output to a 3800 Printer

In order to print on a 3800, the user can:

- Specify the correct FLASH name that was designated by the operator in the START command.
- Specify the correct spool file class, previously assigned to the 3800 printer by the installation.

A user can also nullify any previous or existing settings of the CHARS, MODIFY, or FCB options by specifying NULL in the name field of these special 3800 operands. However, if a spool file containing 3800 options is spooled to a printer other than the 3800, the options are ignored and printing occurs normally.

Also, the user can avoid FCB forms mismatch by specifying 6, 8, or 12 in the name field of the FCB option to the SPOOL command. These special names force the entire spool file to print at 6, 8, or 12 lines per inch respectively. VM/SP calculates the proper FCB to load for these special names.

Notes:

1. When the 3800 printer is activated, the operator specifies the character arrangement table and the FCB value that prints the separator page for every output file. If a user omits the CHARS option on a file spooled to the 3800, the printing defaults to the character arrangement table previously defined by the operator. This is also the case for the FCB option.
2. VM/SP issues a message to the user whenever a file spooled to the 3800 printer contains invalid or undefined option names. If the operator has not automatically purged the file, the user can correct the situation via the CHANGE command or spool the file to another printer.

The Spoolid Number: An Identifier for Your Spool Files

Once you close a spool file by issuing the CMS PRINT or PUNCH command or the CP CLOSE command, CP assigns the spool file a number between 1 and 9900. This number is called the spoolid (spool file identification) for the file. It can be used as a convenient way to identify the file. It can also be used when you are manipulating the file with VM/SP spooling commands such as ORDER, CHANGE or CLOSE.

Spoolids are assigned to all your spool files sequentially. When the maximum number (9900) is assigned, CP begins again with the number 1.

When you print or punch a file, CP displays at your terminal the spoolid it assigned to your file if (1) multiple copies of the file are being processed, (2) the file is being transferred to another user, or (3) the file is placed in a user hold status. You can find out various kinds of information about a file using the spoolid with the many forms of the QUERY command.

Transmitting Spool Files Locally

Spool files can be transferred between users via the SPOOL and TRANSFER commands. To transfer printer or punch files of your virtual machine to the virtual reader of some other user, issue the SPOOL command with the TO option:

```
spool device to userid
```

where device is the virtual device type or virtual address (for example, PRINTER, PUNCH, 00E) and userid is the name of the designated user's reader.

SPOOL

If you only want to print or punch a file for someone else, you can issue the SPOOL command with the FOR option:

```
spool device for userid
```

where device is the same as the previous example. The userid provides the distribution code for the designated user.

To transfer printer or punch files from your virtual card reader to another user's reader, issue the TRANSFER command with the TO option:

```
transfer option to userid
```

where option is SPOOLID, CLASS, or ALL and userid is the designated user's virtual reader.

If you merely wish to reclaim a transferred file, you can issue the same command with the FROM option:

```
transfer option from userid
```

where option and userid are the same as the previous example.

Notes:

1. Files that have been spooled for another user are not in that user's virtual reader and, therefore, you cannot reclaim them.
2. You can transfer any file that is queued on your virtual readers; however, you can reclaim only those files which you originated.

Transmitting Files to Remote Locations

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

```
tag dev device locid
```

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

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

```
spool device to userid
```

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

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

Once you have closed a spool file that is spooled to another user, you cannot change or query the spool file tag. If the file has not been selected for processing by the RSCS virtual machine, you can use the TRANSFER command to get the file back to your virtual reader. When changes are necessary, you can issue the TAG QUERY DEV or FILE command to determine the TAG information associated with a given virtual device or spool file. You can then use the FILE operand of the TAG command to change a tag that was previously set.

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

```
tag dev punch cambridg
spool punch to net
punch myprog assemble
```

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

Receiving Files from a Remote Spooling Network

If your virtual machine is logged on VM/SP, the remote spooling network notifies you of the arrival of a file for your machine from the RSCS network by displaying a message at your terminal. The file is sent to your virtual card reader.

VM/SP can accumulate files from the network destined for your virtual card reader, regardless of whether you are logged on your virtual machine or not. If you are logged on your virtual machine, issue the QUERY command to see if you have any files in your virtual reader. When you log on your virtual machine, the logon process transmits a message informing you of accumulated spool files residing in your virtual reader (punch or printer).

Spooling Considerations During Shutdown and Abend Situations

CP does not automatically close user console spool files when VM/SP abends or the operator issues the SHUTDOWN command. The abend or SHUTDOWN condition causes console records to be lost and, consequently, an incorrect record count for spool files.

SPOOL

Responses

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

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

is received if (1) multiple copies of the file are being processed, (2) the file is being transferred to another user, or (3) the file is placed in a user hold status. However, if SET MSG OFF is in effect, this message is not issued, even though the above conditions are met.

STORE

Privilege Class: G

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

- Virtual storage locations (second level virtual storage only; see "Usage 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 | <table style="border: none; border-collapse: collapse;"> <tr> <td style="border: none; padding: 2px 5px;">hexloc</td> <td style="border: none; padding: 2px 5px;">Lhexloc</td> <td style="border: none; padding: 2px 5px;">hexword1 [hexword2...]</td> </tr> <tr> <td style="border: none; padding: 2px 5px;">Shexloc</td> <td style="border: none; padding: 2px 5px;"></td> <td style="border: none; padding: 2px 5px;">hexdata...</td> </tr> <tr> <td style="border: none; padding: 2px 5px;">{Greg}</td> <td style="border: none; padding: 2px 5px;">{Xreg}</td> <td style="border: none; padding: 2px 5px;">hexword1 [hexword2...]</td> </tr> <tr> <td style="border: none; padding: 2px 5px;">{Yreg}</td> <td style="border: none; padding: 2px 5px;"></td> <td style="border: none; padding: 2px 5px;">hexdword1 [hexdword2...]</td> </tr> <tr> <td style="border: none; padding: 2px 5px;">Psw</td> <td style="border: none; padding: 2px 5px;"></td> <td style="border: none; padding: 2px 5px;">[hexword1] hexword2</td> </tr> <tr> <td style="border: none; padding: 2px 5px;">STATUS</td> <td style="border: none; padding: 2px 5px;"></td> <td style="border: none; padding: 2px 5px;"></td> </tr> </table> | hexloc | Lhexloc | hexword1 [hexword2...] | Shexloc | | hexdata... | {Greg} | {Xreg} | hexword1 [hexword2...] | {Yreg} | | hexdword1 [hexdword2...] | Psw | | [hexword1] hexword2 | STATUS | | |
| hexloc | Lhexloc | hexword1 [hexword2...] | | | | | | | | | | | | | | | | | |
| Shexloc | | hexdata... | | | | | | | | | | | | | | | | | |
| {Greg} | {Xreg} | hexword1 [hexword2...] | | | | | | | | | | | | | | | | | |
| {Yreg} | | hexdword1 [hexdword2...] | | | | | | | | | | | | | | | | | |
| Psw | | [hexword1] hexword2 | | | | | | | | | | | | | | | | | |
| STATUS | | | | | | | | | | | | | | | | | | | |

where:

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

If hexloc is not on a fullword boundary, it is rounded down to the next lower fullword. The operands hexword1 [hexword2...]operand each represents up to 16 hexadecimal digits. If the value being stored is less than a fullword (eight hexadecimal digits), it is right-adjusted in the word and the high order bytes of the word are filled with zeros. If two or more hexwords are specified, they must be separated by one or more blanks.

Shexloc hexdata...
stores the data specified (hexdata...) in the address specified by hexloc, without word alignment. The shortest string that can be stored is one byte (two hexadecimal digits). If the string contains an odd number of characters, the last character is not stored, an error message is sent, and the function is terminated. The hexdata...operand is a string of two or more hexadecimal digits with no embedded blanks.

STORE

Greg hexword1 [hexword2...]

stores the hexadecimal data (hexword1 [hexword2...]) in successive general registers starting at the register specified by reg. The reg operand must be either a decimal number from 0 through 15 or a hexadecimal digit from 0 through F. The hexword1 [hexword2...] operand each represents up to eight hexadecimal digits. If the value being stored is less than a fullword (eight hexadecimal digits), it is right-adjusted in the word and the high order bytes of the word are filled with zeros. If two or more hexwords are specified, they must be separated by one or more blanks.

Xreg hexword1 [hexword2...]

stores the hexadecimal data (hexword1 [hexword2...]) in successive control registers starting at the register specified by reg. The reg operand must either be a decimal number from 0 through 15 or a hexadecimal digit from 0 through F. If the virtual machine is in basic control mode, you can store data in register 0 only. The hexword1 [hexword2...] operand each represents up to eight hexadecimal digits. If the value being stored is less than a fullword (eight hexadecimal digits), it is right-adjusted in the word and the high order bytes of the word are filled with zeros. If two or more hexwords are specified, they must be separated by one or more blanks.

Yreg hexdword1 [hexdword2...]

stores the hexadecimal data (hexdword1 [hexdword2...]) in successive floating-point registers starting at the register specified by reg. The reg operand must be a digit from 0 through 7. If reg is an odd number, it is adjusted to the preceding even number. The hexdword1 [hexdword2...] operand each represents up to 16 hexadecimal digits. If the value being stored is less than a doubleword (16 hexadecimal digits), it is left-justified in the doubleword and low order positions are filled with zeros. If two more more hexdwords are specified, they must be separated by one or more blanks.

PSW [hexword1] hexword2

stores the hexadecimal data in the first and second words of the virtual machine's program status word (PSW). If only hexword2 is specified, it is stored into the second word of the PSW. The [hexword1] hexword2 operand each represents up to eight hexadecimal digits. These operands must be separated by one or more blanks. If the value being stored is less than a fullword (eight hexadecimal digits), it is right-adjusted in the word and the high order bytes of the word are filled with zeros.

STATUS

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

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

Response

STORE COMPLETE

is the response at the successful completion of the command.

SYSTEMPrivilege Class: G

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

| | | | |
|--------|---|---------|---|
| SYSTEM | { | CLEAR | } |
| | | RESET | |
| | | RESTART | } |

where:

CLEAR clears virtual storage and virtual storage keys to binary zeros.

RESET clears all pending interrupts and conditions in the virtual machine.

RESTART simulates the hardware system RESTART function by storing the current PSW at virtual location eight and loading, as the new PSW, the doubleword from virtual location zero. Interrupt conditions and storage remain unaffected.

Usage Notes

1. The RESET function and the CLEAR function leave the virtual machine in a disabled wait state. Both CLEAR and RESET clear pending interrupts, both functions load a PSW of all zeros with only the WAIT bit set on (X'0002000000000000').
2. After issuing the SYSTEM command with RESET or CLEAR specified, either STORE a PSW and issue BEGIN or issue BEGIN with a hexadecimal storage location specified, to resume operation. The virtual machine automatically restarts at the location specified in the new PSW (which is loaded from the doubleword at location zero) after the SYSTEM RESTART command is processed.

Responses

STORAGE CLEARED - SYSTEM RESET
is the response given if the command SYSTEM CLEAR is entered.

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

If the command SYSTEM RESTART is entered, no response is given; the virtual machine resumes execution at the address in the virtual PSW loaded from virtual storage location X'0'.

TAG

Privilege Class: G

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

| | | | | |
|-------|---------|-----------|---------|-----------|
| TAG | { | Printer | } | [tagtext] |
| DEV | { | PUnch | } | [tagtext] |
| | { | CONsole | } | [tagtext] |
| | { | 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.

Note: when a generic device name, such as PRINTER or CONSOLE is used in the TAG command, all current virtual devices of that type are affected.

FILE spoolid

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

The spoolid operand is the spool file identification, a number between 1 and 9900 and assigned by CP when the spool file was closed.

tagtext is the information (up to 134 characters in length, including imbedded blanks) that is to be associated with the specified spool device or spool file. The contents and format of this data is completely flexible and is the responsibility of the file originator and the end user.

Certain control and addressing information, meaningful to the RSCS component of VM/370, or the RSCS Networking program product (5748-XP1), can be specified in this field. For details on how to use the tagtext operand when transmitting files across the RSCS network, refer to the following "Usage Notes" section.

| | | |
|-------|---------|-------------|
| QUERY | DEV | (PRINTER) |
| | | PRT |
| FILE | spoolid | (PUNCH) |
| | | PCH |
| | | (CONSOLE) |
| | | vaddr |

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

Usage Notes

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

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

where:

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

userid is the userid of the VM/SP virtual machine (a one- to eight-character user identification) to which a file is being transmitted. This operand is used by remote stations when they transmit files to a virtual machine in a remote spooling network and want the files sent to a particular VM/SP virtual machine. You can ignore this operand if you are not specifying a priority. However, if you are specifying a priority, you must code some userid operand; it is ignored in a remote spooling network.

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

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

4. Although the buffer that contains TAG information is 134 characters in length including imbedded blanks, only 126 characters of it are usable because of input command line limitations.

Responses by Command Operands

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

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

```

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

```

TAG QUERY DEV vaddr

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

```

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

```

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

TAG QUERY FILE spoolid

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

```

tagtext.....
           if tag data exists

```

(TAG BLANK)
 if the tag is all blanks

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

TERMINAL

TERMINAL

Privilege Class: G

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

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

The format of the TERMINAL command is:

| | | | |
|----------|---|----------|--------------|
| TERminal | { | CHardel | { ON } |
| | | LINEDel | { OFF } |
| | | LINeNd | { char } |
| | | EScape | |
| | | TABchar | |
| | | APL | { ON } |
| | | TEXT | { OFF } |
| | | ATtn | |
| | | HILight | |
| | | SCRNsave | |
| | | MODE | { CP } |
| | | | { VM } |
| | | LINESize | nnn |
| | | CONmode | { 3215 } |
| | | | { 3270 } |
| | | BREakin | { IMmed } |
| | | | { GUESTctl } |
| | | BRKkey | { PA 1 } |
| | | | { PF 1 } |
| | | | { . } |
| | | | { PF24 } |

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

TERMINAL CHARDEL OFF LINESIZE 20 MODE VM

where:

CHARDEL { ON }
 { OFF }
 (char)

defines the logical character delete symbol. If ON is specified, the default symbol becomes the logical character delete symbol. The default symbol is normally @, but depends on what is specified in your VM/SP directory entry. If OFF is specified, no logical character delete symbol is allowed. If char is specified, then that character becomes a logical character delete symbol. The character chosen should be unique and not common to the data stream being entered. When you log on, CHARDEL ON is in effect; if your virtual machine is logged on by the system operator via the AUTOLOG command, CHARDEL OFF is the default.

LINEDEL { ON }
 { OFF }
 (char)

defines the logical line delete symbol. If ON is specified, the default symbol becomes the logical line delete symbol. The default symbol is normally #, but depends on what is specified in your VM/SP directory entry. If OFF is specified, no logical line delete symbol is allowed. If char is specified, then that character becomes the logical line delete symbol. The character chosen should be unique and not common to the data stream being entered. When you log on, LINEDEL ON is in effect; if your virtual machine is logged on by the system operator via the AUTOLOG command, LINEDEL OFF is the default.

LINEND { ON }
 { OFF }
 (char)

defines the logical line end symbol. If ON is specified, the default symbol becomes the logical line end symbol. The default symbol is normally #, but depends on what is specified in your VM/SP directory entry. If OFF is specified, no logical line end symbol is allowed. If char is specified, then that character becomes the logical line end character. The character chosen should be unique and not common to the data stream being entered. When you log on, LINEND ON is in effect; if your virtual machine is logged on by the system operator via the AUTOLOG command, LINEND OFF is the default.

ESCAPE { ON }
 { OFF }
 (char)

defines the logical escape character. If ON is specified, the default symbol becomes the logical escape character. The default symbol is normally ", but depends on what is specified in your VM/SP directory entry. If OFF is specified, no logical escape character is allowed. If char is specified, then that character becomes the logical escape character. The character chosen should be unique and not common to the data stream being entered. When you log on, ESCAPE ON is in effect; if your virtual machine is logged on by the system operator via the AUTOLOG command, ESCAPE OFF is the default.

TERMINAL

TABCHAR { ON }
 { OFF }
 { char }

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

ON sets the logical tab character to the one used by the system.

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

char defines the logical tab character to be used by the system internally. It may be any valid special graphic character. When an invalid character is specified, an error message is issued to the terminal and the logical tab character remains unchanged.

APL { ON }
 { OFF }

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

Notes:

1. This operand is valid for certain 3270 Information Display Stations. See VM/SP Planning and System Generation Guide for applicable APL terminals.
2. If the TERMINAL APL ON command is issued from a terminal that is not equipped with APL keys, or if the user's virtual machine is not running APL when the TERMINAL command is issued, the results may be unpredictable.

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

TEXT { ON }
 { OFF }

controls the use of Text translation tables. If TEXT ON is specified, CP uses the translation tables applicable to display terminals equipped with the appropriate Text hardware features. If TEXT OFF is specified, CP uses the normal translation tables. Unless otherwise specified, TEXT OFF is in effect. When TEXT ON is specified, APL OFF is forced.

Notes:

1. This operand is valid for certain 3270 Information Display Stations. See VM/SP Planning and System Generation Guide for applicable TEXT terminals.
2. The setting of the SET LINEDIT command is independent of the TERMINAL TEXT setting.

ATTN {ON }
 {OFF }

controls signaling of an attention interrupt. If ATTN ON is specified, the exclamation point is displayed when an attention interrupt occurs. The OFF option suppresses the displaying of the exclamation point (!) and 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.

HIGHLIGHT {ON }
 {OFF }

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

SCRNSAVE {OFF }
 {ON }

allows you, as the console operator, to choose between having the guest screen saved when going into CP mode, or not having the guest screen saved. This option is only valid if you have a local 3270 Information Display device and you have issued the CONMODE 3270 operand of the TERMINAL command. If you specify ON with this option, the screen is saved by CP and redisplayed when the guest gets control again without an ATTN interrupt. OFF specifies that CP does not save the guest screen (that is, a CLEAR interrupt is posted by CP to the guest when CP gives control back to the guest). OFF is the default.

MODE {CP }
 {VM }

controls the terminal attention environment. CP specifies that one or more attentions force the virtual machine into the CP environment. VM specifies that one attention is reflected to your virtual machine and that more than one attention forces your virtual machine into the CP environment. VM is the default for all VM/SP users except the primary system operator. For more information on this, see "Controlling the Virtual Machine" in Section 3.

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.

TERMINAL

CONMODE {3215}
{3270}

allows you to choose between 3215 and 3270 support at an SIO level. The initial definition is determined by having the console mode in the CONSOLE statement in your directory. You must have a local display device that has a 3270 compatible command set. If you specify 3215, then 3215 CCW command code processing occurs at an SIO level from the guest. If you specify 3270, then 3270 CCW command code processing occurs at an SIO level from the guest. virtual machine. The default setting is 3215. You must set CONMODE to 3215 to run CMS. CMS sets CONMODE to 3215 at IPL time, but if you dynamically reset CONMODE to 3270 while in CMS mode, results can be unpredictable.

BREAKIN {IMMED}
{GUESTCTL}

allows you to control break in by CP. This option is available only to users with local 3270 Information Display devices. The GUESTCTL option specifies that messages from CP are only displayed when you hit the BRKKEY key. If you specify GUESTCTL, CP break in still occurs if you request a CP function or if a priority message is pending for your virtual machine. If you specify IMMED, the default, break in by CP takes place on any full screen READ and full screen WRITES that have an inactive keyboard; they also take place when you hit the BRKKEY.

BRKKEY (PA1)
(PF1)
(.)
(.)
(PF24)

allows the PA1 key or a PF key you specify to control break in by CP for full screen guests. PA1 is the default. You must be sure that the PF key associated with this function is available on the guest virtual machine console. Choosing a PF key as the break in controller that does not exist on the guest virtual machine console hardware can lock out CP mode.

Usage Notes

1. The terminal settings you specify with the TERMINAL command are in effect only for the duration of that terminal session. Whenever you initially log on, the system defaults are in effect. However, the settings you specify for line-editing (except LINEDEL) and MODE are still in effect when you log on after disconnecting. All the other operands (TEXT, APL, ATTN, HILIGHT, and LINESIZE) are reset if you log on after disconnecting.
2. Although you can define line-editing symbols and status with the TERMINAL command, the LINEDIT operand of the SET command determines whether the VM/SP line-editing functions are on or off.
3. If an error occurs during processing of the command, all functions preceding the one with the error are in effect.
4. You cannot use any of the letters A through Z, or the numbers 0 through 9, as a symbol (char) in association with any of the CHARDEL, LINEDEL, LINEND, and ESCAPE operands.

- Figure 10 describes the other features available for highlighting because the TERMINAL command cannot activate or deactivate these features.

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

Figure 10. 3270 High Intensity Features

- CP automatically masks your password on typewriter terminals. For installations migrating from a VM/370 operating system to VM/SP, and for compatibility purposes, CP will accept the MASK operand on the TERMINAL command as valid. However, there is no need to specify this operand since CP automatically masks your password.
- If you set SCRNSAVE ON and CONMODE 3215 is active for your terminal, CP accepts the TERMINAL SCRNSAVE ON command; however, a screen save does not occur.
- If your virtual machine is not accepting interrupts and SCRNSAVE is set OFF, you must hit the BRKKEY twice in order to enter CP mode.

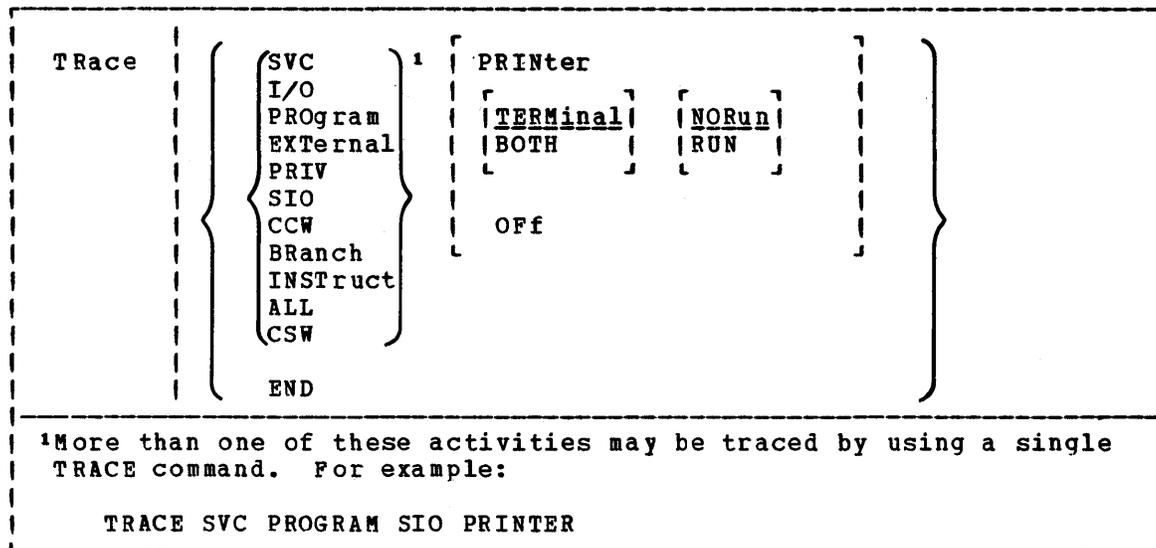
Responses

None.

TRACE

Privilege Class: G

Use the TRACE command to trace specified virtual machine activity and to record the results at the terminal, on a virtual spooled printer, or on both terminal and printer. If you issue more than one TRACE command, the operands are cumulative; that is, operands specified for the first time are activated, whereas those specified with new modifiers are updated. The RUN and NORUN operands, however, can be specified in different TRACE commands for different tracing functions and do not cause a conflict. The format of the TRACE command is:

where:

- SVC traces virtual machine SVC interrupts.
- I/O traces virtual machine I/O interrupts.
- PROGRAM traces virtual machine program interrupts.
- EXTERNAL traces virtual machine external interrupts.
- PRIV traces all virtual machine non-I/O privileged instructions.
- SIO traces TIO, CLRIO, HIO, HDV, and TCH instructions 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 interrupts, PSW instructions, and successful branches.
- INSTRUCT traces all instructions, virtual machine interrupts, and successful branches.

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

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

END terminates all tracing activity and prints a termination message.

PRINTER directs tracing output to a virtual spooled printer. If PRINTER is specified, RUN is assumed; however neither RUN nor NORUN can be specified.

TERMINAL directs tracing output to the terminal (virtual machine console).

BOTH directs tracing output to both a virtual spooled printer and the terminal.

OFF halts tracing of the specified activities on both the printer and terminal.

NORUN stops program execution after the trace output to the terminal and enters the CP command environment.

Note: If a Diagnose code X'008' is being traced, NORUN has no effect and program execution does not stop.

RUN continues the program execution after the trace output to the terminal has completed and does not enter the CP command environment.

Usage Notes

1. If your virtual machine has the virtual=real option and NOTRANS set on, CP forces CCW translation while tracing either SIO or CCW. When tracing is terminated with the TRACE END command, CCW translation is bypassed again.
2. If VM/SP hardware assist is enabled on your virtual machine, CP turns it off while tracing SVC, PRIV, BRANCH, INSTRUCT, or ALL activities. If virtual interval timer support is available on your processor and is active (that is, SET ASSIST TMR is in effect), CP turns the timer support off while tracing external interrupts. After the tracing is terminated with the TRACE END command, CP turns hardware assist on again.
3. If trace output is being recorded at the terminal, the virtual machine stops execution and CP command mode is entered after each output message. This simulates the instruction step function.

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

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

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

TRACE

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

4. If trace output is being recorded on a virtual spooled printer, a CLOSE command must be issued to that printer in order for the trace output to be printed on the real printer.
5. Successful branches to the next sequential instruction and branch-to-self instructions are not detected by TRACE.
6. RUN and NORUN can be specified in different TRACE commands with no conflict as long as the functions being traced are different.
7. Instructions that modify or examine the first two bytes of the next sequential instruction cause erroneous processing for BRANCH and INSTRUCT tracing.
8. When tracing on a virtual machine with only one printer, the trace data is intermixed with other data sent to the virtual printer. To separate trace information from other data, define another printer with a lower virtual address than the previously defined printer. For example, on a system with 00E defined as the only printer, define a second printer as 00B. The regular output goes to 00E and the trace output goes to 00B.
9. If the BRANCH, INSTRUCT, or ALL activities are being traced by a virtual machine using a shared system, the user receives a warning message and the virtual machine is unshared from (given a private copy of) the named system. Other users of the segment continue to operate with an unchanged copy of that segment.
10. I/O operations for virtual channel-to-channel adapters, with both ends connected to the same virtual machine, cannot be traced.
11. Use the TRACE command to trace specified virtual machine activity and to record the results at the terminal, at a virtual printer, or at both. This command is useful in debugging programs because it allows you to trace only the information that pertains to a particular problem.
12. If your virtual machine is doing I/O that results in program-controlled interruptions (PCIs), and you are tracing I/O or CSW activity, some of the PCIs may not be traced. This situation arises when the system is extending its free storage area and the additional demand on available free storage can cause a system abend.
13. The PA1 key on a 3270 display terminal should not be pressed with trace all run active while the display screen indicates RUNNING status as a subsequent CP command entered from the console may result in interleaved output.

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 |
| SEEK | read/write position on count-key-data device |
| REAL EXTENT | real block location on a fixed-block device |
| VIRT EXTENT | virtual block location on a fixed-block device |
| LOCATE | read/write block position within extent |

TRACE STARTED

This response is issued when tracing is initiated.

TRACE ENDED

This response is issued when tracing is suspended.

TCH, TIO, CLRIO, HIO, HDV, SIO, or SIOF

TCH

I/O vvvvvv TCH xxxxxxxx type vadd CC n

TIO, CLRIO, HIO, or HDV

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

SIO or SIOF

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

TRACE

CCW:

```
CCW vvvvvv xxxxxxxx xxxxxxxx rrrrrr yyyyyyyy yyyyyyyy
CCW IDAL vvvvvvvv vvvvvvvv IDAL 00rrrrrr 00rrrrrr
CCW SEEK xxxxxxxx xxxxxxxx SEEK yyyyyyyy yyy
CCW REAL EXTENT yyyyyyyy yyyyyyyy yyyyyyyy yyyyyyyy
CCW VIRT EXTENT xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx
CCW LOCATE xxxxxxxx xxxxxxxx
```

The IDAL or SEEK lines, as well as the REAL EXTENT, VIRT EXTENT, or LOCATE lines, are included only if applicable. The virtual IDAL is not printed if the real CCW operation code does not match the real CCW.

INSTRUCTION TRACING:

Privileged Instruction:

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

Executed Instructions:

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

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

All Other Instructions:

```
vvvvvv mnem xxxxxxxx xxxx
```

SUCCESSFUL BRANCH:

```
vvvvvv mnem xxxxxxxx ==> tttttt
```

INTERRUPTION (SVC, PROGRAM, or EXTERNAL)

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

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

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

BRANCH TRACE: (ALL option selected)

Entry for 'branch from' instruction

```
vvvvvv mnem xxxxxxxx tttttt
```

Entry for 'branch to' instruction

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

TRANSFER

Privilege Class: G

Use the TRANSFER command to transfer your closed spool files to a specified user or queue, or to reclaim closed spool files that you created.

The format of the class G TRANSFER command is:

| | | | | |
|----------|----------------------------------|--|------------------------|----------------------------------|
| TRANSfer | [Printer] [PUnch] [ReadeR] | {spoolid CLASS c FORM form ALL} | {[To] {* userid}} | [Printer] [PUnch] [ReadeR] |
| | | | {From {ALL userid}} | |

where:

PRINTER/PRT

PUNCH/PCH

READER/RDR

specifies the queues to which and from CP transfers files.

spoolid is the spool file you want to transfer to the queue of the userid you specify or retrieve from the queue of the userid specified.

CLASS c is the class of spool files you want to transfer. The c is a one-character alphameric field with values from A to Z and from 0 to 9.

FORM form

specifies that you want to transfer all files that have the form name you specify. 'form' is a one to eight character form name defined at system generation.

ALL specifies that you want to transfer all spool files in a reader/printer/punch queue.

[TO] {userid}
{*}

is the user to whom you want to transfer the files. If you omit the optional keyword TO, the userid may not be TO or T. '*' denotes your own userid.

FROM {userid}
{ALL}

is the userid from whom you want to reclaim spool files of the specified reader/printer or punch queue. You can specify ALL to reclaim spool files you originally created and transferred to the reader/printer/punch queues of other users.

TRANSFER

Usage Note:

The TRANSFER command does not transfer any active spool files.

The command only transfers spool files to valid queues. For example, files created as real card reader input decks, CP system dumps, user dumps (VMDUMP command), network dumps, or VM/SP Monitor spool files can exist only on the reader queue. Files created on a virtual punch, or by accounting to disk support, can exist on either the reader queue or the punch queue. Files created on a virtual printer can exist on either the reader queue or the print queue. The following chart summarizes the valid queues to which you can transfer spool files.

| | Spool Files | Queue | | |
|---|--------------|-------|-----|-----|
| | | RDR | PRT | PCH |
| O | Real Reader | * | | |
| r | Printer | * | * | |
| i | Console | * | * | * |
| n | Punch | * | | * |
| | Dump/Monitor | * | | |

"*" = files may reside on queue

CP flags, as an error, any attempt to transfer spool files from a print queue to punch queue, as well as any attempt to transfer spool files from a punch queue to a print queue. You may, however, transfer files on the reader queue to printer or punch queues, if they are print files and punch files respectively.

Examples:

TRANSFER 0123 TO USERA

transfers the spool file with spoolid 0123 from your reader queue to the reader queue of USERA.

TRANSFER PRT 0123 TO USERA

transfers the spool file with spoolid 0123 from your print queue to the reader queue of USERA.

TRANSFER PRT ALL TO * RDR

transfers all spool files on your print queue to your reader queue. Note that you didn't need to specify the default, RDR.

TRANSFER RDR CLASS B TO * PCH

transfers all the class B punch files on your reader queue to your own punch queue. CP does not transfer any class B files on your reader queue that are not punch files.

TRANSFER RDR ALL FROM ALL PRT

reclaims all print files that you created and sent to the print queues of other users. CP places these reclaimed files on your reader queue.

Responses

{ RDR } FILE spoolid TRANSFERRED TO userid { RDR }
 { PRT }
 { PUN }

is the response when you issue a TRANSFER command with the TO option. CP also sends this response to the previous owner of a file reclaimed with the FROM option.

where:

RDR/PRT/PUN
 (first occurrence) is the queue on which the file previously resided.

RDR/PRT/PUN
 (second occurrence) is the queue on which the file now resides. CP does not display the second occurrence of RDR/PRT/PUN unless the old and new queues are different.

spoolid is the spool identification number of the file. This does not change.

userid is the user identification of the new file owner.

{ RDR } FILE spoolid TRANSFERRED FROM {userid} { RDR }
 { PRT } { * } { PRT }
 { PUN }

is the response to the user issuing the TRANSFER command with the FROM option. This response is also sent to the new owner of a file transferred with the TO option.

where:

RDR/PRT/PUN
 (first occurrence) is the queue on which the file now resides.

RDR/PRT/PUN
 (second occurrence) is the queue on which the file previously resided. CP does not display the second occurrence of RDR/PRT/PUN unless the old and new queues differ.

spoolid is the spool identification number of the file. This does not change.

userid is the identification of the previous file owner.

* is displayed when the "TO *" option is used.

{nnnn} FILES TRANSFERRED
 {NO }

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

CP does not display any of these responses if you issued the CP SET IMSG OFF command.

VMDUMP

Privilege Class: G

The VMDUMP command, when used with the VM/Interactive Problem Control System Extension (VM/IPCS Extension) program product, dumps storage for guest virtual machines (that is, the virtual storage that VM/SP creates for the virtual machine user). See Usage Note 1. The VM/IPCS Extension debugs errors, as well as stores and maintains error information about the virtual machine. The VM/IPCS Extension displays this information at the user's terminal. VMDUMP provides the VM/IPCS Extension with header information to identify the owner of the dump; it also maintains dump information, writes the dump to a spool file, and formats the dump. In addition to saving the storage contents of the virtual machine, the VMDUMP command also dumps:

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

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

The VMDUMP command, when issued without the VM/IPCS Extension installed, only writes an unformatted dump to a spool file. It does not format the dump without the VM/IPCS Extension. However, you may write routines that perform this function. For more information on the VM/IPCS Extension, refer to VM/SP Interactive Problem Control System Extension (VM/IPCS Extension) Program Product: User's Guide and Reference, Order No. SC34-2019.

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

The format of the VMDUMP command is:

```

VMDUMP  [hexloc1]  [{-}  [hexloc2]
         0        {:]  [ END ]
         [         [   [
         [         [   [bytecount]
         [         [   [ END ]
         [         [   ]
         [         [
         [ TO *    ]
         [ TO userid]
         [ SYSTEM  ]
         [         ]
         [ FORMAT vmtype ]
         [ DSS ]
         [ *dumpid ]

```

where:

hexloc1
0

is the first or only hexadecimal virtual storage address dumped. If you omit the hexloc1 operand, the default is zero, the beginning of virtual machine storage. You may also specify the hexloc2 or bytecount operands with the hexloc1 operand when you know the dump size you want. If you specify hexloc2 with hexloc1, CP takes a dump of the storage contents within the range of addresses between hexloc1 and hexloc2. For example, when you enter:

```

vmdump 1000-2000
      or
vmdump 1000:2000

```

CP dumps the contents of virtual machine storage at the hexadecimal addresses between 1000 and 2000. If you specify the bytecount operand with hexloc1, hexloc1 specifies the beginning address of the dump and the hexadecimal number of bytes dumped. If you enter:

```

vmdump 1000.8A

```

CP dumps the contents of virtual machine storage beginning at hexadecimal address 1000 for a total of 8A bytes. You can specify a maximum of three different areas of virtual storage that you want dumped each time you issue the VMDUMP command. See Usage Notes 2, 3, and 5.

VMDUMP

{-}
{:} hexloc2
END

is the last hexadecimal virtual storage address dumped. If you do not specify the hexloc2 operand, the default is END, and CP dumps the contents of virtual machine storage starting from hexloc1 to the end of virtual storage. You must precede the hexloc2 operand with one of the allowable delimiters (either a dash or a colon) to dump the contents of more than one location by storage address. For example, if you enter:

```
vmdump :2000
      or
vmdump -2000
```

CP takes a dump of the contents of virtual storage from hexadecimal location zero (the default for hexloc1) to hexadecimal address 2000. See Usage Notes 1 and 4 for an expanded explanation of the use of delimiters. You may specify up to a maximum of three different virtual machine storage areas you wish to dump each time you issue the VMDUMP command. You must specify a value for hexloc2 which is greater than hexloc1 value. The hexloc2 value must not exceed the limit of virtual storage.

{.}bytecount
END

is the hexadecimal number of bytes dumped, starting with the first byte of virtual storage at hexloc1. If you do not specify bytecount, the default is END, and CP takes a dump of the contents of virtual machine storage from the first byte at hexloc1 to the end of virtual storage. You must specify a period (.) delimiter if you want to dump the contents of more than one storage location by bytecount. When you enter:

```
vmdump 1000.20
```

CP takes a dump of the virtual storage contents from hexadecimal location 1000 for a length of X'20' bytes. See Usage Note 3 for an expanded explanation of delimiter use.

TO userid
TO *

transfers the dump to the virtual card reader of the userid specified. If you enter an asterisk after TO, CP transfers the dump to your virtual card reader.

SYSTEM

transfers the dump to the virtual card reader of the userid specified on the SYSDUMP operand of the SYSOPR system generation macro instruction. You must not specify TO preceding the keyword SYSTEM.

FORMAT vmtyp

provides VM/IPCS Extension with the virtual machine type (vmtyp) which VM/IPCS Extension uses to format the dump. The vmtyp keyword is a one-to-eight byte name of the operating system running in a virtual machine (for example, CMS). CP also uses the specified vmtyp as the virtual card reader filetype. CP does not validity check the vmtyp. Any vmtyp longer than eight bytes generates an error message and halts further VMDUMP processing. The dump header record includes you specify vmtyp and the VM/IPCS Extension uses the vmtyp information to format the dump. If you enter FORMAT, you must also specify a vmtyp. If you do not specify FORMAT, the default vmtyp is FILE. See Usage Note 4.

DSS

specifies that CP take a dump of all discontinuous saved segments in use by your virtual machine.

*dumpid is a line of user input up to 100 characters long including imbedded blanks and asterisks which you can enter for your own benefit (that is, for descriptive purposes, such as the time and date of the dump, or what was being processed at the time of the dump). If you specify this operand, it becomes the DMPDMPID field in the dump file information record (DMPINREC) data area. If specified, you must enter *dumpid as the last operand on the VMDUMP command line.

Usage Notes:

1. First level storage is real main storage. CP, however, dumps only second level storage (that is, storage that VM/SP creates for the guest virtual machine.) Operating systems running in a guest virtual machine such as OS/VS, DOS/VS VSE/AF Rel 1, and VSE/AF Rel 2, have virtual (third level) storage of their own. CP cannot dump this third level storage directly. The user or the virtual operating system is responsible for converting any third level storage addresses to second level storage addresses before issuing the VMDUMP command.
2. The dump profile produced contains a whole number of 4K pages. CP rounds down the hexloc1 operand to a 4K boundary if its value is not a multiple of 4K. CP rounds up the hexloc2 operand to a 4K boundary minus one.
3. Due to 4K boundary rounding for the hexloc1 operand, additional dumps of up to 4095 more bytes than you specified can occur.
4. If you enter the FORMAT operand followed by another valid keyword of the VMDUMP command (for example, SYSTEM), CP uses the keyword following FORMAT as the vmtyp and CP ignores the normal meaning of the keyword following FORMAT.
5. If you omit a dash or a colon between the hexloc1 and hexloc2 operands, and you separate the operands with one or more blanks, CP dumps only the storage contents at those two specific hexadecimal addresses. You must insert blanks to separate operands or sets of operands if you specify more than one operand on the command line. You must not insert blanks to the left or right of the range or length delimiters (-, :, or .) unless you intend to use the default value of the missing operand that the blank represents.

Responses

COMMAND COMPLETE

is the response indicating normal completion of the VMDUMP function.

Appendix: CP Command Summary

This appendix contains brief descriptions of all the commands acceptable to the CP component of the VM/SP system.

Figure 11 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. ACNT also closes the spool file that is accumulating accounting records. |
| 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. |

Figure 11. CP Command Summary (Part 1 of 5)

| Command | Privilege Class | Usage |
|----------|-----------------|--|
| DEFINE | B | Redefine the usage of SYSVIRT and VIRTUAL 3330V devices. |
| | G | Reconfigure your virtual machine. |
| DETACH | B | Disconnect a real device from a virtual machine. |
| | B | Detach a DASD device from CP. |
| | B | Detach a channel from a specific user. |
| | G | Detach a virtual device from a virtual machine. |
| | G | 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, 370X 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 | A,E,G | Indicate resource utilization and contention. |
| IPL | G | Simulate IPL for a virtual machine. |
| LINK | G | Provide access to a specific DASD by a virtual machine. |

Figure 11. CP Command Summary (Part 2 of 5)

| Command | Privilege Class | Usage |
|----------|-------------------|---|
| LOADBUF | D | Load real UCS/UCSB or FCB printer buffers. |
| LOADVFCB | G | Load virtual forms control buffer for a virtual 3203, 3262, 3289E, or 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. |
| MIGRATE | A | Allows the operator to migrate pages either for the entire system or just one user. |
| MONITOR | A,E | Trace events of the real machine and record system performance data. |
| MSGNOH | B | Send a specified message, without the standard message header, from one virtual machine to another. |
| NETWORK | A,B,F | Load, dump, trace, and control the operation of the 370X 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. |
| QVM | A | Request the transition from VM/SP to the V=R virtual machine running in native mode. |
| 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. |

Figure 11. CP Command Summary (Part 3 of 5)

| Command | Privilege Class | Usage |
|----------|-----------------|---|
| REWIND | G | Rewind (to load point) a tape and ready a tape unit. |
| SCREEN | G | Allows the user to change or alter the color and extended highlighting values for his virtual console. |
| SEND | G | Pass commands and message replies to disconnected virtual machine for processing. |
| SAVESYS | E | Save virtual machine storage contents, registers, and PSW. |
| SET | A,B,E, F,G | Operator--establish system parameters. User--control various functions within the virtual machine. |
| SHUTDOWN | A | Terminate all VM/SP functions and checkpoint CP system for warm start. |
| SLEEP | any | Place virtual machine in dormant state. |
| SMSG | G | Send Special Messages to specified virtual machine. |
| SPACE | D | Force single spacing on printer. |
| SPMODE | A | Establish or reset the single processor mode environment. |
| SPOOL | G | Alter spooling control options; direct a file to another virtual machine or to a remote location via the RSCS virtual machine. |
| SPTAPE | D | Dump output spool files on tape or load output spool files from tape. |
| 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. |

Figure 11. CP Command Summary (Part 4 of 5)

| Command | Privilege Class | Usage |
|----------|-----------------|--|
| 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. |
| VMDUMP | G | Dump virtual machine when issued with the VM/IPCS Extension. |
| WARNING | A,B | Transmit a high priority message to a specified user or to all users. |

Figure 11. CP Command Summary (Part 5 of 5)

Index

- *
 - command
 - described 40
 - summary 189
 - * operand
 - of MESSAGE command 102
 - of SPOOL command 153
 - *dumpid operand, VMDUMP command 185

- #CP
 - command
 - commandline operand 41
 - described 41
 - summary 189
 - #CP command, example of usage 42

- 'color' operand, of SCREEN command 134
- 'exthilght' operand, of SCREEN command 134

- A
 - abbreviation, use of 35
 - accounting, information, displaying 138
 - ACNT, command, summary 189
 - ACNT operand, of SET command 138
 - ADSTOP
 - command
 - described 43
 - hexloc operand 43
 - OFF operand 43
 - summary 189
 - AFFINITY operand, of SET command 138
 - affinity setting, relation to the logon process 90
 - ALL operand
 - of CHANGE command 47
 - of PURGE command 107
 - of QUERY command 112
 - of TRACE command 176
 - of TRANSFER command 181
 - altering
 - attributes of
 - virtual printer file 47
 - virtual punch file 47
 - virtual reader file 47
 - logical editing symbols 170
 - virtual machine environment 58
 - annotating your console sheet 40
 - Any user, described 8
 - APL, control of translation tables 170
 - APL operand, of TERMINAL command 170
 - ASSIST operand, of SET command 138
 - ATTACH, command, summary 189

 - attached processor, INDICATE command use for monitor of system load 85
 - attached processor application, use of SET ASSIST 138
 - attention
 - handling, terminal control of 170
 - interrupt
 - entering from your virtual console 45
 - for a virtual machine 131
 - ATTN
 - command
 - described 45
 - summary 189
 - ATTN operand
 - of IPL command 88
 - of TERMINAL command 170
 - attributes
 - modifying, for 3800 printer 153
 - spool file, changing 47
 - virtual device, modifying 153
 - AUTOOCR option, of PARM operand, for IPL command 91
 - AUTOPOLL operand, of SET command 138

- B
 - BACKSPAC, command, summary 189
 - BATCH option, of PARM operand, for IPL command 91
 - BEGIN
 - command
 - described 46
 - hexloc operand 46
 - summary 189
 - BEGIN command, used with the DISCONN command 72
 - blanks, as delimiters 3
 - block address, used to load a system, at IPL 88
 - BOTH operand, of TRACE command 176
 - braces, use of 36
 - brackets, use of 36
 - BRANCH operand, of TRACE command 176
 - branches, tracing 176
 - BREAKIN operand, of TERMINAL command 170
 - BRKKEY operand, of TERMINAL command 170
 - byte alignment, DISPLAY command 28
 - bytecount operand, VMDUMP command 185

- C
 - CAW (Channel Address Word), displaying 74
 - CAW operand, of DISPLAY command 74
 - CCW operand, of TRACE command 176
 - CCW translation, controlling 138

CHANGE

- command
 - ALL operand 47
 - CHARS operand 47
 - CLASS operand 47
 - COPY* operand 47
 - COPY operand 47
 - described 47
 - DIST operand 47
 - FCB operand 47
 - FLASH operand 47
 - FORM operand 47
 - HOLD operand 47
 - MODIFY operand 47
 - NAME operand 47
 - NOHOLD operand 47
 - PRINTER operand 47
 - PUNCH operand 47
 - READER operand 47
 - summary 189
- changing
 - extended control registers in your virtual machine 163
 - filename of a spool file 47
 - floating-point registers in your virtual machine 163
 - general registers in your virtual machine 163
 - Program Status Word (PSW) in your virtual machine 163
 - spool class of a file 47
 - spool file attributes 47
 - spoolid of a file 47
 - virtual storage locations 163
- channel c operand, of DETACH CHANNEL command 69
- CHANNELS operand
 - of DEFINE command 58
 - of QUERY command 112
- channel-to-channel adapters, virtual, connecting 55
- character
 - set 4
 - translation tables, API, controlling 170
- CHARDEL operand, of TERMINAL command 170
- CHARS operand
 - of CHANGE command 47
 - of SPOOL command 153
- class
 - privilege, for CP commands 7
 - spool, changing for a file 47
 - virtual device spool, modifying 153
- class Any, described 9
- class G, described 9
- CLASS operand
 - of CHANGE command 47
 - of ORDER command 105
 - of PURGE command 107
 - of QUERY command 112
 - of SPOOL command 153
 - of TRANSFER command 181
- CLEAR operand
 - of IPL command 88
 - of SYSTEM command 165
- clearing
 - interrupts from a virtual machine 132
 - storage to zeros 88,165

CLOSE

- command
 - CONSOLE operand 51
 - described 51
 - DIST operand 51
 - dsname operand 51
 - fnft operand 51
 - FORM operand 51
 - HOLD operand 51
 - NAME operand 51
 - NOHOLD operand 51
 - PRINTER operand 51
 - PUNCH operand 51
 - PURGE operand 51
 - READER operand 51
 - summary 189
 - vaddr operand 51
 - HOLD restrictions 51
 - NOHOLD restrictions 51
 - restrictions 51
- CLOSE operand, of SPOOL command 153
- closing
 - files 51
 - TRACE output 32
 - virtual devices 153
- code operand, of EXTERNAL command 84
- command
 - name 4
 - operands 4
- command environment
 - CP 3
 - defined 3
- commandline operand
 - of #CP command 41
 - of CP command 57
- commands
 - ADSTOP command, halting execution of virtual machine 26
 - ATTACH command, with dedicated devices 20
 - ATTN command, simulates real console function 23
 - CHANGE command, altering spool file attributes 16
 - CLOSE command, using with spool files 16
 - CP, summarized 189
 - DEFINE command
 - reconfiguring virtual machine 25
 - temporary disks 14
 - DETACH CHANNEL command, with dedicated channels 21
 - DETACH command
 - removing dedicated devices 20
 - temporary disks 14
 - DISPLAY command, displaying virtual storage 28
 - DUMP command, printing virtual storage 29
 - EXTERNAL command, simulating interruptions 22
 - LINK command, sharing virtual disks 14
 - MESSAGE command, obtaining operator assistance 21
 - notational conventions 35
 - ORDER command, reordering spool files 19
 - PURGE command, deleting spool files 19

QUERY command, determining device or file attributes 16
 REQUEST command, simulates real console function 23
 SPOOL command, altering device attributes 16
 STORE command, altering virtual storage 30
 summary of use 189
 TRACE command, follow virtual machine activities 32
 used to control a terminal session, summary of 189
 comments, how to write 4
 CONMODE operand, of TERMINAL command 170
 connecting
 remote terminals to virtual machines 70
 virtual channel-to-channel adapters 55
 console
 input/output, terminating 153
 sheet, annotation of 40
 spool file, modifying 153
 spooling, controlling 153
 CONSOLE operand
 of CLOSE command 51
 of DEFINE command 58
 of QUERY command 112
 of SPOOL command 153
 CONT operand, of SPOOL command 153
 continuous reading of an entire file 153
 control, functions, setting for your virtual machine 138
 Control Program, console function mode 24
 controlling
 a terminal session, summary of commands used 189
 3270 display 172
 controlling virtual machine, during terminals session 11
 COPY* operand
 of CHANGE command 47
 of SPOOL command 153
 COPY operand
 of CHANGE command 47
 of SPOOL command 153
 COUPLE
 command
 described 55
 summary 189
 userid operand 55
 vaddr1 operand 55
 vaddr2 operand 55
 CP
 command
 commandline operand 57
 described 57
 summary 189
 CP command environment 3
 CP commands
 entering 23
 from CP console function mode 24
 from virtual console read mode 24
 while running 22
 entering from the virtual machine environment 57
 privilege classes, described 8
 privilege classes for 7
 text format of 39
 CP operand, of SEND command 137
 CPUID operand, of QUERY command 112
 creating disk space with DEFINE command, precautions 58
 CSW (Channel Status Word), displaying 74
 CSW operand
 of DISPLAY command 74
 of TRACE command 176
 CTCA operand, of DEFINE command 58

 D
 DASD operand, of QUERY command 112
 DCP, command, summary 190
 debugging, programs 26
 dedicated channel, detaching 69
 dedicated 2305 restriction 58
 DEFINE
 command
 CHANNELS operand 58
 CONSOLE operand 58
 CTCA operand 58
 described 58
 GRAF operand 58
 LINE operand 58
 PRINTER operand 58
 PUNCH operand 58
 READER operand 58
 STORAGE operand 58
 summary 190
 TFB-512 operand 58
 TIMER operand 58
 T2305 operand 58
 T2314 operand 58
 T2319 operand 58
 T3310 operand 58
 T3330 operand 58
 T3340 operand 58
 T3350 operand 58
 T3370 operand 58
 T3380 operand 58
 1403 operand 58
 1443 operand 58
 2501 operand 58
 2540P operand 58
 2540R operand 58
 3203 operand 58
 3211 operand 58
 3262 operand 58
 3289E operand 58
 3505 operand 58
 3525 operand 58
 3800 operand 58
 DEFINE command
 temporary disks 14
 use with MSS devices 58
 virtual addresses 58
 defining
 fixed-block mode devices, for the virtual machine 61
 virtual devices 58
 delimiters, in command lines 4
 DETACH
 command
 described 66
 summary 190
 vaddr operand 66

DETACH CHANNEL
 command
 channel c operand 69
 described 69
 DETACH command, temporary disks 14
 detaching
 dedicated channels 69
 virtual devices 66
 DEV operand, of TAG command 167
 device-end interrupt pending for a virtual
 device, specifying 130
 devices
 virtual
 linking to 92
 simulating not ready status for 104
 DIAL
 command
 described 70
 summary 190
 userid operand 70
 vaddr operand 70
 DIAL command, device restriction 70
 directory entry, controlling, virtual
 machine attributes 12
 DISABLE, command, summary 190
 DISCONN
 command
 described 72
 HOLD operand 72
 summary 190
 disconnecting
 virtual console 72
 virtual machine 98
 DISPLAY
 command
 CAW operand 74
 CSW operand 74
 described 74
 PSW operand 74
 summary 190
 DISPLAY command, byte alignment 28
 displaying
 accounting information 138
 Channel Address Word (CAW) 74
 Channel Status Word (CSW) 74
 extended control registers 74
 floating-point registers 74
 general registers 74
 Program Status Word (PSW) 74
 storage keys 74
 virtual storage 27
 virtual storage locations 74
 DIST operand
 of CHANGE command 47
 of CLOSE command 51
 distribution code, for an output file 47
 DMCP, command, summary 190
 DRAIN, command, summary 190
 dsname operand, of CLOSE command 51
 DSS operand, VMDUMP command 185
 DUMP
 command
 described 80
 summary 190

E
 ECHO
 command
 described 84
 nn operand 84
 summary 190
 ECMODE operand, of SET command 138
 editing symbols
 controlling 170
 logical, controlling use of 138
 EMSG operand, of SET command 138
 ENABLE, command, summary 190
 END operand, of TRACE command 176
 entering
 a CP command from the virtual machine
 environment 57
 CP commands 3
 EOF operand, of SPOOL command 153
 error message handling, controlling 138
 ESCAPE operand, of TERMINAL command 170
 execution
 halting at an instruction address 43
 of a virtual machine, resuming 46
 extended control mode, setting 138
 extended control registers
 changing your virtual machine 163
 displaying 74
 printing 80
 EXTERNAL
 command
 code operand 84
 described 84
 summary 190
 external interrupt, simulated 84
 EXTERNAL operand, of TRACE command 176

F
 FB-512 device information, obtained with
 QUERY command 118
 FB-512 devices, tracing CCW information
 178
 FCB operand
 of CHANGE command 47
 of LOADVFCB command 96
 of SPOOL command 153
 FILE operand, of TAG command 167
 files
 changing the spool class of 47
 changing the spoolid for 47
 closing 51
 distribution code for 47
 holding before output processing 47
 holding from further processing 51,153
 holding via CLOSE 153
 purging 51,153
 purging via SPOOL 153
 reclaiming 181
 reordering closed spool files 105
 spool, changing filename for 47
 terminating processing of 51
 transferring to another user 153
 transmitting to a virtual reader 181

FILES operand, of QUERY command 112
 first-level storage, requesting via DISPLAY
 command 78
 FLASH operand
 of CHANGE command 47
 of SPOOL command 153
 floating-point registers
 changing your virtual machine 163
 displaying 74
 printing 80
 FLUSH, command, summary 190
 fnft operand, of CLOSE command 51
 FOR operand, of SPOOL command 153
 FORCE, command, summary 190
 FORM operand
 of CHANGE command 47
 of CLOSE command 51
 of ORDER command 105
 of PURGE command 107
 of SPOOL command 153
 of TRANSFER command 181
 FORMAT vtype operand, VMDUMP command 185
 forms controls for a virtual printer,
 specifying 96
 FREE, command, summary 190
 FROM operand, of TRANSFER command 181

G

general registers
 changing in your virtual machine 163
 displaying 74
 printing 80
 general user, described 8
 GRAF operand
 of DEFINE command 58
 of QUERY command 112

H

HALT, command, summary 190
 halting, execution at an instruction
 address 43
 handshaking, control of VM/VS 138
 hexloc operand
 of ADSTOP command 43
 of BEGIN command 46
 hexloc1 operand, VMDUMP command 183
 hexloc2 operand, VMDUMP command 185
 hiding your password 99
 highlighting, 3270 terminal display 172
 HIGHLIGHT operand, of TERMINAL command 170
 HOLD, command, summary 190
 HOLD operand
 of CHANGE command 47
 of CLOSE command 51
 of DISCONN command 72
 of LOGOFF command 98
 of SPOOL command 153
 holding
 a file before output processing 47
 files by user hold status 153
 files from further processing 51,153

I

IMSG operand, of SET command 138
 INDEX operand, for LOADVFCB command 96
 INDICATE
 command
 attached processor load indication
 85
 described 85
 LOAD operand 85
 multiprocessor load indication 85
 summary 190
 USER operand 85
 initial program load (IPL)
 automatic, suppression of 99
 of a virtual machine operating system
 88
 stopping 88
 input/output operations, tracing 176
 INSTRUCT operand, of TRACE command 176
 internal print file duplication, by 3800
 printer 47
 interrupt codes
 X'1004', clock comparator 23
 X'1005', CPU timer 23
 interruptions, simulating 22
 interrupts
 clearing from a virtual machine 132
 device-end, specifying virtual 130
 external, simulating 84
 for a virtual machine 131
 range of external codes 84
 tracing 176
 I/O devices, virtual, defining for your
 virtual machine 58
 I/O operand, of TRACE command 176
 IPL
 command
 ATTN operand 88
 CLEAR operand 88
 described 88
 nnnnnn operand 88
 NOCLEAR operand 88
 PARM operand 88
 STOP operand 88
 summary 190
 vaddr operand 88
 IPL command, for named systems 91
 ISAM operand, of SET command 138

K

keys
 program function, controlling 138
 storage
 displaying 74
 printing 80

L

line editing symbols, controlling 138
 line length, controlling at your terminal
 170
 LINE operand, of DEFINE command 58
 LINEDEL operand, of TERMINAL command 170
 LINEDIT operand, of SET command 138
 LINEND operand, of TERMINAL command 170

LINES operand, of QUERY command 112
 LINESIZE operand, of TERMINAL command 170
 LINK
 command
 described 92
 password operand 92
 summary 190
 userid operand 92
 vaddr operand 92
 LINK command
 batch job restrictions 92
 passwords 92
 2305 restrictions 92
 linking, a device to your virtual machine 92
 LINKS operand, of QUERY command 112
 load key, simulation of via IPL command 88
 LOAD operand, of INDICATE command 85
 LOADBUF, command, summary 191
 loading, an operating system 21
 loading a system, with a block address, at IPL 89
 LOADVFCB
 command
 described 96
 FCB operand 96
 INDEX operand 96
 summary 191
 vaddr operand 96
 LOCATE, command, summary 191
 LOCK, command, summary 191
 logging off of your virtual machine 98
 logging on your virtual machine 99
 logical editing symbols
 altering 170
 controlling 170
 controlling use of 138
 logically connected terminal 70
 LOGMSG operand, of QUERY command 127
 LOGOFF
 command
 described 98
 HOLD operand 98
 summary 191
 logon, affinity setting effect 99
 LOGON
 command
 described 99
 NOIPL operand 99
 password operand 99
 summary 191
 userid operand 99
 logon procedure, hiding your password during 99

M
 masking, password during logon 99
 Mass Storage Systems (see MSS)
 MESSAGE
 command
 * operand 102
 described 102
 msgtext operand 102
 OPERATOR operand 102
 summary 191
 userid operand 102

message handling, error, handling 138
 messages
 controlling transmission of 138
 sending to other users 102
 MODE operand, of TERMINAL command 170
 MODIFY operand
 of CHANGE command 47
 of SPOOL command 153
 modifying
 spooling control options 153
 storage, using ADSTOP command 43
 virtual device attributes 153
 MONITOR, command, summary 191
 MSG operand, of SET command 138
 MSGNOH, command, summary 191
 msgtext operand
 for MESSAGE command 102
 of SMSG command 152
 MSS, virtual volumes, defined with DEFINE command 58
 multiprocessor, INDICATE command to display system load 85

N
 NAME operand
 of CHANGE command 47
 of CLOSE command 51
 named systems, IPL command for 88
 NAMES operand, of QUERY command 127
 nesting, described 36
 NETWORK, command, summary 191
 nn operand, of ECHO command 84
 nnnnnn operand, of IPL command 88
 NOCLEAR operand, of IPL command 88
 NOCONT operand, of SPOOL command 153
 NOEOF operand, of SPOOL command 153
 NOHOLD operand
 of CHANGE command 47
 of CLOSE command 51
 of SPOOL command 153
 NOIPL operand, of LOGON command 99
 NORUN operand, of TRACE command 176
 notational conventions 35
 NOTERM operand, of SPOOL command 153
 NOTRANS operand, of SET command 138
 NOTREADY
 command
 described 104
 summary 191
 vaddr operand 104

O
 OFF operand
 of ADSTOP command 43
 of SPOOL command 153
 of TRACE command 176
 operands, command 4
 operating system
 for a virtual machine, passing parameters to 91
 initial program load for 88
 operating systems
 loading 21
 passing parameters at IPL 22

operator, sending messages to 102
 OPERATOR operand, of MESSAGE command 102
 ORDER
 command
 CLASS operand 105
 described 105
 FORM operand 105
 PRINTER operand 105
 PUNCH operand 105
 READER operand 105
 spoolid operand 105
 summary 191
 OS Indexed Sequential Access Method (ISAM)
 138

P
 PAGEX operand, of SET command 138
 parameters, passing to a virtual machine
 operating system 88
 PARM operand, of IPL command 88
 passing parameters to your virtual machine
 operating system 91
 password
 hiding during logon procedure 99
 invalid format for commandline,
 precaution 99
 with the LINK command 92
 password operand
 of LINK command 92
 of LOGON command 99
 PFnn operand
 of QUERY command 112
 of SET command 138
 power off vs logoff, with 3270 terminal 98
 print file requirements, for 3800 printer
 49
 printer, virtual, specifying forms
 controls for 96
 printer files, virtual, altering the
 attributes of 47
 PRINTER operand
 of CHANGE command 47
 of CLOSE command 51
 of DEFINE command 58
 of ORDER command 105
 of PURGE command 107
 of QUERY command 112
 of SPOOL command 153
 of TRACE command 176
 printing
 contents of virtual machine components
 80
 Program Status Word (PSW) 80
 virtual storage keys 80
 virtual storage locations 80
 printing first-level storage, with DUMP
 command 81
 PRIV operand, of TRACE command 176
 privilege classes
 for CP commands
 defined 7
 summarized 8
 privileged instructions, tracing 176
 processor, determining, attached or main
 109
 PROCESSOR operand, of QUERY command 112

program function keys, controlling 138
 PROGRAM operand, of TRACE command 176
 Program Status Word (PSW)
 changing your virtual machine 163
 displaying 74
 printing 80
 PSW operand, of DISPLAY command 74
 PUNCH
 operand
 of CHANGE command 47
 of PURGE command 107
 of QUERY command 112
 punch files, virtual, altering the
 attributes of 47
 PUNCH operand
 of CLOSE command 51
 of DEFINE command 58
 of SPOOL command 153
 PUNCH operator, of ORDER command 105
 PURGE
 command
 ALL operand 107
 CLASS operand 107
 described 107
 FORM operand 107
 PRINTER operand 107
 PUNCH operand 107
 READER operand 107
 spoolid operand 107
 summary 191
 PURGE operand
 of CLOSE command 51
 of SPOOL command 153
 purging
 closed files 51
 spooled files 153
 purging closed spool files 107

Q
 QUERY
 command
 ALL operand 112
 CHANNELS operand 112
 CLASS operand 112
 CONSOLE operand 112
 CPUID operand 112
 DASD operand 112
 described 109
 FILES operand 112
 GRAF operand 112
 LINES operand 112
 LINKS operand 112
 LOGMSG operand 127
 NAMES operand 127
 PFnn operand 112
 PRINTER operand 112
 PROCESSOR operand 112
 PUNCH operand 112
 READER operand 112
 SCREEN operand 109,112
 SECUSER operand 109,112
 SET operand 109
 spoolid operand 112
 STORAGE operand 112
 summary 191
 TAPES operand 112

TBL operand 112
 TERMINAL operand 109
 TIME operand 109
 UR operand 112
 USERS operand 127
 vaddr operand 112
 VIRTUAL operand 112
 VMSAVE operand 112
 QUERY operand, of TAG command 167

R
 reader files, virtual, altering the attributes of 47
 READER operand
 of CHANGE command 47
 of CLOSE command 51
 of DEFINE command 58
 of ORDER command 105
 of PURGE command 107
 of QUERY command 112
 of SPOOL command 153
 reading, an entire file continuously 153
 read/write access, with LINK command 92
 READY
 command
 described 130
 summary 191
 vaddr operand 130
 real computer
 reset button, simulating 165
 restart button, simulating 165
 reclaiming files, in order to purge 107
 reclaiming transmitted files 181
 reconfiguring, your virtual machine 25
 registers
 extended control
 displaying 74
 printing 80
 floating-point
 displaying 74
 printing 80
 general
 displaying 74
 printing 80
 remote spooling, TAG command 167
 remote spooling networks
 receiving files 160
 transmitting files 160
 remote terminal, connecting to a virtual machine 70
 removing a virtual device from your virtual machine 66
 REPEAT, command, summary 191
 REQUEST
 command
 described 131
 performs same function as ATTN command 45
 summary 191
 reserved IBM class, described 8
 RESET
 command
 described 132
 summary 191
 vaddr operand 132
 reset button, simulating 165
 RESET operand, of SYSTEM command 165
 response
 for ADSTOP command 44
 for DETACH CHANNEL command 69
 for ORDER command 106
 for PURGE command 108
 for QUERY CPUID command 114
 for QUERY FILES command 115
 for QUERY LINKS command 121
 for QUERY LOGMSG command 127
 for QUERY NAMES command 128
 for QUERY PFnn command 124
 for QUERY PROCESSOR command 126
 for QUERY READER, PRINTER, and PUNCH command 123
 for QUERY SCREEN command 122
 for QUERY SECUSER command 122
 for QUERY SET command 113
 for QUERY SPMODE command 126
 for QUERY S370E command 125
 for QUERY TERMINAL command 114
 for QUERY TIME command 113
 for QUERY USERS command 128
 for QUERY VIRTUAL CHANNELS command 115
 for QUERY VIRTUAL CONSOLE command 116
 for QUERY VIRTUAL DASD command 118
 for QUERY VIRTUAL GRAF command 116
 for QUERY VIRTUAL LINES command 119
 for QUERY VIRTUAL STORAGE command 121
 for QUERY VIRTUAL TAPES command 119
 for QUERY VIRTUAL UR command 119
 for QUERY VIRTUAL vaddr command 121
 for QUERY VMSAVE 125
 for RESET command 132
 for SCREEN command 133
 for SEND command 137
 for STORE command 164
 responses
 for #CP command 42
 for CHANGE command 50
 for CLOSE command 54
 for COUPLE command 55
 for CP command 57
 for DEFINE command 65
 for DETACH command 68
 for DIAL command 70
 for DISCONN command 72
 for DISPLAY command 79
 for DUMP command 82
 for ECHO command 84
 for INDICATE LOAD and USER 85
 for LINK command 93
 for LOGOFF command 98
 for LOGON command 100
 for MESSAGE command 103
 for REWIND command 133
 for SPOOL command 160
 for SYSTEM command 165
 for TAG command 169
 for TRACE command 177
 for TRANSFER command 182
 for VMDUMP command 183
 responses to CP commands, types of 39
 responses vs system messages 39
 restart button, simulating 165
 RESTART operand, of SYSTEM command 165

resume
 execution of your virtual machine 46
 use of your virtual machine after
 disconnecting it 72
 REWIND
 command
 described 133
 summary 192
 vaddr operand 133
 rewinding a real tape 133
 RUN operand
 of SET command 138
 of TRACE command 176

S
 saved systems, IPL command for 88
 SAVESYS, command, summary 192
 saving, virtual machine data 163
 SCREEN
 command
 'color' operand 134
 'exthilight' operand 134
 described 134
 SCREEN operand, of QUERY command 109,112
 SCRNSAVE operand, of TERMINAL command 170
 SECUSER operand, of QUERY command 109,112
 SEND
 command
 CP operand 137
 described 137
 text operand 137
 userid operand 137
 sending messages
 to other users 102
 to the operator 102
 service representative, described 8
 SET
 command
 ACNT operand 138
 AFFINITY operand 138
 ASSIST operand 138
 AUTOPOLL operand 138
 described 138
 ECMODE operand 138
 EMSG operand 138
 IMSG operand 138
 ISAM operand 138
 LINEDIT operand 138
 MSG operand 138
 NOTRANS operand 138
 PAGEX operand 138
 PFnn operand 138
 RUN operand 138
 SMSG operand 138
 summary 192
 TIMER operand 138
 VMSAVE operand 138
 WNG operand 138
 SET operand, of QUERY command 109
 shared segment, modification of 43,177
 shared segments, modification of 164
 sharing virtual disks 14

SHUTDOWN, command, summary 192
 simulating
 not ready status for a virtual device
 104
 the reset button on a real computer 165
 the restart button on a real computer
 165
 simulating interruptions 22
 SIO operand, of TRACE command 176
 SLEEP
 command
 described 151
 summary 192
 SMSG
 command
 described 152
 msgtext operand 152
 summary 192
 userid operand 152
 SMSG operand, of SET command 138
 SPACE, command, summary 192
 specifying, device-end interrupt for a
 virtual device 130
 SPOOL
 command
 * operand 153
 CHARS operand 153
 CLASS operand 153
 CLOSE operand 153
 CONSOLE operand 153
 CONT operand 153
 COPY* operand 153
 COPY operand 153
 described 153
 EOF operand 153
 FCB operand 153
 FLASH operand 153
 FOR operand 153
 FORM operand 153
 HOLD operand 153
 MODIFY operand 153
 NOCONT operand 153
 NOEOF operand 153
 NOHOLD operand 153
 NOTERM operand 153
 OFF operand 153
 PRINTER operand 153
 PUNCH operand 153
 PURGE operand 153
 READER operand 153
 START operand 153
 STOP operand 153
 summary 192
 SYSTEM operand 153
 TERM operand 153
 TO operand 153
 userid operand 153
 vaddr operand 153
 spool class
 for a file, changing 47
 modifying 153
 SPOOL command, use with TAG command 159
 spool command options, set for a virtual
 device 51

spool file
 altering TAG information 167
 changing filename for 47
 changing the attributes of 47
 characteristics
 changing 16
 querying 16
 setting 15
 reordering 105
 spoolid 158
 TAG command 167
 transmitting locally 158
 transmitting to remote location 159, 167
 spoolid number 158
 spoolid operand
 of ORDER command 105
 of PURGE command 107
 of QUERY command 112
 of TRANSFER command 181
 spooling
 considerations during abend 153
 considerations during shutdown 153
 control, options, modifying 153
 operator, described 8
 terminals session 15
 virtual console 17
 spooling output to a 3800 printer 158
 SPTAPE, command, summary 177
 START, command, summary 192
 START operand, of SPOOL command 153
 status
 virtual machine, determining 109
 words, tracing 176
 STATUS operand
 of STORE command 163
 STORE command, simulating hardware store 31
 STCP, command, summary 192
 STOP operand
 of IPL command 88
 of SPOOL command 153
 stopping, initial program load (IPL) procedure 88
 storage
 clearing to zeros 88
 keys, printing 80
 locations, virtual, displaying 74
 modification of 43
 virtual, altering 30
 storage keys, displaying 74
 storage locations
 virtual machine
 altering 163
 displaying 74
 printing 80
 STORAGE operand
 of DEFINE command 58
 of QUERY command 112
 STORE
 command
 described 163
 STATUS operand 163
 summary 192
 STORE command, STATUS operand, simulate hardware store 31
 storing
 CPU status for virtual machine 163
 virtual machine data 163
 summary, of CP commands 189
 suppressing, automatic IPL of a virtual machine operating system 99
 SVC operand, of TRACE command 176
 SYSTEM
 command
 CLEAR operand 165
 described 165
 RESET operand 165
 RESTART operand 165
 summary 192
 system analyst, described 8
 system messages vs CP command responses 39
 SYSTEM operand
 of SPOOL command 153
 VMDUMP command 185
 system operator, primary, described 8
 system programmer, described 8
 system resource operator, described 8
 system resources, indication of 85
 T
 TABCHAR operand, of TERMINAL command 170
 TAG
 command
 described 167
 DEV command 167
 FILE operand 167
 QUERY operand 167
 summary 192
 TAG command
 altering TAG information 167
 transmitting files to remote location 167
 use with SPOOL command 159
 tapes, rewinding 133
 TAPES operand, of QUERY command 112
 TBL operand, of QUERY command 112
 temporary, virtual disks 14
 temporary disk space, formatting and clearing 58
 TERM operand, of SPOOL command 153
 TERMINAL
 command
 APL operand 170
 ATTN operand 170
 BREAKIN operand 170
 CHARDEL operand 170
 CONMODE operand 170
 described 170
 ESCAPE operand 170
 HIGHLIGHT operand 170
 LINEDEL operand 170
 LINEND operand 170
 LINESIZE operand 170
 MODE operand 170
 SCRNSAVE operand 170
 summary 193
 TABCHAR operand 170
 TEXT operand 170

TERMINAL command, BRKKEY operand 170
 TERMINAL operand
 of QUERY command 109
 of TRACE command 176
 terminals
 console, disconnecting from your virtual machine 72
 control of attention handling 170
 input/output processing, controlling 170
 line length, controlling 170
 remote, connecting to a virtual machine 70
 session
 altering virtual storage 30
 changing spool file characteristics 16
 commands used to control, summary of 189
 controlling I/O functions 12
 controlling virtual machine 21
 debugging programs 26
 dedicated channels 21
 dedicated devices 18
 determining the length of time of 109
 displaying virtual storage 27
 entering CP commands 23
 logging on 21
 purging spool files 18
 querying spool file characteristics 16
 reordering spool files 19
 setting spool file characteristics 15
 simulating interruptions 22
 spooling 15
 store virtual machine status 31
 testing programs 26
 tracing virtual machine activity 31
 virtual console spooling 18
 virtual disks 13
 virtual disks, permanent 14
 virtual disks, temporary 14
 virtual unit record devices 15
 testing using the ECHO command 84
 terminating
 console input/output 153
 processing of files 51
 testing
 programs 26
 terminal 84
 TEXT, control of translation tables 170
 text format, of CP commands 39
 text operand, of SEND command 137
 TEXT operand, of TERMINAL command 170
 TFB-512 devices, defining fixed-block mode 61
 TFB-512 operand, of DEFINE command 58
 time, determining length for a terminal session 109
 TIME operand, of QUERY command 109
 timer, virtual, controlling 138
 TIMER operand
 of DEFINE command 58
 of SET command 138
 TO * operand, VMDUMP command 185
 TO operand
 of SPOOL command 153
 of TRANSFER command 181
 TO userid operand, VMDUMP command 185
 TRACE
 command
 ALL operand 176
 BOTH operand 176
 BRANCH operand 176
 CCW operand 176
 CSW operand 176
 described 176
 END operand 176
 EXTERNAL operand 176
 INSTRUCT operand 176
 I/O operand 176
 NORUN operand 176
 OFF operand 176
 PRINTER operand 176
 PRIV operand 176
 PROGRAM operand 176
 RUN operand 176
 SIO operand 176
 summary 193
 SVC operand 176
 TERMINAL operand 176
 TRACE output, closing 32
 tracing
 fixed-block mode device information 178
 virtual machine activity 31,176
 TRANSFER
 command
 ALL operand 181
 CLASS operand 181
 described 181
 FORM operand 181
 FROM operand 181
 spoolid operand 181
 summary 193
 TO operand 181
 userid operand 181
 transferring
 files
 to a virtual reader 181
 to another user 153
 translating, CCWs, controlling 138
 transmitting
 spool files
 locally 158
 to remote locations 159
 T2305 operand, of DEFINE command 58
 T2314 operand, of DEFINE command 58
 T2319 operand, of DEFINE command 58
 T3310 operand, of DEFINE command 58
 T3330 operand, of DEFINE command 58
 T3340 operand, of DEFINE command 58
 T3350 operand, of DEFINE command 58
 T3370 operand, of DEFINE command 58
 T3380 operand, of DEFINE command 58

 U
 underscore, use of 36
 UNLOCK, command, summary 193
 uppercase (letters and words) and special symbols, use of 35
 UR operand, of QUERY command 112

USER operand
 of INDICATE command 85
 of QUERY command 127
 userid operand
 of COUPLE command 55
 of DIAL command 70
 of LINK command 92
 of LOGON command 99
 of MESSAGE command 102
 of SEND command 137
 of SMSG command 152
 of SPOOL command 153
 of TRACE command 181
 using virtual forms control buffer
 with the 3262 printer 96
 with the 3289E printer 96

V
 vaddr operand
 of CLOSE command 51
 of DETACH command 66
 of DIAL command 70
 of IPL command 88
 of LINK command 92
 of LOADVFCB command 96
 of NOTREADY command 104
 of QUERY command 112
 of READY command 130
 of RESET command 132
 of REWIND command 133
 of SPOOL command 153
 vaddr1 operand, of COUPLE command 55
 vaddr2 operand, of COUPLE command 55
 VARY, command, summary 193
 virtual console
 disconnecting from your virtual machine 72
 read environment 24
 spooling 17,153
 virtual devices
 closing 153
 defining 58
 detaching from your virtual machine 66
 determining number of for your virtual machine 109
 determining the status of 109
 I/O, defining for your virtual machine 58
 removing from your virtual machine 66
 simulating not ready status for 104
 specifying a device-end interrupt for 130
 virtual disks, permanent 14
 virtual machine
 activity, tracing 176
 configuration, altering 58
 determining status of 109
 device, linking to 92
 directory entry 12
 disconnecting 98
 execution, resuming 46
 logging off of 98
 logging on 99
 operating system, passing parameters to 88
 placing in a dormant state 151
 reconfiguring 25
 sending and processing special messages 152
 setting control functions for 138
 storing information from 163
 tracing activity 31
 virtual machine environment, entering a CP command from 57
 VIRTUAL operand, of QUERY command 112
 virtual printer, specifying forms controls for 96
 virtual storage, altering 30
 virtual storage locations
 changing 163
 printing 80
 virtual timer, controlling 138
 virtual unit record devices 15
 virtual volumes (3330V), defining 58
 VMDUMP command
 *dumpid operand 185
 bytecount operand 185
 DSS operand 185
 FORMAT vmttype operand 185
 hexloc1 operand 183
 hexloc2 operand 185
 SYSTEM operand 185
 TO * operand 185
 TO userid operand 185
 VMSAVE operand
 of QUERY command 112
 of SET command 138
 VM/VS handshaking, control of 138

W
 WARNING, command, summary 193
 WNG operand, of SET command 138
 writing, comments in VM/370 4

Z
 zeros
 clearing storage to 165
 clearing storage to via IPL 88
 clearing storage to via SYSTEM command 165

1
 1403 operand, of DEFINE command 58
 1443 operand, of DEFINE command 58

2
 2501 operand, of DEFINE command 58
 2540P operand, of DEFINE command 58
 2540R operand, of DEFINE command 58

3
 3203 device
 as DEFINE operand 58
 LOADVFCB restrictions 96

3203 operand, of DEFINE command 58
3203 printer, virtual, specifying forms
controls for 96
3211 operand, of DEFINE command 58
3211 printer, virtual, specifying forms
controls for 96
3262 operand, of DEFINE command 58
3262 printer, loading a virtual forms
control buffer 96
3270 device message length 102,152
3270 terminal, power off vs logoff,
precautions 98
3270 terminal display, highlighting
features available 172
3289E operand, of DEFINE command 58
3289E printer, loading a virtual forms
control buffer 96
3505 operand, of DEFINE command 58
3525 operand, of DEFINE command 58
3800 operand, of DEFINE command 58
3800 printer
 modifying device attributes 153
 modifying spool file attributes 49
 print file duplication 47
 spooling output to one 158
 using CHANGE command 49
 using SPOOL command 153
3850 Mass Storage Systems (see MSS)

This manual is part of a library that serves as a reference source for systems analysts, programmers, and operators of IBM systems. This form may be used to communicate your views about this publication. They will be sent to the author's department for whatever review and action, if any, is deemed appropriate. Comments may be written in your own language; use of English is not required.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation whatever. You may, of course, continue to use the information you supply.

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

| | Yes | No | | |
|---|--------------------------|----------------------------|--------------------------|--|
| • Does the publication meet your needs? | <input type="checkbox"/> | <input type="checkbox"/> | | |
| • Did you find the material: | | | | |
| Easy to read and understand? | <input type="checkbox"/> | <input type="checkbox"/> | | |
| Organized for convenient use? | <input type="checkbox"/> | <input type="checkbox"/> | | |
| Complete? | <input type="checkbox"/> | <input type="checkbox"/> | | |
| Well illustrated? | <input type="checkbox"/> | <input type="checkbox"/> | | |
| Written for your technical level? | <input type="checkbox"/> | <input type="checkbox"/> | | |
| • What is your occupation? | <hr/> | | | |
| • How do you use this publication: | | | | |
| As an introduction to the subject? | <input type="checkbox"/> | As an instructor in class? | <input type="checkbox"/> | |
| For advanced knowledge of the subject? | <input type="checkbox"/> | As a student in class? | <input type="checkbox"/> | |
| To learn about operating procedures? | <input type="checkbox"/> | As a reference manual? | <input type="checkbox"/> | |

Your comments:

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

Thank you for your cooperation. No postage stamp necessary if mailed in the U.S.A. (Elsewhere, an IBM office or representative will be happy to forward your comments.)

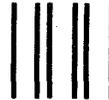
Reader's Comment Form

IBM Virtual Machine/System Product CP Command Reference for General Users
Cut or Fold Along Line
Printed in U.S.A.

Fold and Tape

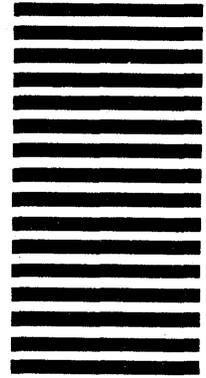
Please Do Not Staple

Fold and Tape



NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL
FIRST CLASS PERMIT NO. 40 ARMONK, N.Y.



POSTAGE WILL BE PAID BY ADDRESSEE:

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

Fold

Fold

If you would like a reply, please print:

Your Name _____

Company Name _____ Department _____

Street Address _____

City _____

State _____ Zip Code _____

IBM Branch Office serving you _____



International Business Machines Corporation
Data Processing Division
1133 Westchester Avenue, White Plains, N. Y. 10604

IBM World Trade Americas/Far East Corporation
Town of Mount Pleasant, Route 9, North Tarrytown, N. Y., U. S. A. 10591

IBM World Trade Europe/Middle East/Africa Corporation
360 Hamilton Avenue, White Plains, N. Y., U. S. A. 10601



International Business Machines Corporation
Data Processing Division
1133 Westchester Avenue, White Plains, N.Y. 10604

IBM World Trade Americas/Far East Corporation
Town of Mount Pleasant, Route 9, North Tarrytown, N.Y., U.S.A. 10591

IBM World Trade Europe/Middle East/Africa Corporation
360 Hamilton Avenue, White Plains, N.Y., U.S.A. 10601