File No. S370-36
Order No. GC20-1820-4

Systems

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

**Release 6 PLC 17**

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

Each CP command available to the general class of user as well those CP commands that are available to all users are listed alphabetically and contain general usage information, the command line format, descriptions of all allowable operand values, 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/370 CP commands.

PREREQUISITE PUBLICATIONS

*IBM Virtual Machine Facility/370:*

*Introduction,* Order No. GC20-1800

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

COREQUISITE PUBLICATIONS

*IBM Virtual Machine Facility/370:*

*System Messages,* Order No. GC20-1808

Fifth Edition (April 1981)

This is a major revision of, and obsoletes, GC20-1820-3 and Technical Newsletter GN25-0494. This edition applies to Release 6 PLC 17 (Program Level Change) of the IBM Virtual Machine Facility/370 and to all subsequent releases unless otherwise indicated in new editions or Technical Newsletters.

The entire section headed "CP Command Usage" has been added to show users how to control virtual machine functions with CP commands.

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

Changes are periodically made to the information contained herein; before using this publication in connection with the operation of IBM systems, consult the IBM System/370 Bibliography, Order No. 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 which 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.

© Copyright International Business Machines Corporation 1972, 1973, 1974, 1975, 1976, 1977, 1979, 1981

# Preface

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

The user should be knowledgeable of the concepts of VM/370 as outlined in the IBM Virtual Machine Facility/370: Introduction, Order No. GC20-1800. 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 Facility/370: Terminal User's Guide, Order No. GC20-1810.

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/370 CP commands, available only to system operators, system programmers and analysts, and service representatives, are described in the publication IBM Virtual Machine Facility/370: Operator's Guide, Order No. GC20-1806.

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 VM/370 Command Languages" describes the VM/370 command environments and the general structure of the command languages.

"Section 2. The CP Command Language" provides additional information on the CP subset of VM/370 commands.

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

In this publication, the following terminology is used:

- "2305" refers to the IBM 2305 Fixed Head Storage, Models 1 and 2.
- The term, "3270", is used in this publication to refer to a series of display devices, namely, the IBM 3275, 3276, 3277 and 3278 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, and 3158 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 3287, 3288 and 3289 printers unless otherwise noted.
- "3262" refers to the IBM 3262 Printer, Models 1 and 11.
- "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.
- "FB-512" refers to the IBM 3310 and 3370 Direct Access Storage Devices.
- "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 Facility/370: Planning and System Generation Guide, Order No. GC20-1801, 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. The terms "processor" and "attached processor" are not to be construed to mean an operation within a system complex where both processors are to perform I/O operations. VM/370 supports I/O on only one processor in a two processor configuration.
- "3800" refers to the IBM 3800 Printing Subsystem.
- "3850" refers to the IBM 3850 Mass Storage System.

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

PREREQUISITE PUBLICATIONS

The concepts of VM/370 and its four components, CP, CMS, RSCS, and IPCS are described in the IBM Virtual Machine Facility/370: Introduction, Order No. GC20-1800.

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

If the IBM 3767 Communications Terminal is to be used as a virtual console, the publication IBM 3767 Operator's Guide, Order No. GA18-2000, should be used in conjunction with VM/370 Terminal User's Guide.

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 Facility/370: System Messages, Order No. GC20-1808.

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

IBM Virtual Machine Facility/370:

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

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

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

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

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

# Contents

The entries in this Table of Contents are accumulative. They list additions to this publication by the following VM/370 System Contol Program Products:

- VM/370 Basic System Extensions, Program Number 5748-XX8
- VM/370 System Extensions, Program Number 5748-XE1

However, the text within the publication is not accumulative; it only relates to the one SCP program product that is installed on your system. Therefore, there may be topics and references listed in this Table of Contents that are not contained in the body of this publication.


SUMMARY OF AMENDMENTS. . . . . ix

PART 1. GUIDANCE INFORMATION . . . . . .1

PART 1. GUIDANCE INFORMATION . . . . . .1

SECTION 1. THE VM/370 COMMAND LANGUAGES. 3
VM/370 Command Environments. . . . . 3
VM/370 CP Command Structure. . . . . 3
The Command Name . . . . . 4
The Command Operands . . . . . 4
Comments in the CP Command Language. . 4
Character Set Usage. . . . . 4

SECTION 2. THE CP COMMAND LANGUAGE . . . 7
Privilege Classes for CP Commands. . . . 7
CP Commands Accepted From Each User Class Class. . . . . .9
The Class G Commands . . . . . .2
The Class Any Commands . . . . . 12

SECTION 3. CP COMMAND USAGE. . . . . 13
Controlling Input and Output Functions . 14
Virtual Disks. . . . . 15
Permanent Virtual Disks. . . . . 16
Temporary Virtual Disks. . . . . 16
Sharing Virtual Disks. . . . . 16
Virtual Unit Record Devices. . . . . 17
Virtual Unit Record Spooling . . . . . 17
Spool File Characteristics . . . . . 18
Virtual Console Spooling . . . . . 19
Reordering and Purging Spool Files . . 20
Dedicated Devices. . . . . 20
Dedicated Channels . . . . . 22
Controlling the Virtual Machine. . . . . 22
Loading an Operating System. . . . . 22
Simulating Interrupts. . . . . 24
Entering CP Commands . . . . . 24
Entering CP Commands when the Virtual Machine is Running . . . . . 24
Entering CP Commands from the Virtual Console Read Environment . . . . . 25
Entering CP Commands from CP Console Function Mode. . . . . 26
Reconfiguring the Virtual Machine. . . 26
Testing and Debugging of Programs. . . . 27
Stopping Execution of Your Virtual Machine. . . . . 28
Displaying Virtual Storage . . . . . 28
Terminal Output. . . . . 28
Byte Alignment on Terminal Output. . . 29
Printer Output . . . . . 30
Altering Virtual Storage . . . . . 31
Tracing Virtual Machine Activity . . . 33

PART 2. REFERENCE INFORMATION. . . . . 35

SECTION 4. NOTATIONAL CONVENTIONS. . . . 37

SECTION 5. FORMAT OF CP COMMANDS . . . . 41
*. . . . . 42
#CP. . . . . 43
ADSTOP . . . . . 45
ATTN . . . . . 47
BEGIN. . . . . 48
CHANGE . . . . . 49
CLOSE. . . . . 53
COUPLE . . . . . 57
CP . . . . . 59
DEFINE . . . . . 60
DETACH . . . . . 66
DETACH CHANNEL . . . . . 69
DIAL . . . . . 70
DISCONN. . . . . 72
DISPLAY. . . . . 74
DUMP . . . . . 80
ECHO . . . . . 83
EXTERNAL . . . . . 84
INDICATE . . . . . 85
IPL. . . . . 88
LINK . . . . . 91
LOADVFCB . . . . . 95
LOGOFF . . . . . 97
LOGON. . . . . 99
MESSAGE. . . . . .102
NOTREADY . . . . . .104
ORDER. . . . . .105
PURGE. . . . . .107
QUERY. . . . . .109
QUERY Command for Class G Users. . . . .110

QUERY Command for All Classes of Users (Except Class Any) . . . . . .123
READY. . . . . .125
REQUEST. . . . . .126
RESET. . . . . .127
REWIND . . . . . .128
SET. . . . . .129
SLEEP. . . . . .139
SMSG . . . . . .140
SPOOL. . . . . .141
STORE. . . . . .150
SYSTEM . . . . . .153
TAG. . . . . .154
TERMINAL . . . . . .157
TRACE. . . . . .161
TRANSFER . . . . . .166
VMDUMP . . . . . .168

APPENDIX: VM/370 COMMAND SUMMARY . . . .173

INDEX. . . . . .185

FIGURES

Figure 1. The Virtual Machine Facility/370 Library.....vii
Figure 2. Character Sets and Their Contents.....5
Figure 3. CP Privilege Class Descriptions.....8
Figure 4. Commands Accepted from Each User Class.....9
Figure 5. Controlling the Virtual Machine During a Terminal Session.....13
Figure 6. System Action Taken on a #CP Command.....44
Figure 7. Action Taken on CLOSE for a Virtual Input Device.....55
Figure 8. Action Taken on CLOSE for a Virtual Output Device.....55
Figure 9. Password Requirements on the Link Command.....93
Figure 9A. 3270 High Intensity Features (5748-XX8).....160.1
Figure 9A. 3270 High Intensity Features (5748-XE1).....160.1
Figure 10. CP Command Summary.....173
Figure 11. CMS Command Summary.....178
Figure 12. CMS Commands for System Programmers.....183
Figure 13. RSCS Command Summary.....184

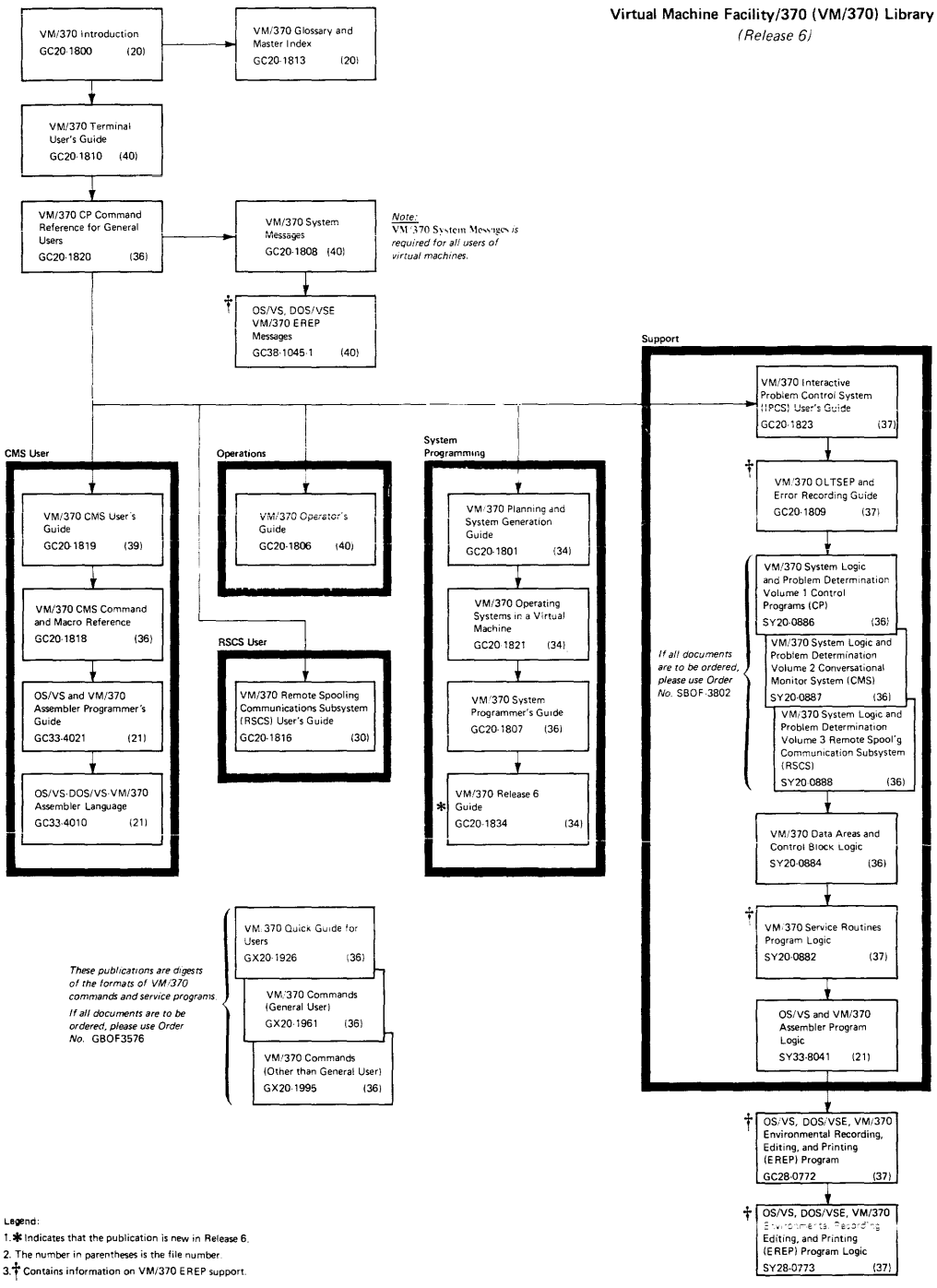



Figure 1. The Virtual Machine Facility/370 Library

MISCELLANEOUS

Changed: Documentation only

This major revision incorporates minor technical and editorial changes.

CP DUMP SERVICES FOR VIRTUAL MACHINES

New: Command for CP

A new CP VMDUMP command for general users dumps virtual storage of quest virtual machines when used with the VM/Interactive Problem Control System Extension (VM/IPCS Extension) program product. VMDUMP provides the VM/IPCS Extension with header information to identify the owner of the dump; the command also formats the dump and writes it to a spool file.

Summary of Amendments
for GC20-1820-3
VM/370 Release 6 PLC 1

## VM/370 SUPPORTS THE 3800 PRINTER SUBSYSTEM

New: Device Support

VM/370 provides support for the 3800 printer as a real spooling device. As a spooling device, the following hardware features of the 3800 printer are supported:

- Automatic loading of character set arrangement tables
- Graphic modification support
- Forms overlay feature (flashing)
- Copy modification

To facilitate 3800 Printer Subsystem support, the following CP commands have been modified:

CHANGE
QUERY
SPOOL

## VM/370 SUPPORTS THE 4331 AND 4341 PROCESSORS

New: Device Support

With the 4331 and 4341 processors, VM/370 provides system console support for the 3278 Model 2A display station. The 3278 Model 2A console is a variation of the 3278 display console.

Impact of this new device support has resulted in the addition of a new operand to the TERMINAL command and a new response to the QUERY TERMINAL command.

## VM/370 OFFERS MASS STORAGE SYSTEM (MSS) EXTENDED SUPPORT

New: Program and Documentation

For Release 6, MSS minidisk support has been added. The minidisk appears to the virtual machine to be defined on a permanently mounted IBM 3330-1 disk drive.

This new support has resulted in the modification of the DEFINE command in Figure 4 and Figure 10.

## VM/370 SUPPORTS SPECIAL MESSAGE FACILITY

New: Program Feature

The special message facility provides a means of transmitting special messages from a user to a specially programmed virtual machine for processing.

A new SMSG command has been added to the class G CP commands that permits a user to issue special messages. The SET command also has a new operand, SMSG. Further information on the special message facility can be obtained in the description of these commands.

## VM/370 SUPPORTS AUTOMATIC REINITIALIZATION

New: Program Feature

This support enables a CMS virtual machine to automatically execute its PROFILE EXEC and establish itself without forcing the user to hit the enter key after IPL.

The AUTOCR (automatic carriage return) option has been added to the PARM operand of the IPL command. Refer to the "Usage Notes" discussion for the IPL command.

VM/370 SHARED SEGMENT SUPPORT MODIFIED

Changed: Program and Documentation

VM/370 has modified the handling of users who alter storage in shared segments. The unshare process has been enhanced to reflect this support.

The "Usage Notes" discussions for the ADSTOP, STORE, and TRACE commands have been documented to reflect this support.

VM/370 SUPPORTS PASSWORD-ON-THE-COMMAND-LINE SUPPRESSION

New: Program Feature

VM/370 supports the suppression of the entering of passwords on the command line for LOGON and LINK commands. The intent is to force passwords to be masked for security purposes.

The LOGON and LINK commands have been updated to reflect this system enhancement. Refer to the discussion of these commands.

MISCELLANEOUS

Changed: Program and Documentation

- A new guidance section "CP Command Usage" has been added to Part 1.
- Various technical and editorial changes have been made to this publication.

# Part 1. Guidance Information

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

"Section 1. The VM/370 Command Language" describes the VM/370 command environments and the general structure of the command languages.

"Section 2. The CP Command Language" provides additional information on the CP subset of VM/370 commands.

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

# Section 1. The VM/370 Command Languages

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

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

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

## VM/370 Command Environments

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

You are in the control program (CP) command environment when you log on to VM/370 and issue CP commands.

You are in the virtual machine command environment when you load your virtual machine with CMS or 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/370 responds to a null line by displaying the current command environment, CMS or CP.

## VM/370 CP Command Structure

A VM/370 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/370 system. In this case, you would use the CP QUERY command.

## THE COMMAND OPERANDS

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

Some commands require no operands; others require several. You 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/370 commands are entered using a combination of characters from six different character sets. The contents of each of the character sets is described in Figure 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/370 system and (2) by a user to control a virtual machine. In this publication, only those commands that are used to control a virtual machine are fully documented. Other commands are referenced only to the extent necessary to show relationships and provide general knowledge.

## Privilege Classes for CP Commands

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

Figure 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.

Figure 4 is a list of CP commands by privilege classes.

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

| Class | User and Function |
|---|---|
| A[1] | Primary System Operator: The class A user controls the VM/370 system. Class A is assigned to the user at the VM/370 system console during IPL. The primary system operator is responsible for the availability of the VM/370 system and its communication lines and resources. In addition, the class A user controls system accounting, broadcast messages, virtual machine performance options, and other command operands that affect the overall performance of VM/370. The system operator controls operation of the real machine using the system control panel and console device.<br><br>Note: The class A system operator who is automatically logged on during CP initialization is designated as the primary system operator. |
| B[1] | System Resource Operator: The class B user controls allocation and deallocation of all the real resources of the VM/370 system, except those controlled by the primary system operator and spooling operator. |
| C[1] | System Programmer: The class C user updates certain functions of the VM/370 system. The system programmer can modify real storage in the real machine. |
| D[1] | Spooling Operator: The class D user controls spool data files and specific functions of the system's unit record equipment. |
| E[1] | System Analyst: The class E user displays the contents of real storage, performs the functions required to generate saved systems and discontiguous saved segments, and controls the collection and recording of performance measurement data. This class of user can display the contents of specified real storage areas on the virtual operator's console or on a spooled virtual printer, but cannot modify real storage. |
| F[1] | Service Representative: The class F user obtains, and examines, in detail, certain data about input and output devices connected to the VM/370 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. |
| G[2] | General User: The class G user controls functions associated with the execution of his virtual machine. A general user cannot display or modify real storage. |
| Any[2] | The Any classification is given to certain CP commands that are available to any user. These are primarily for the purpose of gaining and relinquishing access to the VM/370 system. |
| H | Reserved for IBM use. |

[1]Described in the VM/370 Operator's Guide.
[2]Described in this publication.

Figure 3. CP Privilege Class Descriptions

## CP Commands Accepted from Each User Class

The following information should help the user interpret Figure 4. The table contains every CP command but it does not contain every operand. Figure 4 only lists those operands that fall into these categories:

- The operand for a command is unique to one particular privilege class; the operand cannot be specified by the same command when the command belongs to another privilege class. An example is the TRACE operand which is only valid for the class F NETWORK command, but invalid for the command in class A or B.
- The operand is available for a command in several privilege classes, but one or more privilege classes exist that do not support the operand. An example is the SET command which has an AFFINITY operand available for class A and G commands, but not in the B or F class.

Commands listed without operands indicate that any valid operand associated with the command can be specified in any class the command belongs to.

The operands listed in the table in lowercase indicate either an alphameric or numeric specification is required.

These lowercase operands include user identification (userid), virtual address (vaddr), real address (raddr), and device type (device). For example:

| Operand | Alphameric or Numeric |
|---|---|
| userid | pd58gh1c |
| vaddr | 00e, 00c, 1ac, 005 |
| raddr | 008, 00a, fff |
| device | t3330, 3211, 2314, 1403 |

| Class | Commands | Operands |
|---|---|---|
| Any | * | |
| | #CP | |
| | CP | |
| | DIAL | |
| | DISCONN | |
| | LOGOFF | |
| | LOGON | |
| | MESSAGE | |
| | SLEEP | |
| | | |
| A | ACNT | |
| | AUTOLOG | |
| | DISABLE | |
| | ENABLE | |
| | FORCE | |
| | HALT | |
| | LOCK | |
| | MESSAGE | ALL |
| | MONITOR | |
| | NETWORK | DISABLE |
| | | DISPLAY |
| | | DUMP |
| | | ENABLE |

| Class | Commands | Operands |
|---|---|---|
| A | | HALT |
| (cont.) | | LOAD |
| | | POLLDLAY |
| | | QUERY |
| | | SHUTDOWN |
| | | VARY |
| | QUERY | AFFINITY |
| | | CPASSIST |
| | | JOURNAL |
| | | LOGMSG |
| | | NAMES |
| | | PAGING |
| | | PRIORITY |
| | | PROCESSOR |
| | | SASSIST |
| | | userid |
| | | USERS |
| | SET | AFFINITY |
| | | CPASSIST |
| | | FAVORED |
| | | JOURNAL |
| | | PRIORITY |
| | | RESERVE |

Figure 4. Commands Accepted from Each User Class (Part 1 of 3)

| Class | Commands | Operands |
|---|---|---|
| A (cont.) | | SASSIST |
| | SHUTDOWN | |
| | UNLOCK | |
| | WARNING | |
| B | ATTACH | |
| | ATTACH | CHANNEL |
| | AUTOLOG | |
| | DEFINE | SYSVIRT |
| | | VIRTUAL |
| | DETACH | |
| | DETACH | CHANNEL |
| | DISABLE | |
| | ENABLE | |
| | MESSAGE | ALL |
| | MSGNOH | |
| | NETWORK | DISABLE |
| | | DISPLAY |
| | | DUMP |
| | | ENABLE |
| | | LOAD |
| | | POLLDLAY |
| | | QUERY |
| | | VARY |
| | QUERY | ALL |
| | | DASD |
| | | DUMP |
| | | GRAF |
| | | LINES |
| | | LOGMSG |
| | | NAMES |
| | | PROCESSOR |
| | | raddr |
| | | STORAGE |
| | | SYSTEM |
| | | TAPES |
| | | TDSK |
| | | UR |
| | | userid |
| | | USERS |
| | SET | DUMP |
| | | LOGMSG |
| | VARY | |
| | WARNING | |
| C | DCP | |
| | DMCP | |
| | LOCATE | |
| | QUERY | LOGMSG |
| | | NAMES |
| | | PROCESSOR |
| | | userid |
| | | USERS |
| | STCP | |
| D | BACKSPAC | |
| | CHANGE | |
| | DRAIN | |
| | FLUSH | |
| | FREE | |
| D (cont.) | HOLD | |
| | LOADBUF | |
| | ORDER | |
| | PURGE | |
| | QUERY | FILES |
| | | HOLD |
| | | LOGMSG |
| | | NAMES |
| | | PRINTER |
| | | PUNCH |
| | | READER |
| | | userid |
| | | USERS |
| | REPEAT | |
| | SPACE | |
| | START | |
| | TRANSFER | |
| E | DCP | |
| | DMCP | |
| | INDICATE | I/O |
| | | LOAD |
| | | PAGING |
| | | QUEUES |
| | | USER |
| | LOCATE | |
| | MONITOR | |
| | QUERY | AFFINITY |
| | | CPASSIST |
| | | JOURNAL |
| | | LOGMSG |
| | | NAMES |
| | | PAGING |
| | | PRIORITY |
| | | PROCESSOR |
| | | SASSIST |
| | | userid |
| | | USERS |
| | SAVESYS | |
| F | NETWORK | TRACE |
| | QUERY | LOGMSG |
| | | NAMES |
| | | userid |
| | | USERS |
| | SET | RECORD |
| | | MODE |
| G | ADSTOP | |
| | ATTN | |
| | BEGIN | |
| | CHANGE | |
| | CLOSE | |
| | COUPLE | |
| | DEFINE | READER |
| | | PRINTER |
| | | PUNCH |
| | | CONSOLE |
| | | CTCA |
| | | TIMER |
| | | CHANNELS |

Figure 4. Commands Accepted from Each User Class (Part 2 of 3)

| Class | Commands | Operands |
|---|---|---|
| G (cont.) | | LINE |
| | | GRAF |
| | | STORAGE |
| | | vaddr |
| | | device[1] |
| | DETACH | |
| | DETACH | CHANNEL |
| | DISPLAY | |
| | DUMP | |
| | ECHO | |
| | EXTERNAL | |
| | INDICATE | LOAD |
| | | USER |
| | IPL | |
| | LINK | |
| | LOADVFCB | |
| | NOTREADY | |
| | ORDER | |
| | PURGE | |
| | QUERY | ALL |
| | | CHANNELS |
| | | CONSOLE |
| | | DASD |
| | | FILES |
| | | CUPID |
| | | GRAF |
| | | LINES |
| | | LINKS |
| | | LOGMSG |
| | | NAMES |
| | | PFnn |
| | | PRINTER |
| | | PUNCH |
| | | PROCESSOR |
| | | READER |
| | | SET |
| | | STORAGE |
| | | TAPES |
| | | TERMINAL |
| | | TIME |
| | | UR |

| Class | Commands | Operands |
|---|---|---|
| G (cont.) | | userid |
| | | USERS |
| | | vaddr |
| | | VIRTUAL |
| | READY | |
| | REQUEST | |
| | RESET | |
| | REWIND | |
| | SET | AFFINITY |
| | | ACNT |
| | | ASSIST |
| | | AUTOPOLL |
| | | CPUID |
| | | ECMODE |
| | | EMSG |
| | | IMSG |
| | | ISAM |
| | | LINEDIT |
| | | MSG |
| | | NOTRANS |
| | | PAGEX |
| | | PFnn |
| | | RUN |
| | | SMSG |
| | | TIMER |
| | | WNG |
| | SMSG | |
| | SPOOL | |
| | STORE | |
| | SYSTEM | |
| | TAG | DEV |
| | | FILE |
| | | QUERY |
| | TERMINAL | |
| | TRACE | |
| | TRANSFER | |
| | VMDUMP | |

[1] For valid spooling device types see the discussion of the DEFINE command in Section 5.

Figure 4. Commands Accepted from Each User Class (Part 3 of 3)

## 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); simulate an external interruption for the CPU timer, clock comparator, or external interruption button on the system console (EXTERNAL command); initiate tracing facilities (TRACE 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. The LOGON command is used to gain initial access to the VM/370 system. This command also 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/370 system with console keys and switches. Figure 5 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 installation | Establish userid, user classes, passwords, storage requirements, disk space, and I/O devices | None |
| 2 Installation builds VM/370 directory entry for system user information | Directory entry provides CP with information necessary to create a virtual machine | DIRECT command (Figure 12) |
| 3 User can activate terminal and begin work session | Enter CP commands to perform the function required | Available class G commands (Figure 10) |
| | a. activate terminal | LOGON command (Figure 10) |
| | b. transfer and store data | LINK, SPOOL, STORE, and TRANSFER commands (Figure 10) |
| | c. record and control data | CHANGE, CLOSE, DISPLAY, DUMP, ORDER, PURGE, QUERY, and SPOOL commands (Figure 10) |

[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/370 commands and system responses. All user input is shown in lowercase, and all system responses are shown in uppercase.

Figure 5. 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 10) |
| | e. communicating with the virtual machine | #CP, ADSTOP, ATTN, BEGIN, EXTERNAL, IPL, REQUEST, SET commands (Figure 10) |
| | f. handling errors | TRACE, DISPLAY, and DUMP commands (Figure 10) |
| | g. end session | LOGOFF command (Figure 10) |

Figure 5. Controlling the Virtual Machine During a Terminal Session (Part 2 of 2)

## Controlling Input and Output Functions

The VM/370 directory entries for your virtual machine include, 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 3270                 (4)
        SPOOL 00C 2540 R
(5)     SPOOL 00D 2540 P
        SPOOL 00E 1403
        MDISK 130 2314 050 050 VDISK1 WR RWPASSWD
(6)     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/370 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 or it is spooled to a virtual reader. 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/370 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 VM/370 directory entry). They are temporarily attached to a user's virtual machine as needed.

## VIRTUAL DISKS

Under VM/370, 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/370, are extents on real DASD. To the virtual machine, they are functional equivalents of real disks. They can range from 1 cylinder to all of the cylinders 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/370 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.

Note: When temporary disk space is released to the system, it is not automatically cleared. Another user, requesting temporary disk space and receiving all or part of your former disk area, can access any data that you had left there. To preserve security, you should clear all temporary disk space before detaching it or logging off.

## 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/370 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/370 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/370 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/370 organizes the output data into a spool file. VM/370 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/370 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/370. Some of these characteristics such as spooling class and distribution code are assigned in the VM/370 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/370 identifies individual spool files by assigning each file a spool identification number (spoolid). This number ranges from 0001 through 9900 after 9900, VM/370 starts again from 0001. One series of spoolids covers the reader, printer, punch, and console spool files. VM/370 normally assigns the spoolid to a spool file when that file is closed. An exception to this is a console spool file where VM/370 assigns the number when the file is opened. Also, if you close a console file without stopping console spooling, VM/370 automatically opens another console spool file with a new spoolid at the same time.

The spooling class of a virtual device logically groups its output with that 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.

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

```
change printer 1234 class m
```

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

```
query printer all
```

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

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

```
spool printer hold
```

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

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

```
close printer nohold
```

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

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

```
query printer
```

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

```
change printer 0246 nohold
```

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

```
spool printer nohold
```

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

```
spool reader hold
```

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

```
spool reader nohold
```

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

```
purge reader all
```

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

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

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

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

## VIRTUAL CONSOLE SPOOLING

While spooling is usually associated with unit record equipment, VM/370 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/370 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/370 System" of the VM/370 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/370 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/370 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/370 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/370 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/370 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 locked when the virtual machine is running, you can enter either of the following commands:

```
#cp attn

 -- or --

#cp request
```

to both stop the 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 VIPTUAL 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/370 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 can use only six channels 0-5), you can only define virtual devices with addresses up through 5FF; in extended control (EC) mode, your virtual machine can use 16 channels 0-F, and 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.

When using the SPOOL, SPECIAL and DEDICATE directory control statements to define virtual devices, specify virtual addresses that do not conflict with or contend with the virtual control unit interface. This conflict or contention occurs because devices can require special I/O interface protocols 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) devices 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 both 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/370 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, and STORE commands, it is important that you understand the levels of storage you can specify on the command line.

Only first 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 and OS/VS as well as storage in OS/PCP, MFT, and MVT. The V=V partitions or regions of DOS/VS and OS/VS constitute second 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.

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

## TERMINAL OUTPUT

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

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

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

For the first format, enter the DISPLAY command as:

```
display 1026-102c
```

you receive the response:

```
001024 xxxxxxxx xxxxxxxx xxxxxxxx
```

For the second format, enter the command as:

```
display t1026-102c
```

and the response is:

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

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

```
display 1024.12
```

The response displays 20 bytes as follows:

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

## BYTE ALIGNMENT ON TERMINAL OUTPUT

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

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

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

If you enter:

```
display k1024-3200
```

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

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

To display all storage keys, enter:

```
display k
```

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

```
display X1 4 A
```

and receive the response:

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

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

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

As each operand in the command line is processed, VM/370 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.

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.
You can store data into one or multiple consecutive registers.

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

```
store y2 00123456789
```

loads floating-point register 2 with the value 0012345678900000.

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

```
store status
```

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

```
set ecmode on
```

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

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

| Virtual Address Dec. | 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 |
| 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

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/370 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/370 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.

- The abbreviations fn and ft are used to refer to filename and filetype respectively. The combination of fn ft are sometimes collectively referred to as file identification or fileid.

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

  A
  B
  C

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

  Example
  The representation

  A
  B
  C

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

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

  Example
  The representation

  { A B C }

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

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

  Example
  The representation

  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 ]]]
  [filename [filetype [filemode]]]
  ```

  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. Thus in the previous example, the user has the option of selecting a file by filename only, or filename filetype only or by filename filetype filemode. The user cannot select filetype alone because filetype is nested within filename; and, as noted, our rule states: the higher level of nesting must be selected in order to select the next level (lower level) operand. The same is true if the user wants to select filemode; filename and filetype must also be selected.

- 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/370 system messages. The command responses are not prefixed and, therefore, they are not contained in VM/370 System Messages. 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. For example, if you issue the following command:

```
spool copy 23
```

you receive the following system message:

```
DMKCSP022E  VADDR MISSING OR INVALID
```

where:

DMK is the CP component identifier.

CSP is the module identifier (SPOOL command).

022 is the return code.

E is the action code (indicating a severity classification).

msgtext describes the error situation in the command line. In this case, Printer (00E), Punch (00D), or Reader (00C) was omitted. Refer to VM/370 System Messages for additional 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 is the VM/370 supplied logical line end symbol. However, your installation can redefine the logical line end symbol currently in effect for your virtual machine. If you have redefined the logical line end symbol, #CP is an invalid command; you must substitute your line end symbol for the pound sign when using this command.

2. For the command to operate, the following conditions must be met:

   - The virtual machine must be operating with SET LINEDIT ON (a default).
   - The first three characters of the edited line must be #CP (uppercase or lowercase) with the "#" representing the logical line end symbol currently defined.
   - At least one blank must separate the #CP from any command line. Do not use attention interruption in any part of the line or to enter the line.
   - If in a virtual machine command environment, the virtual machine must have issued a READ for at least four bytes of data. If the READ is for less than 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.

5. If you have previously issued a TERMINAL APL ON command, you cannot issue the #CP command. This is because CP does not translate the #CP command. When APL is set on, CP uses a different set of

translation tables and does not recognize the logical line end character (#).

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

Figure 6. System Action Taken on a #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.

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. The SET RUN OFF command should be issued before an 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:

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 I/O is in progress at the console, or if other interrupts are pending. This condition can occur if you issue the ATTN command using DIAGNOSE X'08'.

Responses

None.

# BEGIN

Privilege Class: G

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

```
| Begin | [hexloc]                                                |
```

where:

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

Usage Notes

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

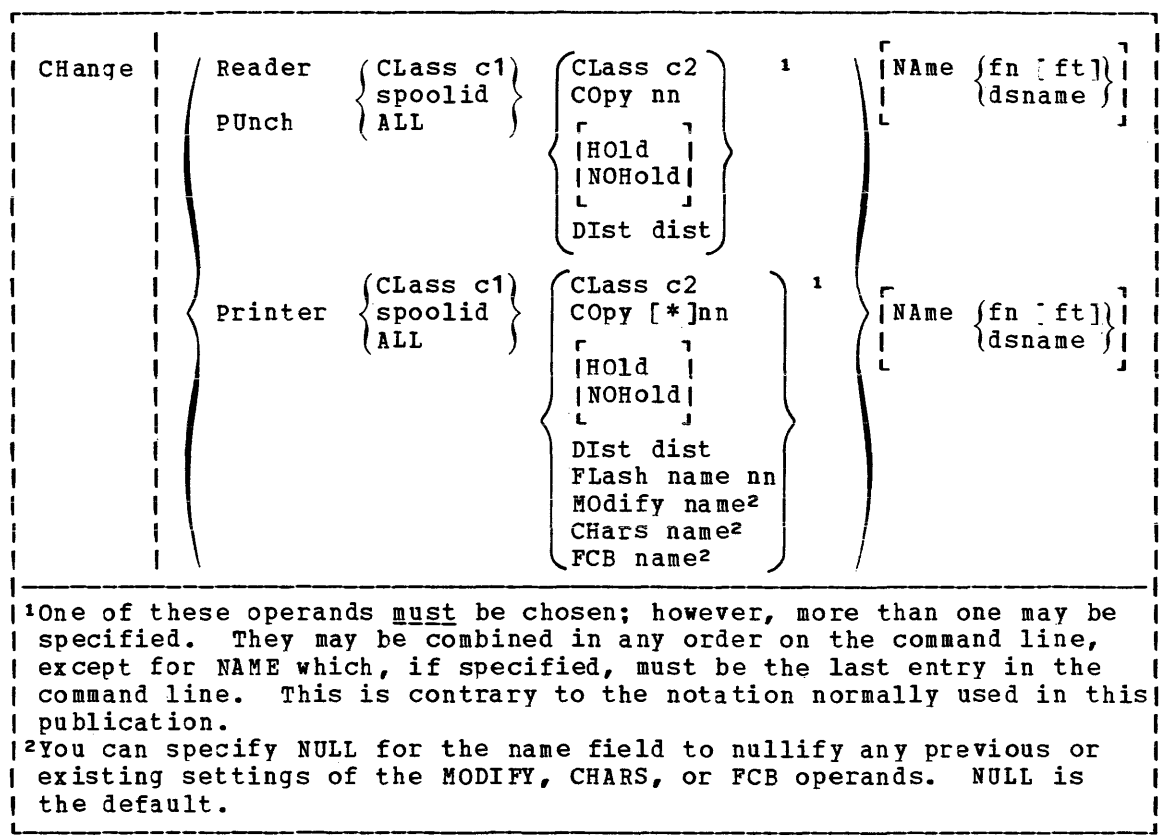
Responses

None. The virtual machine begins execution.

# CHANGE

Privilege Class: G

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




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

[2]You can specify NULL for the name field to nullify any previous or existing settings of the MODIFY, CHARS, or FCB operands. NULL is the default.

where:

READER
RDR — changes the reader spool file.

PUNCH
PCH — changes the punch spool file.

PRINTER
PRT — changes the printer spool file.

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

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

ALL — changes all your spool files.

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

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

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

NOHOLD releases the specified file from user HOLD status.

DIST dist
changes the distribution code specified in the VM/370 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.

FLASH name nn
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 nn can be anything from 0 to 99. The number specifies the first nn pages of output are to be superimposed. 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; it determines which writable character generation module (WCGM) is used for printing on the 3800.

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

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

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

Usage Notes

1. Issue the QUERY command to determine the current attributes of the file.

2. In order to change an output file, the file must have been closed but not yet selected for printing or punching.

3. An input (READER) file can be changed at any time before it is opened, that is, before the first read is issued for the file.

4. Use the QUERY command to verify the new attributes of the file or files.

5. If the spoolid operand is used and if more than one spool file with that number exists, then only the first spool file encountered is changed.

6. Unique to spool class X is a special line of text that can be printed on the separator page and at the top of each page of printer output. The default line internal to the system's program is a line of blanks. If you wish to use this spool class X facility, ask your system programmer to modify the text to your requirements.

7. In order to print on the 3800, the name the user specifies in the FLASH option must match the FLASH name the VM/370 operator specifies in the START command, unless the FLASH number nn is 0. If the number is 0 or the FLASH option is omitted, 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 a user fails to meet these requirements, the spool file is submitted to another printer and the special 3800 print options are ignored.

8. 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.

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/370 calculates the proper FCB form to load.

10. 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.

11. VM/370 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.

Response

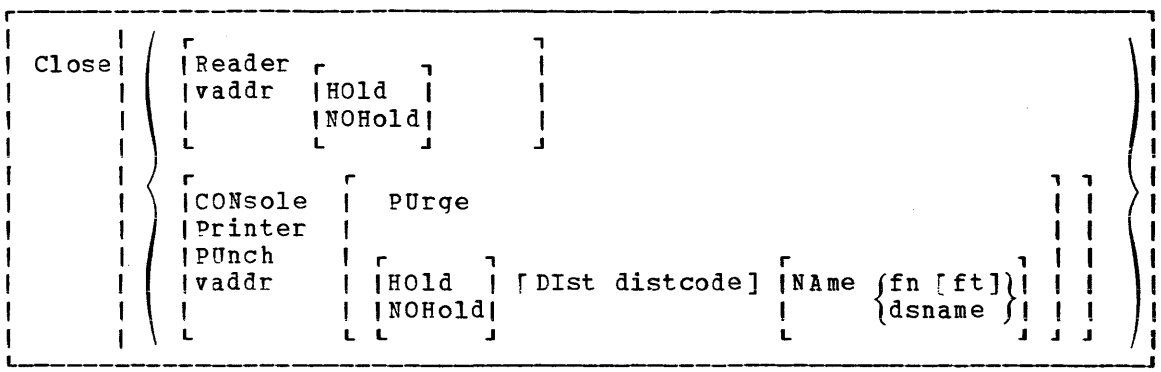
```
{nnnn} FILES CHANGED
{ NO }
```

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

# CLOSE

Privilege Class: G

Use the CLOSE command to terminate the spooling activity on any virtual spooled unit record or console device. The format of the CLOSE command is:




where:

READER
RDR — closes all reader spool files.

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

PRINTER
PRT — closes all printer spool files.

PUNCH
PCH — closes all punch spool files.

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

HOLD — makes the output spool file being closed unavailable for further processing, until it is specifically requested or changed. However, input reader files are still available for processing (see Figure 7). This option, specified in the CLOSE command, overrides any previously specified HOLD or NOHOLD option for the files being closed.

NOHOLD — makes the output spool file being closed available for further processing, but the input reader file is no longer available for processing (see Figure 7). Specify NOHOLD if a HOLD established by the SPOOL command is still in effect and the current active file is not to be held.

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

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

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

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

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

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

Usage Notes

1. If the file is an input reader file, the file being processed is purged unless SPOOL READER HOLD was previously specified (see the SPOOL command). The effect of HOLD or NOHOLD for a particular file established by the SPOOL command can be overridden by specifying NOHOLD or HOLD, respectively, in the CLOSE command.

2. If 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 7 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 7. Action Taken on CLOSE for a Virtual Input Device.

6. Figure 8 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] |

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

Figure 8. Action Taken on CLOSE for a Virtual Output Device

Responses

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

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

# 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.

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:

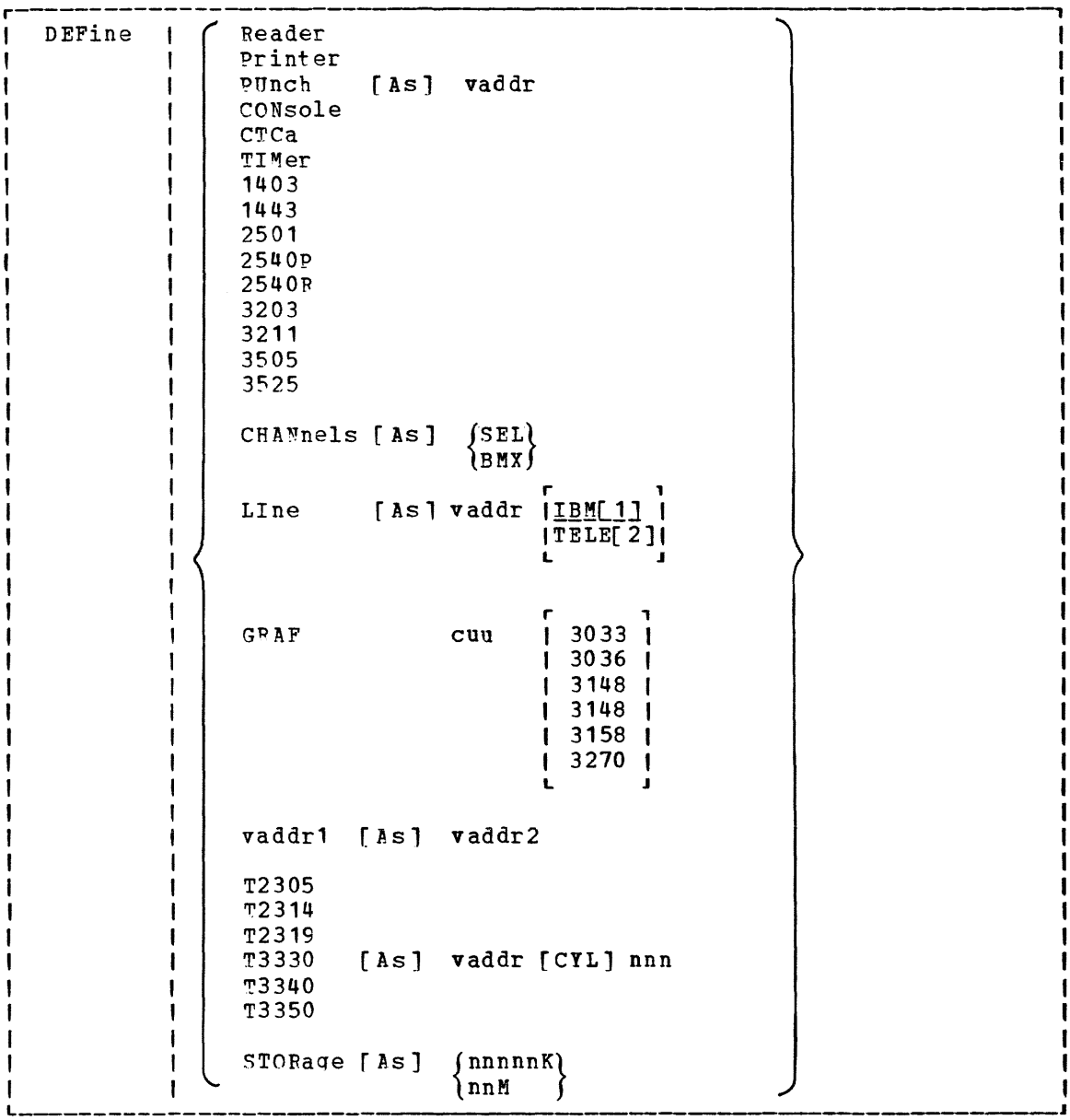
```
CP
```

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

# DEFINE

Privilege Class: G

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




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).

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

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

CHANNELS [AS] {SEL / BMX}

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

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

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

LINE [AS] vaddr [IBM[1] / TELE[2]]

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 / 3148 / 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.

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/370 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/370 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/370 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/370 so that VM/370 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/370, 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.

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

adds a temporary virtual disk of the specified type to the virtual machine configuration. The vaddr specifies the address of the disk, and must not be on a virtual control unit already defined as a CTCA. CYL nnn specifies the number of cylinders that the disk contains. Specify T3350 if a 3350 is used in native mode; specify T3330 if a 3350 is used in 3330 compatibility mode. Specify T3340 if a 3344 is used.

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. For the same reason, you should always clear temporary disk space before you either release it or log off. Data that is not erased before detaching the T-disk might be available to the next user assigned that same area in a subsequent DEFINE operation.

STORAGE [AS] {nnnnnK / nnM}

redefines the size of the virtual storage for the virtual machine as nnnnnK (where K represents 1024 bytes) or nnM (where M represents 1,048,576 bytes). The value specified becomes the new virtual storage size. Sizes must be in 4K increments and are limited by the maximum value in the VM/370 directory entry. The minimum size you can specify is 8K. All entries not specified in a 4K increment are rounded up to the next 4K boundary. Changing the virtual storage size (increasing or decreasing) causes a virtual system reset and clears all virtual storage to binary zeros.

Usage Notes

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

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

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

4. If your virtual machine is operating in the basic control mode (ECMODE OFF), virtual addresses specified in vaddr, vaddr1, and vaddr2 can be no higher than 5FF (channels 0 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 (cuu) in DEFINE statements in order to avoid a usage conflict caused by control unit I/O interface protocol. The following is an example of a virtual machine's DEFINE statements that can cause operational conflict:

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

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.

Responses

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

type vaddr DEFINED

where the possible values for type are:

| Type | Meaning |
|---|---|
| DASD | Direct access storage device/3330V(virtual volumes) |
| TAPE | Magnetic tape |
| LINE | Communication line |
| RDR | Card reader |
| PRT | Line printer |
| PUN | Card punch |
| GRAF | Graphics device |
| CONS | Console |
| 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).

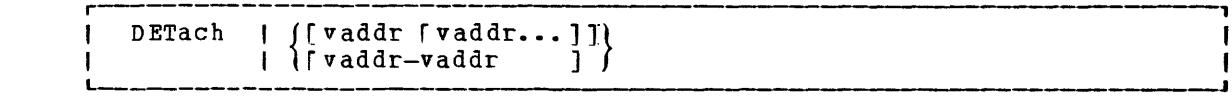
STORAGE = {nnnnnK / nnM}

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

# DETACH

Privilege Class: G

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




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/370 directory entries or during CP system initialization.

2. When you detach a virtual device, it becomes inaccessible to your virtual machine. If the device was previously attached to your virtual machine by an ATTACH command, it is released and becomes available for attachment to your virtual machine, another user, or the CP system. Tape devices are automatically rewound and unloaded when detached.

3. If you detach a device that was previously attached to your virtual machine by the operator, a message is sent to the operator informing him that the device is free.

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/370 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 / vaddr... / vaddr-vaddr} DETACHED

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

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

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.

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:

| Type | Meaning |
|---|---|
| DASD | Direct access storage device/3330V(virtual) volumes |
| TAPE | Magnetic tape |
| LINE | Communication line |
| RDR | Card reader |
| PRT | Line printer |
| PUN | Card punch |
| GRAF | Graphics device |
| CONS | Console |
| 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:

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/370 and the multiple-access virtual machine. See the VM/370 Terminal User's Guide for details on running and gaining access to multiple-access machines.

2. 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.

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

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

DROP FROM userid vaddr

is the message sent to the user when the line is disabled.

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

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

# DISCONN

Privilege Class: Any

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

```
| DISConn | [HOld] |
```

where:

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

Usage Notes

1. The terminal remains disconnected until it is reconnected via a LOGON command. The virtual machine is logged off 15 minutes after an attempt is made to read from the terminal or if the virtual machine goes into a disabled WAIT state. 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 TP control unit errors, CP places the virtual machine in disconnect mode for up to 15 minutes and your virtual machine does not continue to run 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.

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/370 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.

# DISPLAY

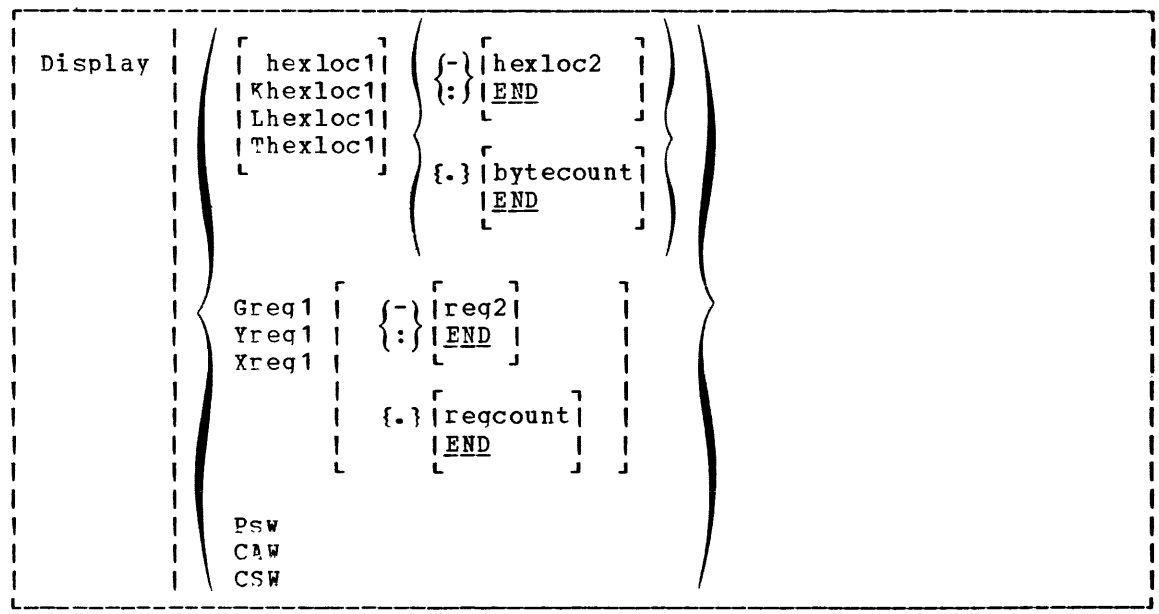
Privilege Class: G

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

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

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

The format of the DISPLAY command is:




where:

hexloc1
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 discontiguous saved segments.) The address, hexloc2, may be from one to six hexadecimal digits; leading zeros are optional.

{.}bytecount
END

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

Greg1

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

Yreg1

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

Xreg1

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

{-}reg2
{:}END

is a number representing the last register whose contents are to be displayed at the terminal. The dask (-) 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 (third level) storage of their own. This third-level 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. If a dash or colon is omitted between the hexloc1 and hexloc2 operands and they are separated by one or more blanks, only the storage contents at those two specific locations are displayed. Blanks are used to separate operands or sets of operands if more than one operand is entered on the same command line. Blanks must not be used to the right or left of the range or length delimiters (: or - or .), unless it is intended to take the default value of the missing operand defined by the blank. For example:

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

   displays the following, respectively:

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

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

5. The DISPLAY command does not distinguish between shared and nonshared storage; it displays any of the virtual machine's addressable storage whether shared or not.

6. Use the DISPLAY command to display the contents of various storage locations, registers, and control words at the terminal. By examining this type of information during the program's execution, you may be able to determine the cause of program errors. Usually, an address stop is set to stop the program execution at a specified point. The system enters the CP environment and you may then issue the DISPLAY command.

7. When you must examine large portions of storage, use the DUMP command rather than the DISPLAY command. Because the terminal operates at a much slower speed than the printer, only limited amounts of storage should be printed (via the DISPLAY command) at the terminal.

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

Responses

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

## Displaying Storage Locations

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

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

## Displaying Storage Keys

```
xxxxxx TO xxxxxx   KEY = kk
```

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

## Displaying General Registers

```
GPR n = genreg1 genreg2 genreg3 genreg4
```

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

## Displaying Floating-Point Registers

```
FPR n = xxxxxxxxxxxxxxxx  .xxxxxxxxxxxxxxxx  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

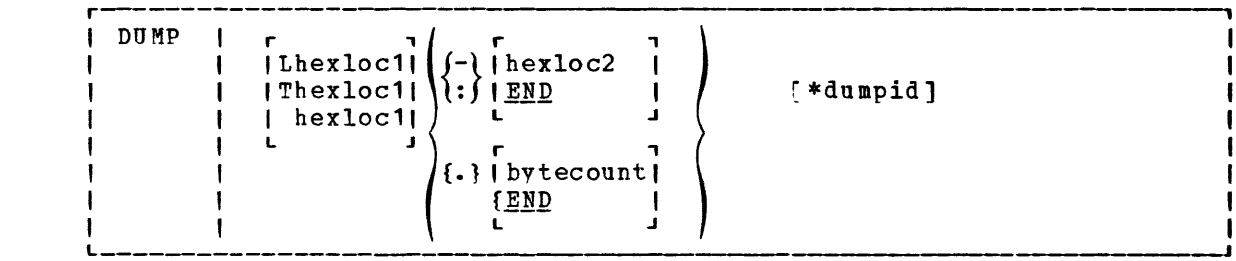
Privilege Class: G

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/370 directory entry)
- Storage keys
- Virtual storage locations (second-level storage only; see "Usage Notes").

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

The format of the DUMP command is:




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, but only 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 and OS/VS have virtual (third level) storage of their own. This third-level 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/370 directory entry, the control registers are also printed. To print the dump on the real printer, a CLOSE command

must be issued for the spooled virtual printer.

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. If a dash or a colon is not specified between the hexloc1 and hexloc2 operands and the two operands are separated by one or more blanks only the contents of the locations defined by the two operands is dumped. Blanks are used to separate operands or sets of operands if more than one operand is entered on the same command line. Blanks must not be used to the right or left of range or length delimiters ( : or - or .), unless it is intended to take the default value of the missing operand defined by the blank. Thus, all of the following produce full storage dumps:

```
dump 1      dump t:       dump 0-end
dump t      dump 1.       dump 1:end
dump -      dump t.       dump t:end
dump :      dump 0-       dump 0:end
dump .      dump 0:       dump 1.end
dump 1-     dump 0.       dump t.end
dump t-     dump 1-end    dump 0.end
dump 1:     dump t-end
```

The following produces three full dumps:

```
dump 1 . t
dump - . :
```

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

## Responses

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

DUMPING LOC hexloc

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

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

COMMAND COMPLETE

is the response indicating normal completion of the dump function.

# 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:

where:

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

Usage Notes

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

Responses

ECHO ENTERED; TO TERMINATE TEST, TYPE END

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

ENTER LINE

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

# EXTERNAL

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:

where:

code
40 is the interrupt code, a hexadecimal number to be associated with the external interrupt. All codes within the range X'01' through X'FF' are valid. If your virtual machine has the ECMODE option specified in the VM/370 directory, you can also code X'1004' (clock comparator interrupt) or X'1005' (CPU timer interrupt). The default is the external interrupt button on the system console, X'40'.

Usage Note

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

Responses

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

# 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:

where:

LOAD provides an indication of the operating load of VM/370 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%  RATIO-nn.n
```

where:

CPU-nnn%
: is a smoothed value that indicates the percentage of time that the main processor is executing.

APU-nnn%
: is a smoothed value that indicates the percentage of time that the attached processor is executing. In uniprocessor mode of operation, VM/370 indicates this response field as zero.

Q1-nn Q2-nn
: indicates the contention for CP as represented by smoothed values of 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. It is a smoothed 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.

RATIO-nn.n
: indicates the scheduler contention ratio; RATIO is a smoothed measure of the contention for real storage, and is defined as RATIO=(E+M)/M. M is the number of users in queue 1 and queue 2 (the runlist), and E is the number of users waiting to be allocated real storage by the scheduler and, therefore, temporarily resident in the scheduler's eligible lists. Thus, RATIO is the ratio of active users to users being serviced, and is 1.0 for optimum response. Optimum response occurs when enough storage is available to accommodate all active users, assuming the processor can process their commands. If E and M are both 0, the value of RATIO is set to 1.0. Values of RATIO=1.5 and M=10 mean that users are in the eligible lists waiting for the scheduler to allocate real storage to them, and that the full discrimination ability of the biased scheduler may be exercised upon them.

## Response for INDICATE USER

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

where:

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

WS-nnnn
: is the most recent system estimate of your working set size.

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

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

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

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

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

TTIME=mmm:ss is your total virtual machine time plus total processor time (virtual plus overhead) that you have used since you logged on or since the last ACNT command was issued for your virtual machine.

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

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

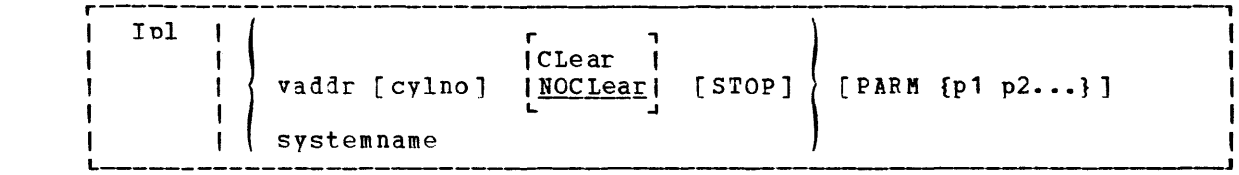
PRT-nnnnnn is the total number of virtual lines printed since you logged on or since the last ACNT command was issued for your virtual machine.

PCH-nnnnnn is the total number of virtual cards punched since you logged on or since the last ACNT command was issued for your virtual machine.

# IPL

Privilege Class: G

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




where:

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

simulates the IPL function when loading by device address.

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

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

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

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

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

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

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

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 CMSb
```

where b 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 computer console. The specified virtual address is accessed and the required input/output operations are performed to retrieve the IPL data.

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

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

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

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

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

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

6. 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 discontiguous segments (DCSS). However, the only discontiguous 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).

7. The BATCH option in the PARM operand indicates that the CMS initial program load is being performed in a batch instead of an interactive virtual machine. This option causes the CMS nucleus to issue the CMSBATCH command to load the batch processor. Specifying the BATCH option eliminates the need for operator intervention between jobs. Consult VM/370 CMS Command and Macro Reference for information on the CMSBATCH facility.

8. CMS users can use the PARM operand to pass the AUTOCR (automatic carriage return) option to the CMS initialization routine. AUTOCR simulates the pressing of the ENTER key as input to the virtual machine. This results in automatic execution of the user's PROFILE EXEC, if it exists on virtual minidisk 191.

Note: For a disconnected CMS service virtual machine, this mechanism can prove to be very useful. The name of a CMS restart routine may be specified in the CMS nucleus. In the event of an ABEND, the specified restart routine receives control instead of the service virtual machine entering a disabled wait state. If this routine issues an IPL CMS PARM AUTOCR command, the PROFILE EXEC executes initial setup procedures that allow the service virtual machine to reestablish itself.

## Responses

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

# LINK

Privilege Class: G

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

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

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

where:

[TO] userid
: is the name of the user whose VM/370 directory is to be searched for device vaddr1. An asterisk (*) is used to specify that the device is in your own VM/370 directory. If the keyword TO is omitted, the userid may not be "TO" or "T".

vaddr1
: is the virtual device address (cuu) in the VM/370 directory for that userid.

[AS] vaddr2
: is the virtual address (cuu) 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 access. The link is not done if any other user has the disk in write status. R is the default mode if the link is to another userid. |
| RR | Read-only access. The link is established even if another user has the disk in write status. |
| W | Write access. The link is not done if any other user has the disk in read or write status. |
| WR | Write and read access. If another user has the disk in read or write status, an alternate access of read-only is acceptable. |
| M | Multiple access. This means that a write-link is to be given to the disk unless another user already has write access to it, in which case no link is to be done. |

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 link to a device that is being updated by a user who has the device in read/write 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/370 directory for your disks.

PASS= password

is a one- to eight-character string that must match the access mode password for device vaddr1 in the VM/370 directory for the user (userid) specified. The password should be specified only when the LINK is executed by a virtual machine (for example, from CMS), since 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.

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

Usage Notes

1. If you link to one of your own disks, no password is required. Also, if the link is to a device whose password is ALL, meaning that the device can be used by all users, the password is not required. However, if the link is to any other userid, a password for the desired device must be provided. 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/370 directory for the disk being linked. The third row indicates whether the disk is already being used, and if so, the mode of its access. The last row indicates the type of link established. For example, the third column is interpreted as follows: if you request a read access link (R) to a disk that has a read password defined and that already is accessed in read mode, you can establish a read link.

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:
**MMMMMMMM**

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

ENTER WRITE PASSWORD:
**MMMMMMMM**

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

ENTER MULT PASSWORD:
**MMMMMMMM**

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 threshhold value for attempts is exceeded, an installation userid receives a message. There is another system threshhold 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,

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

and that no other users are linked to the same disk.

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

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

(nnn USERS) is replaced by 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

Privilege Class: G

Use the LOADVFCB command to specify the forms control buffer image for a virtual spooled 3203 or 3211 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 or 3211 printer.

FCB — is a required reserved keyword meaning Forms Control Buffer.

name — is a system-defined name for the 3203 or 3211 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/370 System Programmer's Guide for a discussion of the FCB macro and forms control images.

The 3203 does not support the index feature and 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 or 32111 printer. The virtual machine's VM/370 directory entry must indicate a 3203 or 3211, even though the program and operating system have a 1403 printer defined. Then the LOADVFCB command can be used to specify a virtual FCB image for 1403 printers so that programs that use printer overflow sensing may be spooled to disk.

2. If no virtual FCB image has been previously loaded for a spooled, virtual 3203 or 3211 printer, the following default virtual FCB image is used:

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

3. VM/370 provides one FCB image (FCB1) with the following format:

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

4. Care should be taken to load a virtual 3203 or 3211 printer with an FCB that is compatible with an FCB used in a real 3203 or 3211. Failure to do so can produce unpredictable results.

Responses

None.

# LOGOFF

Privilege Class: Any

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

```
| LOGoff | [HOld]
| LOGout |
```

where:

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

Usage Notes

1. This command causes all active spool files to be closed, temporary disks to be relinquished, dedicated devices to be detached, and an accounting record to be created for the user.

2. You should always log off at 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/370 discovers that the terminal has been turned off (that is, VM/370 attempts to send a message to the terminal, but gets back an error code indicating that the terminal is turned off). Because many hours may pass before VM/370 discovers that the terminal is turned off, you run the risk of compromising the security of the virtual machine and data files. Anyone turning the 3270 power back on has access to the virtual machine without logging on. This is because the machine is still logged on, although inactive.

Responses

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

where:

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

VIRTCPU= 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 / DEV rid} LOGOFF AS userid USERS = nnn

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

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

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

```
┌──────────────────────────────────────────────────────────────────────┐
│ Logon  │ userid  [password¹] [Mask] [Noipl]                          │
│ Login  │                                                             │
├──────────────────────────────────────────────────────────────────────┤
│¹Refer to "Note" under description of password operand.               │
└──────────────────────────────────────────────────────────────────────┘
```

where:

userid  is the identifier assigned to you in the VM/370 system.

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

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.

MASK  types masking characters to cover your password on typewriter terminals without the print inhibit feature. The mask types on the line following a prompting message from VM/370 requesting you to enter your password. Should you forget to ask for masking when you type LOGON, you can press the carriage return after the prompt for the password types, and VM/370 then types out the masking characters.

NOIPL  specifies that the IPL device or name in the VM/370 directory should not be used for an automatic IPL.

Usage Notes

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

2. If you use LOGON because a teleprocessing line or terminal error disconnected you from your virtual machine, you have 15 minutes to log on again. If you do not log on within 15 minutes, your virtual machine automatically logs off. In this case, you may have to reconstruct files and restart jobs interrupted by the teleprocessing line or terminal error.

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. After a reconnect, CP resets the linedel character to the one specified in the directory (the default value). Therefore, if you issue the SET or TERMINAL command with more than one operand and you specify an operand that is reset following a disconnect, re-issue the same SET or TERMINAL command again after you are reconnected. 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/370 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 the LOGON command is disabled. The default is four incorrect logon passwords.

Responses

ENTER PASSWORD:

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

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

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

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

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

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

-- or --

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

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

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

-- or --

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

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

# MESSAGE

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 by 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/370, LOGON or LOGN and the line number are displayed instead of userid.

hh:mm:ss

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

# NOTREADY

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:

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.

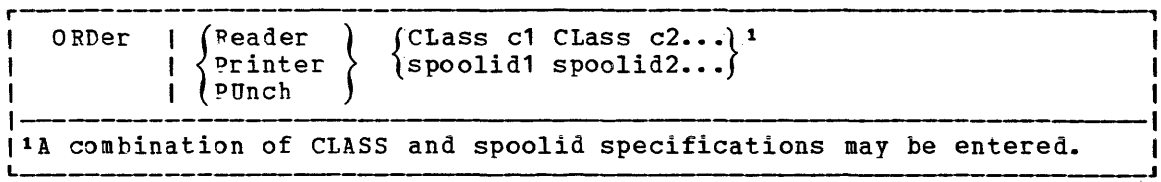
Response

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:




where:

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

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

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

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

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

Usage Notes

1. You may order only your own files.

2. Use the QUERY command to determine spool file attributes such as class and spoolid for files you want to rearrange.

3. A combination of class and spoolid specifications can be used to rearrange files. For example:

   order printer class A 1963 class C

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

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

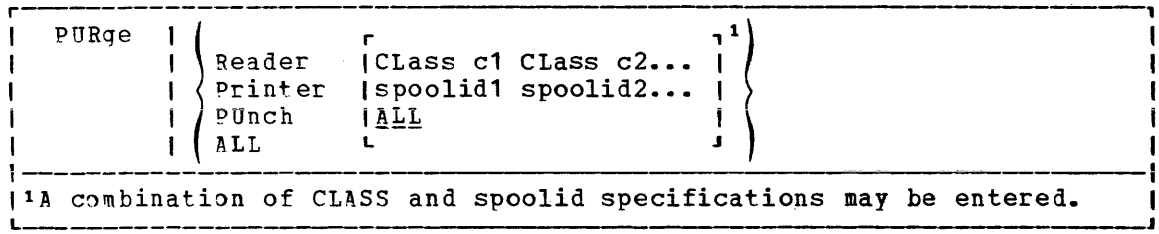
{nnnn} FILES ORDERED
{NO  }

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

# PURGE

Privilege Class: G

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




where:

READER
RDR
: purges reader files.

PRINTER
PRT
: purges printer spool files.

PUNCH
PCH
: purges punch spool files.

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

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

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

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

Usage Notes

1. Any closed spool file, owned by you, may be purged, regardless of its status, as long as it has not been selected for processing.

2. Purging can be done using a combination of CLASS and spoolid specifications. For example:

   purge printer class A 1932 class D 619

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

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

Response

{nnnn} FILES PURGED
{NO  }

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

# QUERY

Privilege Class: G and all classes except class Any

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

- The time you have used during a terminal session.
- The number of closed input and output spool files associated with your virtual machine.
- The current settings of the SET command functions.
- The current settings of the TERMINAL command functions.
- The status of all the devices on your virtual machine.
- The channel operating mode of your virtual machine, whether block-multiplexer or selector.
- A listing of all users who are linked to a given virtual address, together with their device addresses and access modes.
- Identification and attributes associated with your virtual printer, punch, and reader spool files.
- The identification of your virtual processor.
- The mode of processor operation for your virtual machine whether uniprocessor mode (UP) or attached processor mode (AP).

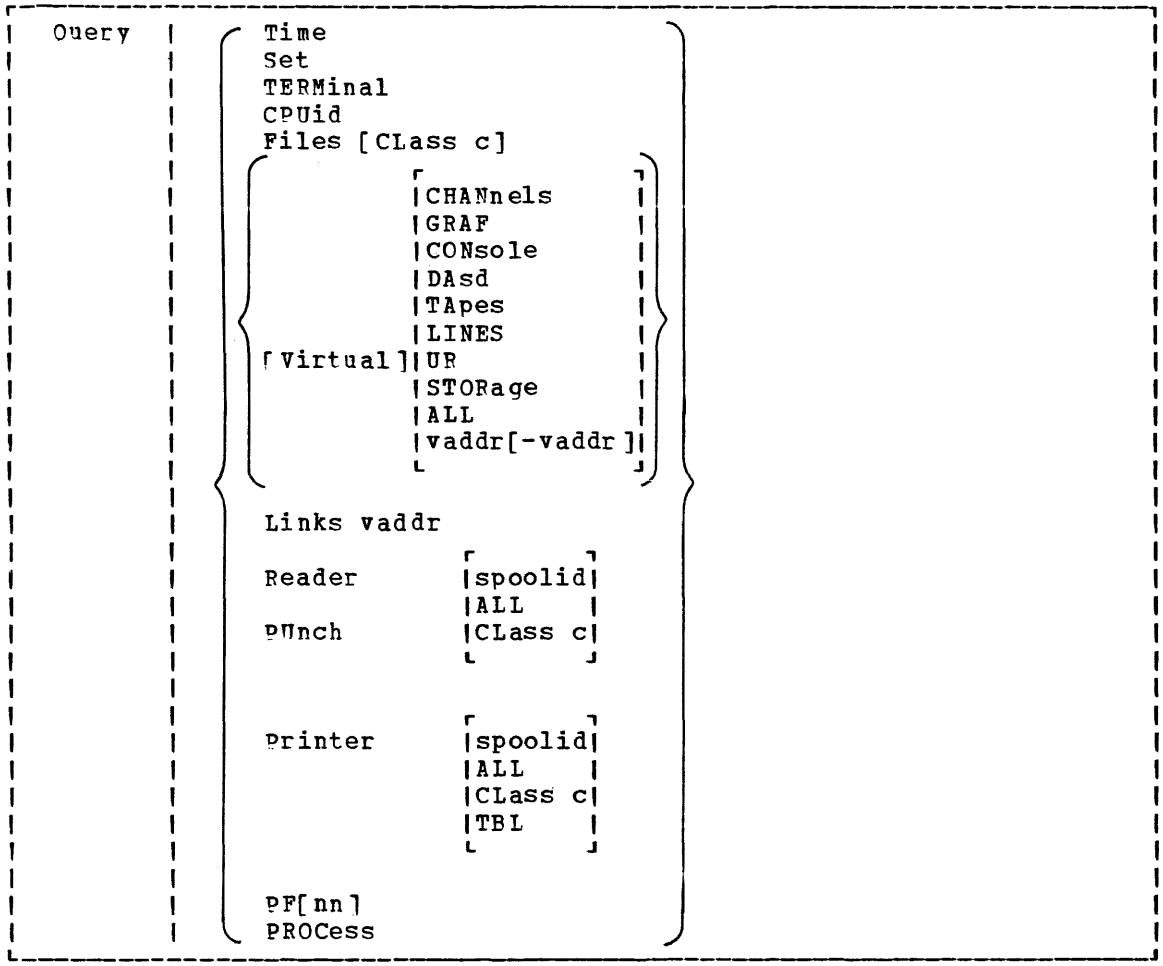
There are other operands you can use with the QUERY command if you have the privilege class required to use them. These are described in the VM/370 Operator's Guide. Also, if you are a CMS user, you can use the CMS QUERY command to query the status of your CMS virtual machine.

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

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

# QUERY Command for Class G Users

The format of the Class G QUERY command is:




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 [CLASS c]
displays the number of spooled input and output files for your virtual machine. Files currently being processed are not included in the totals. If CLASS is specified, the number of spooled input and output files of the class specified is displayed.

VIRTUAL displays the status of all virtual devices.

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

GRAF displays the status of all your virtual display devices that are locally attached.

CONSOLE displays the status of your virtual consoles.

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

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

LINES displays the status of all your virtual communication lines.

UR displays the status of all your unit record devices.

STORAGE displays the size of your virtual storage.

ALL displays the status of all your virtual devices.

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

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.

READER
RDR
PRINTER
PRT
PUNCH
PCH
displays the following information, pertaining to your closed virtual reader, virtual printer, and virtual punch spool files:

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

One line of information is displayed for each spool file.

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

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

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

{READER / PRINTER / PUNCH} 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 20 characters of the 24-character field are displayed.
- Distribution code of the file (PRINTER and PUNCH files only)

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

Note: For information on how to alter spool file attributes, see the CHANGE command discription in this section.

{READER / PRINTER / PUNCH} CLASS c

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

PRINTER TBL

displays additional information about print files spooled to the 3800 printer. This option must follow the PRINTER operand. In addition to the information normally displayed for print files, 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

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

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

PROCESS displays for the user which processors are currently online to VM/370.

## QUERY Command for Class G Users

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

### QUERY TIME

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

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

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

The time spent in the current terminal session is displayed.

where:

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

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

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

### QUERY SET

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

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

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

     {ON }           {NONE}          {ON }        {ON }
IMSG {OFF}, AFFINITY {nn  }, NOTRANS {OFF}, SMSG {OFF}
```

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

## QUERY TERMINAL

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

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

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: The TABCHAR option of the QUERY TERMINAL command appears on the 3278 Model 2A console screen.

## QUERY CPUID

```
CPUID: aabbbbbbccccdddd
```

where:

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

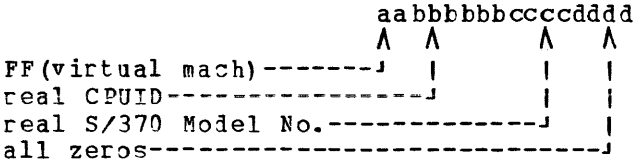
aa
: is the version code; these two digits are forced to X'FF' to identify that the virtual machine is running under VM/370.

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/370 directory's OPTION control statement.

cccc
: is the model number; this field contains a leftmost digit of 0 followed by three digits of the model number (0-9). This field defaults to the model number of the real machine.

dddd
: is the machine check extended logout; this field is forced to X'0000' since CP does not reflect machine checks back to the virtual machine.

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



QUERY FILES [CLASS c]

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

The total number of spool files in your system is displayed. If you specify the CLASS option with QUERY FILES, only the totals for the class you specify are indicated rather than for all classes on your system.

QUERY VIRTUAL CHANNELS

```
CHANNELS= {SEL}
          {BYT}
```

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

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

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

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

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

where:

vaddr is the virtual address of the virtual machine console.

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

c is the spooling class of the console.

nn is the number of copies spooled.

userid is the user identification.

distcode is the distribution code.

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

The default settings for a virtual console are:

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

## QUERY VIRTUAL DASD

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

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

where:

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

type is an IBM direct access device.

Note: A display of 3340 represents either a 3340 or 3344 device.

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

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

nnn is the number of cylinders on the virtual disk.

## QUERY VIRTUAL TAPES

TAPE vaddr ON DEV raddr

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

where:

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

raddr is the real address of the tape.

## QUERY VIRTUAL LINES

```
LINE vaddr ON DEV raddr
```

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

where:

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

raddr is the real address of the line.

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

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

## QUERY VIRTUAL UR

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

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

where:

vaddr is the virtual device address of the virtual reader.

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

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

The default settings for a reader are:

```
RDR vaddr CL * NOCONT NOHOLD READY EOF
```

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

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

```
                 {  CONT} {  HOLD}           {   READY}
PRT vaddr CL c   {NOCONT} {NOHOLD} CPY [*]nn {NOTREADY}

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

vaddr FLASH ovly  CHAR ctab  MDFY cmod  FCB pcpi
```

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

where:

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

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

[*]nn is the number of copies from 1 to 99 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.

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

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

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

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

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

Note: The distcode in this case indicates the FOR userid; however, the distcode produced on the output files when the file is closed is the distcode assigned to the FOR userid as specified in the VM/370 directory.

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

The default settings are:

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

PRT vaddr CL A  NOCONT  NOHOLD  COPY 01  READY
    vaddr  FOR  userid  DIST distcode  FLASHC 00
    vaddr FLASH     CHAR     MDFY     FCB
```

where:

userid and distcode are assigned for the virtual machine.

QUERY VIRTUAL STORAGE

STORAGE = nnnnnK

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

QUERY VIRTUAL ALL

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

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

QUERY VIRTUAL vaddr

QUERY VIRTUAL vaddr-vaddr

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

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

```
DASD vaddr ON DASD vaddr
```

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

```
MSC vaddr ON DEV vaddr
```

QUERY LINKS vaddr

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

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

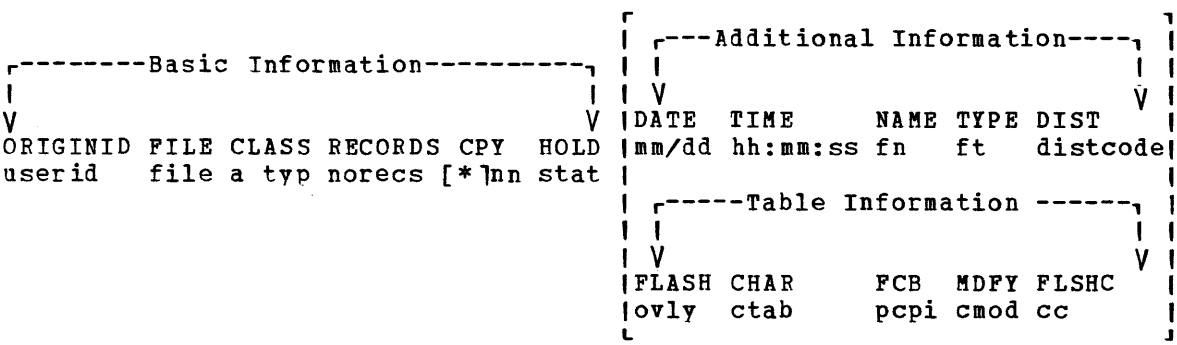
where:

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

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

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

QUERY READER, QUERY PRINTER, QUERY PUNCH




where:

userid — is the user who originally created the file.

file — is a unique, system-assigned number which is used by VM/370 to identify the file.

a — is the spool file class.

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

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

[*]nn — is the number of copies assigned to the file (it has no effect for virtual reader files). 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).

Note: For further information on altering spool file attributes, refer to the CHANGE command in Section 5.

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

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

hh:mm:ss — is the time the file was opened for creation in hours:minutes:seconds. 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 20 characters are displayed. These characters extend from the "name" field through the "type" field.

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

distcode is the distribution code assigned to the file.

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.

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

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.

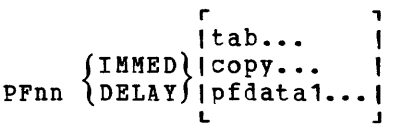
The information listing DATE, TIME, NAME, TYPE, and DIST (date of file creation, time of file creation, filename, filetype and file distribution code) is displayed only when you specify the ALL or spoolid operands. However, if you specify the TBL operand, the information listing FLASH, CHARS, FCB, MDFY, and FLSHC (3800 overlay form, 3800 character arrangement table, 3800 vertical formatting, 3800 copy modifications, and the number of copies to be printed by 3800 with overlay form) 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 commands SPOOL READER HOLD and CLOSE READER 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).

QUERY PFnn




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 PROCESS

PROCESSOR mm ONLINE[,PROCESSOR nn ONLINE]

The first processor displayed is the main processor. The second processor displayed is the attached processor and this part of the response only occurs if VM/370 is running in AP mode. VM/370 is running in uniprocessor mode when the bracketed part of the response is omitted.

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

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

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

The format for this QUERY command is:

```
┌──────────────────────────────────────────────────────────────────────┐
|  Query  | (LOGmsg            )                                        |
|         | (Names             )                                        |
|         | (Users   [userid]  )                                        |
|         | (userid            )                                        |
└──────────────────────────────────────────────────────────────────────┘
```

where:

LOGMSG displays the log messages of the day.

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

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

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

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

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

QUERY LOGMSG

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

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

QUERY NAMES

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

A list of all logged-on users is displayed; if the user is currently connected, the real address (raddr), or the resource identification of a 3704/3705 line (resid) to which he is connected is displayed; if he is not connected to the system, DSC is displayed.

QUERY USERS

```
nnn USERS, mmm DIALED
```

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

where:

nnn is the total number of logged-on users.

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

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

QUERY userid or QUERY USERS userid

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

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

# READY

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:

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 I/O is in progress at the console, or if other interrupts are pending. This condition may occur if the user issues the REQUEST command using DIAGNOSE X'08'.

Responses

None.

# RESET

Privilege Class: G

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

```
| RESET | vaddr |
```

where:

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

## Usage Notes

1. In addition to clearing all pending interrupts, all error conditions occurring as a result of unit checks and virtual sense bytes are reset.
2. When the DIAL command is issued from a local 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:

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.

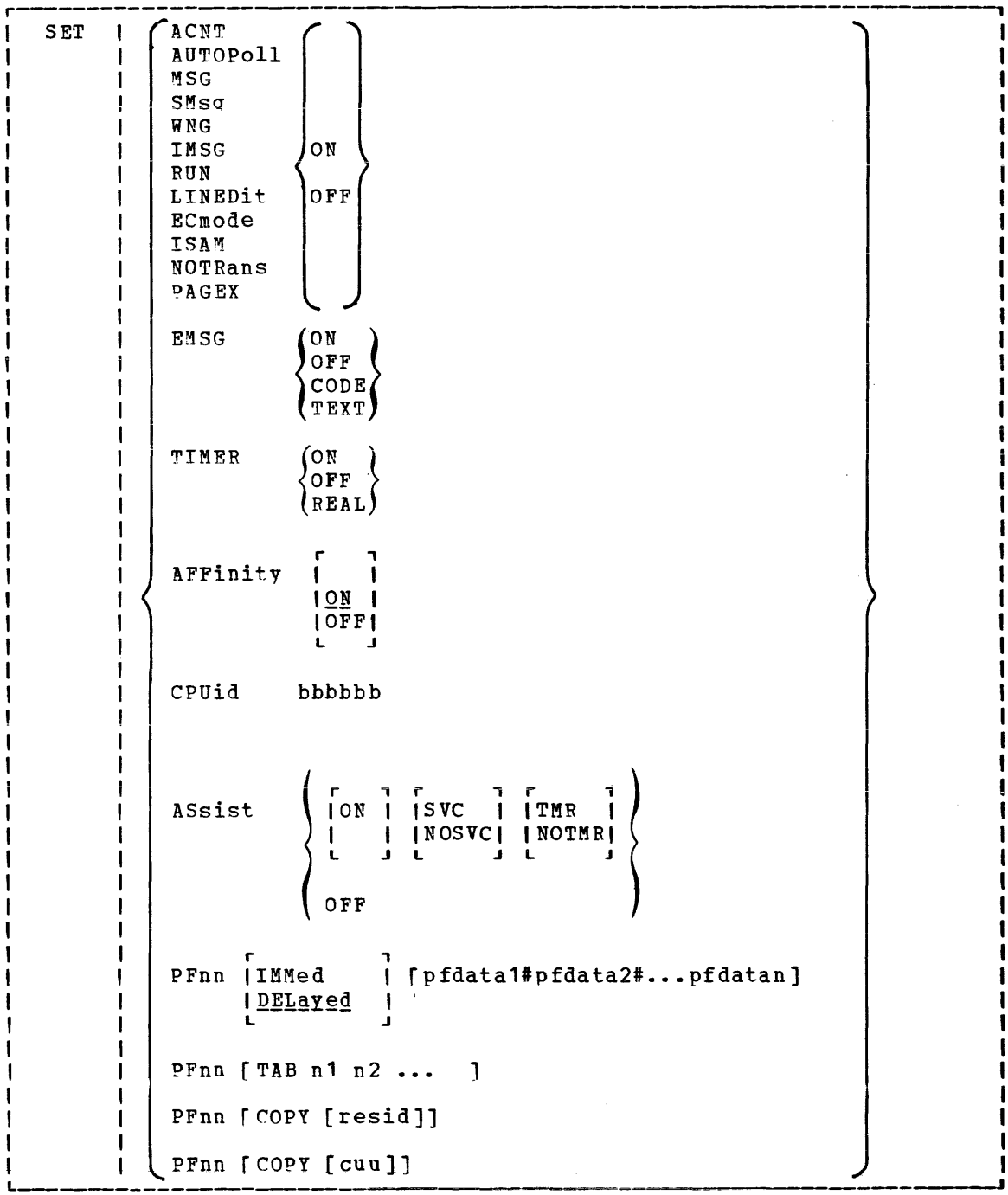
REWIND NOT PERFORMED

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

# SET

Privilege Class: G

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



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/370, 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/370, MSG is set on.

SMSG {ON / OFF}

controls whether a virtual machine is receiving special messages or not. If OFF is specified, the virtual machine cannot receive special msssages.

WNG {ON / OFF}

controls whether warning messages are displayed at the terminal. If ON is specified, all warning messages sent via the CP WARNING command from the system operator are received at the terminal. If OFF is specified, no warning messages are received. When you log on VM/370, WNG is set on.

IMSG {ON / OFF}

controls whether certain informational responses issued by the CP CHANGE, DEFINE, DETACH, 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/370, IMSG is set on.

RUN {ON | OFF}

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

LINEDIT {ON | OFF}

controls the line editing functions. ON specifies that the line editing functions and the symbols of the VM/370 system are to be used to edit virtual 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/370, LINEDIT is set on.

ECMODE {ON | OFF}

controls whether the virtual machine operating system may use System/370 extended control mode and control registers 1 through 15. Control register 0 may be used with ECMODE either ON or OFF. When you log on VM/370, ECMODE is set according to the user's directory option; ON if ECMODE was specified and OFF if not.

Note: Execution of the SET ECMODE {ON|OFF} command always causes a virtual system reset, but does not change the PSW ECMODE bit.

ISAM {ON | OFF}

controls whether additional checking is performed on virtual I/O requests to DASD in order to support the OS Indexed Sequential Access Method (ISAM). When you log on VM/370, 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/370, 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/370, EMSG is set to TEXT.

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

Note: CMS recognizes EMSG settings for all error (E), information (I), and warning (W) messages, but ignores the EMSG setting and displays the complete message (error code and text) for all response (R), severe error (S), and terminal (T) messages. Also, data typed in such as a list of names following a message, will not be controlled by the EMSG setting.

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/370 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 or the attached processor of an attached processor 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 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/370 to ensure that all virtual machine using the SET CPUID command have specified unique CPU serials. The hexadecimal field 'bbbbbb' is the CPU identification number (CPU serial).

Note:

The CPU identification number (serial) is only a portion of the complete CPUID. The CPUID identification stored in response to a STIDP instruction is a string of 16 hexadecimal digits as follows:

aabbbbbbccccdddd

where

aa - version code; these two digits are forced to X'FF' to identify that the virtual machine is running under VM/370.

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/370 hardware assist. VM/370 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/370. In attached processor systems, VM/370 hardware assist is available for installation on either or both processors.

All, some, or none of the function performed by VM/370 hardware assist may be available on the processor or attached processor that the virtual machine is logged on. (For the extent of VM/370 hardware assist available on supported VM/370 processors, refer to the VM/370 Operator's Guide.) The extent of VM/370 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/370 system operator permits the associated hardware assist function to be active for the entire system (or in attached processor systems, the appropriate processor). Overall system control of hardware assist is controlled by SET SASSIST and SET CPASSIST commands; see the VM/370 Operator's Guide for details on these commands.

If VM/370 hardware assist is active for the entire system, then hardware assist is active and used during virtual machine operations immediately after logon. Hardware assist as it pertains to a virtual machine can then be turned off and on again by SET ASSIST OFF and SET ASSIST ON respectively.

The SVC handling portion of the hardware assist is invoked when you log on unless your VM/370 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/370 is handling SVC interruptions. All SVC 76 requests are passed to CP for handling, regardless of the SVC and NOSVC operands. If you issue the SET ASSIST command line and specify SVC or NOSVC while the 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/370 handles all the SVC interruptions.

If the user requests VM/370 hardware assist and it is available and active, the request is honored; if VM/370 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 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/370 System Programmer's Guide for additional information on how to use VM/370 hardware assist.

```
      ┌      ┐
PFnn  |IMMED  | [pfdata1#pfdata2#...pfdatan]
      |DELAYED|
      └      ┘
```

defines a program function for a program function key on a 3270 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/370 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 12 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/370 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#...pfdatan
: defines the VM/370 command or data lines that constitute the program function. If more than one command line is to be entered, the pound sign (#) must separate the lines. If you use the pound sign (#) to separate commands that you want executed with the designated PF key, you must precede the command line with #CP, turn line editing off, or precede each pound sign with the logical escape character ("). For further explanation, see the "Usage Notes" section that follows. If no command lines are entered following the PFnn designation, then the program function is marked undefined. Program functions cannot be embedded within one another.

PFnn [TAB n1 n2 ...]
: specifies a program function key number to be associated with tab settings on a terminal. The number of the PF key, nn, can be a value from 1 (or 01) to 12. 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/370 CMS User's Guide. TAB is a keyword identifying the tab function. The tab settings (n1 n2 ...) may be entered in any sequence. Note: When defining the TAB

function, the existence of blanks preceding the keyword TAB or the lack of trailing tab settings forces the pf key to be treated as pfdata.

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

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 device is not a 3284/3286/3287/3288/3289 printer.
   - The SET PFnn COPY command format is invalid.

2. You may include your own identification on the printed output by entering the data into the user input area of the screen before you press the PF key. The identification appears on the last two lines of the printed copy.

3. To ensure the definition of a pf key for the COPY function, you can either start with one leading blank before the keyword COPY or follow the keyword with at least five blanks.

4. The existence of more than one data fileid (cuu or resid) following the keyword COPY, identifies the string as pfdata rather than a hardcopy request.

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 keyword is ignored. The program function still executes immediately.

4. In the case of systems with attached processors, the VM/370 hardware assists may be installed on none, one, or both processors. Similarly, hardware assist, if equipped on both processors of an attached processor 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/370 hardware assist is installed and active on his processor. In attached processor setups where the user has no processor preference (AFFINITY=NONE) upon entering SET ASSIST he too receives no response if either processor has VM/370 hardware assist active. Any error message is indicative that VM/370 hardware assist is not available or active for user. If the user receives a message indicating that VM/370 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/370 hardware, need only change his affinity setting, then initiate the SET ASSIST command.

5. 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:

6. The SET RUN ON command should be issued before the user issues an ADSTOP command.

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:

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

              -- or --

cp set pf02 immed q rdr"#q prt"#q pun

              -- or --

set pf02 immed q rdr"#q prt"#q pun
```

and press the ENTER key.

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

Example 2

Enter the following command while in CMS mode:

```
set linedit off
```

and press the ENTER key.

Then enter:

```
set pf02 immed q rdr#q prt#q pun

            -- or --

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

and press the ENTER key.

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

Then enter:

```
set linedit on
```

and press the ENTER key.

Pressing the PF02 key executes the three QUERY commands.

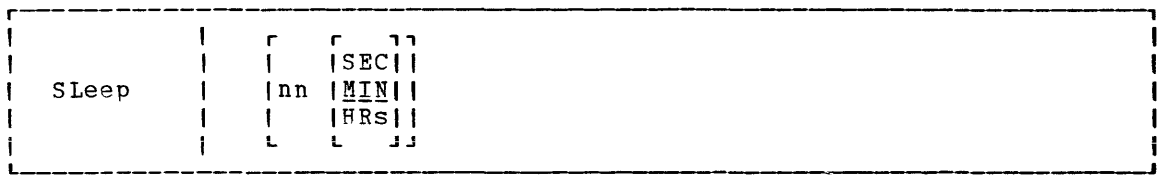
Responses

None.

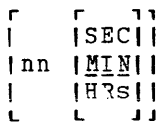
# SLEEP

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:




where:




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:

```
smsg pd58gh1c        special message
14 characters used   125 characters left

sm u                 special message
5 characters used    134 characters left
```

Note: Although the 3270 display unit contains 139 characters in its input area, CP only reads 134 characters of data. When the SMSG command is issued using DIAGNOSE X'08', only 132 characters are read by CP.

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/370.

   Note: Refer to VM/370 System Programmer's Guide for more information on the AUTHORIZE function of VMCF.

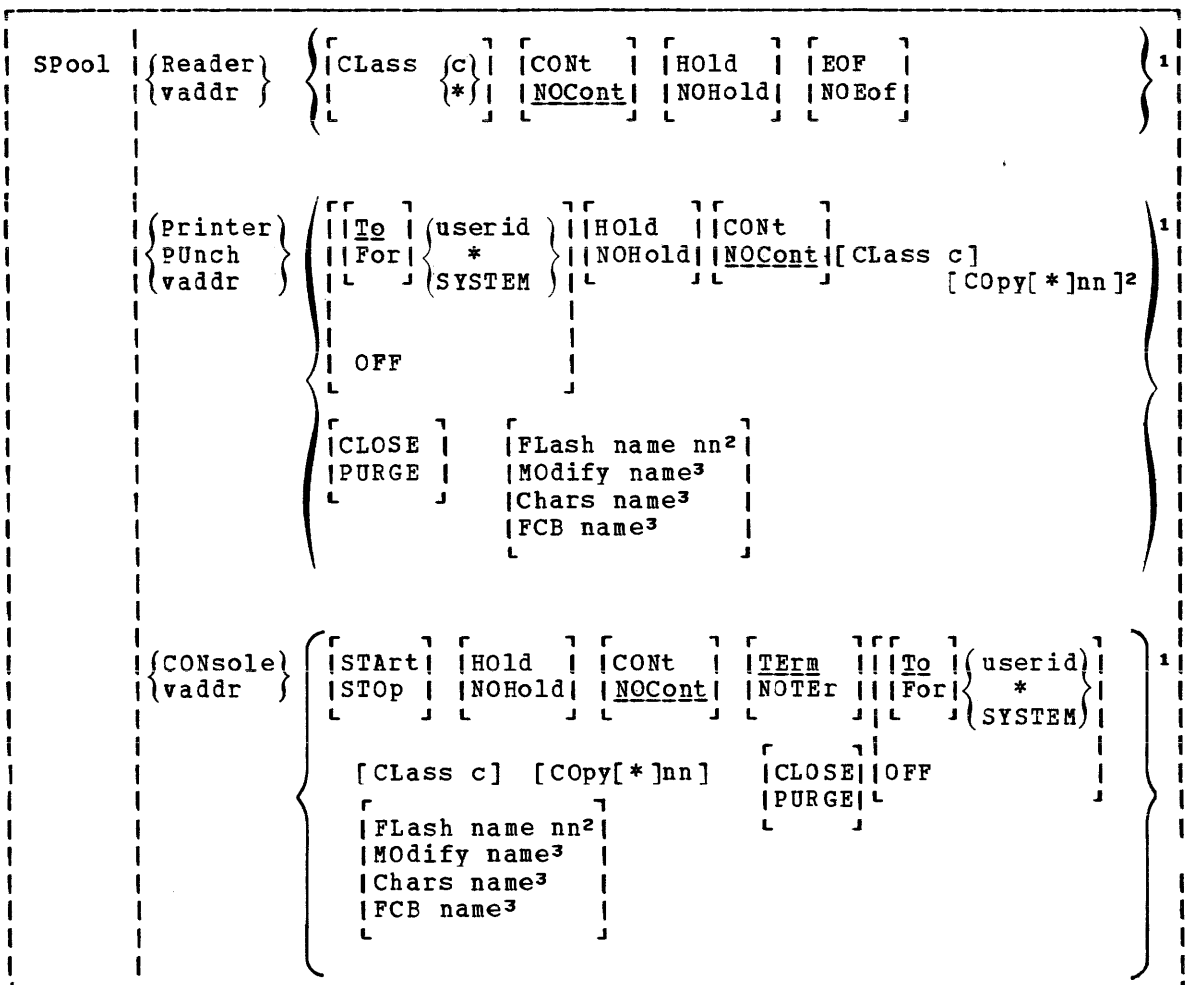
Responses

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 this command appear under "Usage Notes" following the operand descriptions. The format of the SPOOL command is:




1 At least one of the options within braces must be selected; however, more than one may be specified, and they may be entered in any order on the command line.
2 These options can only be used to modify a virtual spooling printer Furthermore, these options only apply to a 3800 printer as a virtual spooling device.
3 You can specify NULL for the name field to nullify any previous or existing settings of the MODIFY, CHARS, or FCB operands. NULL is the default.

where:

READER
RDR — modifies the options for all reader spool files.

PRINTER
PRT — modifies the options for all printer spool files.

PUNCH
PCH — modifies the options for all punch spool devices.

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

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

CLASS {c / *} — specifies the spool class of the device. c is a one-character alphameric field whose values can be A through Z, or 0 through 9.

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

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

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.

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.

$$\left[\underline{\text{TO}}\right] \left\{\begin{matrix}\text{userid}\\ *\\ \text{SYSTEM}\end{matrix}\right\}$$

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/370 directory. The file is not transferred to the user's reader input. The default setting is for your own virtual machine identification. FOR *, or FOR SYSTEM can be coded to specify your own identification and is equivalent to the OFF option.

OFF
: resets the transferred spool option.

COPY[*]nn
: is the number of copies from 1 to 99 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.

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

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

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

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

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

FLASH name nn
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 (nn) can be anything from 0 to 99. The number (nn) specifies the number of copies of output to be superimposed. 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; it determines which writable character generation modules (WCGMs) are used for printing on the 3800.

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 a warning
- it is the result of Commands entered on a 3270 in CMS EDIT mode.
- it is the result of a CP command entered by means of the #CP command.
- 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.

## 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 01 |
| Printer | NOHOLD NOCONT CLASS A[1] OFF COPY 01 FLASHC 00 |
| Console | NOHOLD NOCONT CLASS T[1] OFF COPY 01 TERM |

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

## 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/370 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/370 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/370 spooling commands such as ORDER, CHANGE or CLOSE.

Spoolids are assigned to all your spool files sequentially. When the maximum number (9900) is assigned, CP begins again with the number 1.

When you print or punch a file, CP displays at your terminal the spoolid it assigned to your file if (1) multiple copies of the file are being processed, (2) the file is being transferred to another user, or (3) the file is placed in a user hold status. You can find out various kinds of information about a file using the spoolid with the many forms of the QUERY command.

## Transmitting Spool Files Locally

Spool files can be transferred between users via the SPOOL and TRANSFER commands. To transfer printer or punch files of your virtual machine to the virtual reader of some other user, issue the SPOOL command with the TO option:

```
spool device to userid
```

where device is the virtual device type or virtual address (for example, PRINTER, PUNCH, 00E) and userid is the name of the designated user's reader.

If you only want to print or punch a file for someone else, you can issue the SPOOL command with the FOR option:

```
spool device for userid
```

where device is the same as the previous example. The userid provides the distribution code for the designated user.

To transfer printer or punch files from your virtual card reader to another user's reader, issue the TRANSFER command with the TO option:

```
transfer option to userid
```

where option is SPOOLID, CLASS, or ALL and userid is the designated user's virtual reader.

If you merely wish to reclaim a transferred file, you can issue the same command with the FROM option:

```
transfer option from userid
```

where option and userid are the same as the previous example.

Notes:

1. Files that have been spooled for another user are not in that user's virtual reader and, therefore, you cannot reclaim them.
2. You can transfer any file that is queued on your virtual readers; however, you can reclaim only those files which you originated.

## Transmitting Files to Remote Locations

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

```
tag dev device locid
```

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

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

```
spool device to userid
```

where device is the same virtual device type or virtual device address specified in the TAG command and userid is the userid of the RSCS virtual machine at your installation. You can find out the userid of your installation's RSCS virtual machine and the locid for the various remote stations from your installations'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/370 spool file) and transmits it across the RSCS network.

## Receiving Files from the RSCS Network

If your virtual machine is logged on VM/370, RSCS notifies you of the arrival of a file for your machine from the RSCS network by displaying a message at your terminal. The file is sent to your virtual card reader.

VM/370 can accumulate files from the RSCS network destined for your virtual card reader, regardless of whether you are logged on your virtual machine or not. If you are logged on your virtual machine, issue the QUERY command to see if you have any files in your virtual reader. When you log on your virtual machine, the logon process transmits a message informing you of accumulated spool files residing in your virtual reader (punch or printer).

## Responses

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

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

is received if (1) multiple copies of the file are being processed, (2) the file is being transferred to another user, or (3) the file is placed in a user hold status. However, if SET MSG OFF is in effect, this message is not issued, even though the above conditions are met.

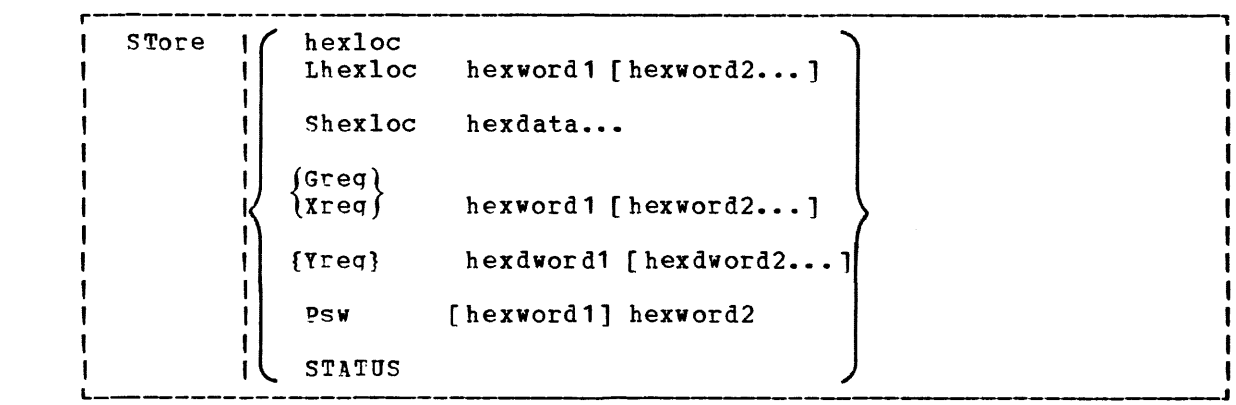
# STORE

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

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




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.

Greg hexword1 [hexword2...]
: stores the hexadecimal data (hexword1 [hexword2...]) in successive general registers starting at the register specified by reg. The reg operand must be either a decimal number from 0 through 15 or a hexadecimal digit from 0 through F. The hexword1 [hexword2...]operand each represents up to eight hexadecimal digits. If the value being stored is less than a fullword (eight hexadecimal digits), it is right-adjusted in the word and the high order bytes of the word are filled with zeros. If two or more hexwords are specified, they must be separated by one or more blanks.

Xreg hexword1 [hexword2...]
: stores the hexadecimal data (hexword1 [hexword2...]) in successive control registers starting at the register specified by reg. The reg operand must either be a decimal number from 0 through 15 or a hexadecimal digit from 0 through F. If the virtual machine is in basic control mode, you can store data in register 0 only. The hexword1 [hexword2...] operand each represents up to eight hexadecimal digits. If the value being stored is less than a fullword (eight hexadecimal digits), it is right-adjusted in the word and the high order bytes of the word are filled with zeros. If two or more hexwords are specified, they must be separated by one or more blanks.

Yreg hexdword1 [hexdword2...]
: stores the hexadecimal data (hexdword1 [hexdword2...]) in successive floating-point registers starting at the register specified by reg. The reg operand must be a digit from 0 through 7. If reg is an odd number, it is adjusted to the preceding even number. The hexdword1 [hexdword2...] operand each represents up to 16 hexadecimal digits. If the value being stored is less than a doubleword (16 hexadecimal digits), it is left-justified in the doubleword and low order positions are filled with zeros. If two more more hexdwords are specified, they must be separated by one or more blanks.

PSW [hexword1] hexword2
: stores the hexadecimal data in the first and second words of the virtual machine's program status word (PSW). If only hexword2 is specified, it is stored into the second word of the PSW. The [hexword1] hexword2 operand each represents up to eight hexadecimal digits. These operands must be separated by one or more blanks. If the value being stored is less than a fullword (eight hexadecimal digits), it is right-adjusted in the word and the high order bytes of the word are filled with zeros.

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

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

## Usage Notes

1. First level storage is real main storage. Only second-level storage (storage that is real to the virtual machine) can be stored into. Operating systems running in a virtual machine such as DOS/VS and OS/VS have virtual (third level) storage of their own. This third-level 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 CP commands.

## Response

STORE COMPLETE

is the response at the successful completion of the command.

# SYSTEM

Privilege Class: G

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

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 that is all zeroes except for the wait bit which is set on.

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

Responses

STORAGE CLEARED - SYSTEM RESET

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

SYSTEM RESET

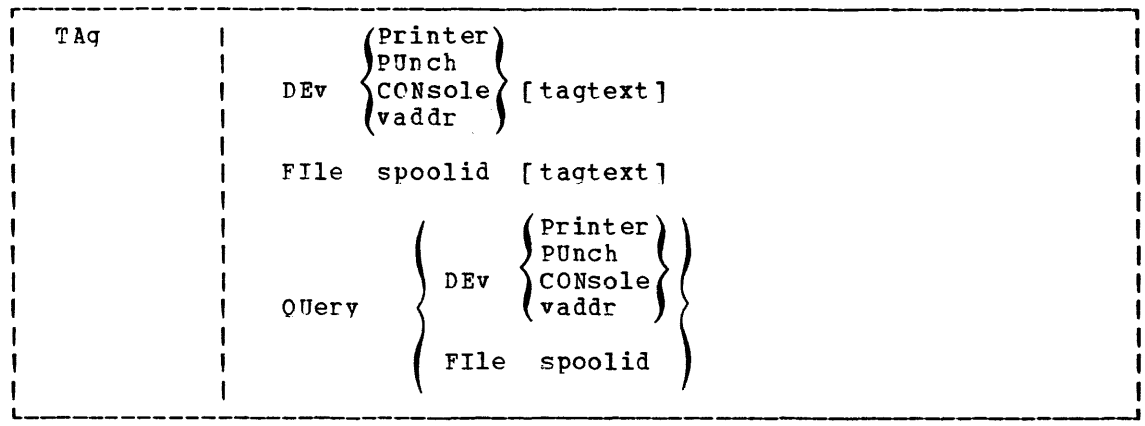
is the response given if the command SYSTEM RESET is entered.

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

# TAG

Privilege Class: G

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




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 only be used for reader spool files that are queued on your virtual reader.

  The spoolid operand is the spool file identification, a number between 1 and 9900 and assigned by CP when the spool file was closed.

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

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

     FILE  spoolid
```

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

## Usage Notes

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

   locid [userid] [priority]

   where:

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

   userid — is the userid of the VM/370 virtual machine (a one- to eight-character user identification) to which a file is being transmitted. This operand is used by remote stations when they transmit files to the RSCS virtual machine and want the files sent to a particular VM/370 virtual machine. You can ignore this operand if you are not specifying a priority. However, if you are specifying a priority, you must code some userid operand; it is ignored by RSCS.

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

2. If you enter the TAG command with no tagtext information, the tag area associated with the device or file is set to all blanks.

3. A spool file that you have created and spooled to another virtual machine is owned by the other user as soon as the file is closed. If you want to query or change the tag data, you must first reclaim the file via the TRANSFER command.

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

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

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

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

TAG QUERY DEV vaddr

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

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

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

TAG QUERY FILE spoolid

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

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

(TAG BLANK)
          if the tag is all blanks

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

# 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  }  }[1]
          |  { LINEDel   {OFf }  }
          |  { LINENd    {char}  }
          |  { EScape    {    }  }
          |  { TABchar[2]         }
          |  {                    }
          |  { Mask      {    }  }
          |  { APL       {ON  }  }
          |  { TEXT      {OFf }  }
          |  { ATtn      {    }  }
          |  {                    }
          |  { MODE      {CP  }  }
          |  {           {VM  }  }
          |  {                    }
          |  { LINESize   nnn     }
```

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

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

[2] The TABCHAR operand is available on the 3278 Model 2A console.

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/370 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/370 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/370 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/370 directory entry. If OFF is specified, no logical escape character is allowed. If char is specified, then that character becomes the logical escape character. The character chosen should be unique and not common to the data stream being entered. When you log on, ESCAPE ON is in effect; if your virtual machine is logged on by the system operator via the AUTOLOG command, ESCAPE OFF is the default.

TABCHAR {ON | OFF | char}

allows a user to define a variety of logical tab characters on graphics devices. The 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.

MASK {ON / OFF}

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

APL {ON / 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 only for 3277 terminals.
2. If the TERMINAL APL ON command is issued from a terminal that is not equipped with APL keys, or if the virtual machine is not running APL when the command is issued, the results are unpredictable.

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

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.

Note: This operand is valid only for 3277 terminals.

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.

MODE $\left\{\begin{matrix}CP\\VM\end{matrix}\right\}$

controls the terminal attention environment. CP specifies that one or more attentions force the virtual machine into the CP environment. VM specifies that one attention is reflected to your virtual machine and that more than one attention forces your virtual machine into the CP environment. VM is the default for all VM/370 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.

## 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 (MASK, TEXT, APL, ATTN, and LINESIZE) are reset if you log on after disconnecting.

2. Although you can define line-editing symbols and status with the TERMINAL command, the LINEDIT operand of the SET command determines whether the VM/370 line-editing functions are on or off.

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

4. You 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.

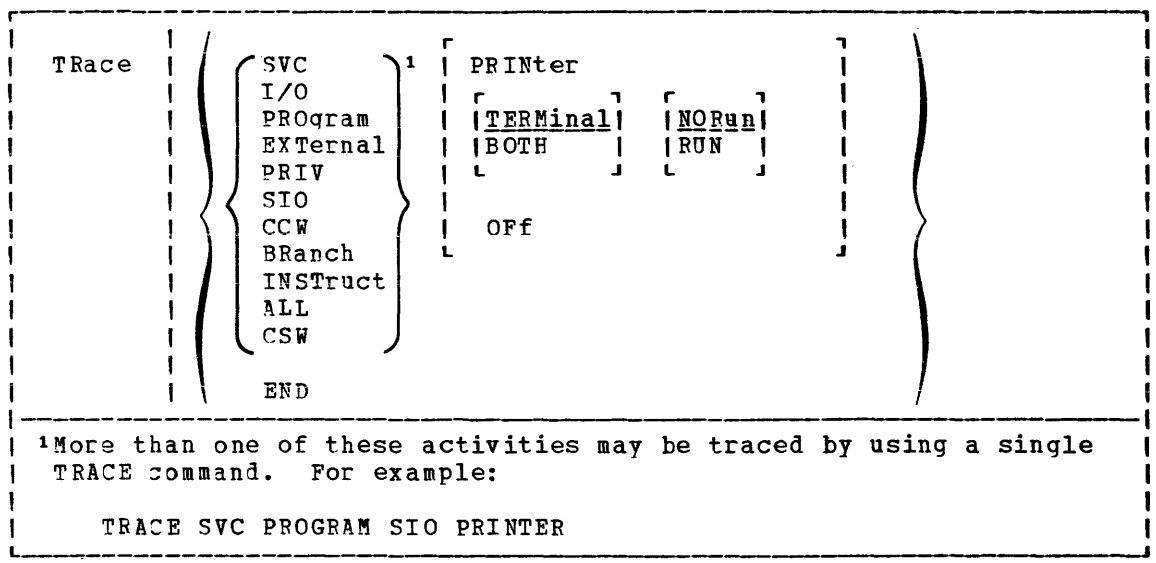
## Responses

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/370 hardware assist is enabled on your virtual machine, CP turns it off while tracing SVC, PRIV, BRANCH, INSTRUCT, or ALL activities. If virtual interval timer support is available on your processor and is active (that is, SET ASSIST TMR is in effect), CP turns the timer support off while tracing external interrupts. After the tracing is terminated with the TRACE END command, CP turns hardware assist on again.

3. If trace output is being recorded at the terminal, the virtual machine stops execution and CP command mode is entered after each output message. This simulates the instruction step function.

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

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

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

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

4. If trace output is being recorded on a virtual spooled printer, a CLOSE command must be issued to that printer in order for the trace output to be printed on the real printer.

5. Successful branches to the next sequential instruction and branch-to-self instructions are not detected by TRACE.

6. RUN and NORUN can be specified in different TRACE commands with no conflict as long as the functions being traced are different.

7. Instructions that modify or examine the first two bytes of the next sequential instruction cause erroneous processing for BRANCH and INSTRUCT tracing.

8. When tracing on a virtual machine with only one printer, the trace data is intermixed with other data sent to the virtual printer. To separate trace information from other data, define another printer with a lower virtual address than the previously defined printer. For example, on a system with 00E defined as the only printer, define a second printer as 00B. The regular output goes to 00E and the trace output goes to 00B.

9. If the BRANCH, INSTRUCT, or ALL activities are being traced by a virtual machine using a shared system, the user receives a warning message and the virtual machine is unshared from (given a private copy of) the named system. Other users of the segment continue to operate with an unchanged copy of that segment.

10. I/O operations for virtual channel-to-channel 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 could 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:

| Symbol | Meaning |
|---|---|
| vvvvvv | virtual storage address |
| tttttt | virtual transfer address or new PSW address |
| rrrrrr | real storage address |
| xxxxxxxx | virtual instruction, channel command word, CSW status |
| yyyyyyyy | real instruction, CCW |
| ss | argument byte (SSM-byte) for SSM instruction |
| ns | new system mask after execution of STOSM/STNSM |
| zz | low order byte of R1 register in an execute instruction (not shown if R1 register is register 0) |
| zzzzzzzz | referenced data |
| type | virtual device name (DASD, TAPE, LINE, CONS, RDR, PRT, PUN, GRAF, DEV) |
| V vadd | virtual device address |
| R radd | real device address |
| mnem | mnemonic for instruction |
| int | interruption type (SVC, PROG, EXT, I/O) |
| code | interruption code number (in hexadecimal) |
| CC n | condition-code number (0, 1, 2, or 3) |
| IDAL | Indirect data address list |
| *** | virtual machine interrupt |
| ::: | privileged operations |
| ==> | transfer of control |
| SEEK | read/write position on DASD device |

TRACE STARTED

This response is issued when tracing is initiated.

TRACE ENDED

This response is issued when tracing is suspended.

TCH, TIO, CLRIO, HIO, HDV, SIO, or SIOF

TCH

```
I/O vvvvvv TCH xxxxxxxx type vadd CC n
```

TIO, CLRIO, HIO, or HDV

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

SIO or SIOF

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

CCW:

```
CCW vvvvvv xxxxxxxx xxxxxxxx rrrrrr yyyyyyyy yyyyyyyy
CCW  IDAL  vvvvvvvv vvvvvvvv  IDAL  00rrrrrr 00rrrrrr
CCW  SEEK  xxxxxxxx xxxxxx    SEEK  yyyyyyyy yyyy
```

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

INSTRUCTION TRACING:

Privileged Instruction:

```
::: vvvvvv SSM   xxxxxxxx ss           (normal SSM)
::: vvvvvv SSM   xxxxxxxx ss  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
```

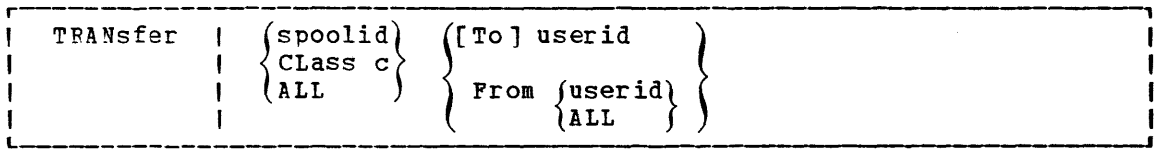
Entry for 'branch to' instruction

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

# TRANSFER

Privilege Class: G

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




where:

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

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

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

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

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

## Usage Note

The TRANSFER command does not transfer any active spool files.

## Responses

RDR FILE spoolid TRANSFERRED {TO / FROM} userid

where:

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

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

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

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

{nnnn / NO} FILES TRANSFERRED

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.

# 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/370 creates for the virtual machine user). See Usage Note 5. 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:

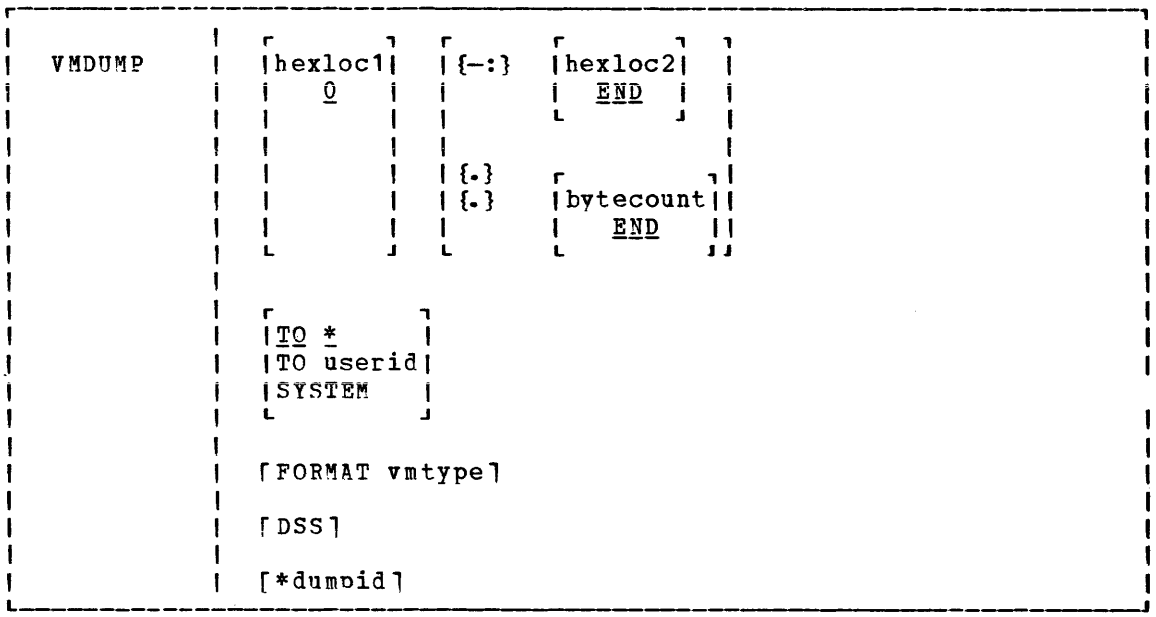
- Virtual program status word (PSW)
- General registers
- Floating point registers
- Control registers
- Storage protection keys
- Virtual machine type identification
- Timer values

VMDUMP also provides the user 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, a user may write routines that perform this function. For more information on the VM/IPCS Extension, refer to VM/370 Interactive Problem Control System Extension (VM/IPCS Extension) Program Product: User's Guide and Reference, Order No. SC34-2019.

Except for the *dumpid operand, the user can specify the operands of the VMDUMP command in any order. The *dumpid operand must be the last operand specified on the command line.

The format of the VMDUMP command is:




where:

hexloc1
0
: is the first or only hexadecimal virtual storage address dumped. If the user omits the hexloc1 operand, the default is zero, the beginning of virtual machine storage. The user may also specify the hexloc2 or bytecount operands with the hexloc1 operand when he knows the dump size he wants. If the user specifies hexloc2 with hexloc1, CP takes a dump of the storage contents within the range of addresses between hexloc1 and hexloc2. For example, when the user enters:

```
vmdump 1000-2000
     or
vmdump 1000:2000
```

CP dumps the contents of virtual machine storage at the hexadecimal addresses between 1000 and 2000. If the user specifies the bytecount operand with hexloc1, hexloc1 specifies the beginning address of the dump and the hexadecimal number of bytes dumped. If the user enters:

```
vmdump 1000.8A
```

CP dumps the contents of virtual machine storage beginning at hexadecimal address 1000 for a total of 8A bytes. The user can specify a maximum of three different areas of virtual storage that he wants dumped each time he issues the VMDUMP command. See Usage Notes 1, 2, and 4.

{-:}hexloc2
END

is the last hexadecimal virtual storage address dumped. If the user does 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. The user 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 the user enters:

```
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. The user may specify up to a maximum of three different virtual machine storage areas he wishes to dump each time he issues the VMDUMP command. The user 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 the user does 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. The user must specify a period (.) delimiter if he wants to dump the contents of more than one storage location by bytecount. When the user enters:

```
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 2 for an expanded explanation of delimiter use.

TO userid
TO *

transfers the dump to the virtual card reader of the userid specified. If the user enters an asterisk after TO, CP transfers the dump to the user's own 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. The user must not specify TO preceding the keyword SYSTEM.

FORMAT vmtype

provides VM/IPCS Extension with the virtual machine type (vmtype) which VM/IPCS Extension uses to format the dump. The vmtype keyword is a one-to-eight byte name of the operating system running in a virtual machine (for example, CMS). CP also uses the specified vmtype as the virtual card reader filetype. CP does not validity check the vmtype. Any vmtype longer than eight bytes generates an error message and halts further VMDUMP processing. The dump header record includes the user specified vmtype and the VM/IPCS Extension uses the vmtype information to format the dump. If the user enters FORMAT, he must also specify a vmtype. If he does not specify FORMAT, the default vmtype is FILE. See Usage Note 3.

DSS — specifies that CP take a dump of all discontiguous saved segments in use by the user's virtual machine.

*dumpid — is a line of user input up to 100 characters long including imbedded blanks and asterisks which the user can enter for his 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 the user specifies this operand, it becomes the DMPDMPID field in the dump file information record (DMPINREC) data area. If specified, the user must enter *dumpid as the last operand on the VMDUMP command line.

Usage Notes:

1. 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.

2. Due to 4K boundary rounding for the hexloc1 operand, additional dumps of up to 4095 more bytes than the user specified can occur.

3. If the user enters the FORMAT operand followed by another valid keyword of the VMDUMP command (for example, SYSTEM), CP uses the keyword following FORMAT as the vmtype and CP ignores the normal meaning of the keyword following FORMAT.

4. If the user omits a dash or a colon between the hexloc1 and hexloc2 operands, and he separates the operands with one or more blanks, CP dumps only the storage contents at those two specific hexadecimal addresses. The user must insert blanks to separate operands or sets of operands if he specifies more than one operand on the command line. The user must not insert blanks to the left or right of the range or length delimiters (-, :, or .) unless he intends to use the default value of the missing operand that the blank represents.

5. First level storage is real main storage. CP, however, dumps only second level storage (that is, storage that VM/370 creates for the guest virtual machine.) Operating systems running in a guest virtual machine such as OS/VS and DOS/VS, 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.

Responses

COMMAND COMPLETE

is the response indicating normal completion of the VMDUMP function.

# Appendix. VM/370 Command Summary

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

Figure 10 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 date. |
| ADSTOP | G | Halt execution at a specific virtual machine instruction address. |
| ATTACH | B | Attach a real device to a virtual machine. |
| | B | Attach a DASD device for CP control. |
| | B | Dedicate all devices on a particular channel to a virtual machine. |
| ATTN | G | Make an attention interruption pending for the virtual machine console. |
| AUTOLOG | A,B | Automatically log on a virtual machine and have it operate in disconnect mode. |
| BACKSPAC | D | Restart or reposition the output of a unit record spooling device. |
| BEGIN | G | Continue or resume execution of the virtual machine at either a specific storage location or at the address in the current PSW. |
| CHANGE | D,G | Alter one or more attributes of a closed spool file. |
| CLOSE | G | Terminate spooling operations on a virtual card reader, punch, printer, or console. |
| COUPLE | G | Connect channel-to-channel adapters. |
| CP | any | Execute a CP command while remaining in the CMS virtual machine environment. |
| DCP | C,E | Display real storage at terminal. |
| DEFINE | B | Redefine the usage of SYSVIRT and VIRTUAL 3330V devices. |
| | G | Reconfigure your virtual machine. |

Figure 10. CP Command Summary (Part 1 of 4)

| Command | Privilege Class | Usage |
|---|---|---|
| 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 | 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. |
| LOADBUF | D | Load real UCS/UCSB or FCB printer buffers. |

Figure 10. CP Command Summary (Part 2 of 4)

| Command | Privilege Class | Usage |
|---|---|---|
| LOADVFCB | G | Load virtual forms control buffer for a virtual 3203 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. |
| 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. |
| READY | G | Simulate device end interruption for a virtual device. |
| REPEAT | D | Repeat (a specified number of times) printing or punching of a specific real spool output file. |
| REQUEST | G | Make an attention interruption pending for the virtual machine console. |
| RESET | G | Clear and reset all pending interruptions for a specified virtual device and reset all error conditions. |
| REWIND | G | Rewind (to load point) a tape and ready a tape unit. |
| SAVESYS | E | Save virtual machine storage contents, registers, and PSW. |
| SET | A,B,F,G | Operator--establish system parameters.<br>User--control various functions within the virtual machine. |

Figure 10. CP Command Summary (Part 3 of 4)

| Command | Privilege Class | Usage |
|---|---|---|
| SHUTDOWN | A | Terminate all VM/370 functions and checkpoint CP system for warm start. |
| SLEEP | any | Place virtual machine in dormant state. |
| SMSG | G | Send Special Messages to specified virtual machine. |
| SPACE | D | Force single spacing on printer. |
| SPOOL | G | Alter spooling control options; direct a file to another virtual machine or to a remote location via the RSCS virtual machine. |
| START | D | Start spooling device after draining or changing output classes. |
| STCP | C | Change the contents of real storage. |
| STORE | G | Alter specified virtual storage locations and registers. |
| SYSTEM | G | Simulate RESET, CLEAR STORAGE, and RESTART buttons on a real system console. |
| TAG | G | Specify variable information to be associated with a spool file or output unit record device.<br>Interrogate the current TAG text setting of a given spool file or output unit record device. |
| TERMINAL | G | Define or redefine the input and attention handling characteristics of your virtual console. |
| TRACE | G | Trace specified virtual machine activity at your terminal, spooled printer, or both. |
| TRANSFER | D,G | Transfer input files to or reclaim input files from a specified user's virtual card reader. |
| UNLOCK | A | Unlock previously locked page frames. |
| VARY | B | Mark a device unavailable or available. |
| 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 10. CP Command Summary (Part 4 of 4)

Figure 11 contains an alphabetical list of the CMS commands with a brief statement about the use of each command. Unless otherwise noted under Ref. Code, the CMS commands are described in the VM/370 CMS Command and Macro Reference.

Figure 12 contains an alphabetical list of the CMS commands used by CMS system programmer. These commands are described in the VM/370 System Programmer's Guide.

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

| Ref Code | Meaning |
|---|---|
| SYSGEN | Described in the VM/370 Planning and System Generation Guide. |
| OS PP | Invokes an OS Program Product, available from IBM for a license fee. |
| EREP | Described in the VM/370 OLTSEP and Error Recording Guide with references to OS/VS, DOS/VSE, VM/370 Environmental Recording Editing and Printing (EREP) Program. |
| Op Gd | Described in the VM/370 Operator's Guide. |
| SCRIPT | Invokes a text processor that is an IBM Installed User Program, available from IBM for a license fee. |
| DOS PP | Invokes a DOS Program Product available from IBM for a license fee. |
| SPG | Described in the VM/370 System Programmer's Guide. |
| IPCS | Described in the VM/370 Interactive Problem Control System (IPCS) User's Guide. |

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

Figure 11. CMS Command Summary (Part 1 of 4)

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

Figure 11. CMS Command Summary (Part 2 of 4)

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

Figure 11. CMS Command Summary (Part 3 of 4)

| Command | Code | Usage |
|---|---|---|
| SORT | | Arrange a specified file in ascending order according to sort fields in the data records. |
| SSERV | | Copy a DOS/VS source statement book onto a CMS disk, display it at the terminal, or spool a copy to the virtual punch or printer. |
| START | | Begin execution of programs previously loaded (OS and CMS) or fetched (CMS/DOS). |
| STATE | | Verify the existence of a CMS disk file. |
| STATEW | | Verify a file on a read/write CMS disk. |
| SVCTRACE | | Record information about supervisor calls. |
| SYNONYM | | Invoke a table containing synonyms you have created for CMS and user-written commands. |
| TAPE | | Perform tape-to-disk and disk-to-tape operations for CMS files, and position tapes. |
| TAPEMAC | | Create CMS MACLIB libraries directly from an IEHMOVE-created partitioned data set on tape. |
| TAPPDS | | Load OS partitioned data set (PDS) files or card image files from tape to disk. |
| TESTCOB | OS PP | Invoke the OS COBOL Interactive Debug Program. |
| TESTFORT | OS PP | Invoke the FORTRAN Interactive Debug Program. |
| TXTLIB | | Generate and modify text libraries. |
| TYPE | | Display all or part of a CMS file at the terminal. |
| UPDATE | | Make changes in a program source file as defined by control cards in a control file. |
| VSAPL | OS PP | Invoke VS APL interface in CMS. |
| VSBASIC | OS PP | Compile and execute VS BASIC programs under CMS. |
| VSBUTIL | OS PP | Convert BASIC 1.2 data files to VS BASIC format. |

Figure 11. CMS Command Summary (Part 4 of 4)

| Command | Code | Usage |
|---|---|---|
| ASM3705 | SYSGEN | Assemble 370x source code. |
| ASMGEND | SYSGEN | Regenerate the VM/370 assembler command modules. |
| CMSGEND | SYSGEN | Generate a new CMS disk-resident module from updated TEXT files. |
| CMSXGEN | SYSGEN | Generate the CMSSEG discontiguous saved segment. |
| CPEREP | EREP | Formats and edits system error records for output. |
| DIRECT | SYSGEN | Set up VM/370 directory entries. |
| DOSGEN | SYSGEN | Load and save the CMSDOS shared segment. |
| DUMPSCAN | IPCS | Provide interactive analysis of CP abend dumps. |
| GEN3705 | SYSGEN | Generate an EXEC file that assembles and link-edits the 370x control program. |
| GENERATE | SYSGEN | Update VM/370 or the VM/370 directory, or generate a new standalone copy of a service program. |
| LKED | SYSGEN | Link-edit the 370x control program. |
| NCPDUMP | OP Gd, SPG | Process CP spool reader files created by 370x dumping operations. |
| PRB | IPCS | Update IPCS problem status. |
| PROB | IPCS | Enter a problem report in IPCS. |
| SAVENCP | SYSGEN, SPG | Read 370x control program load into virtual storage and save an image on a CP-owned disk. |
| SETKEY | SPG | Assign storage protect keys to storage assigned to named systems. |
| STAT | IPCS | Display the status of reported system problems. |
| VMFBLD | SYSGEN | Generate and/or update VM/370 using the PLC tape. |
| VMFDUMP | Op Gd, IPCS | Format and print system abend dumps; under IPCS, create a problem report. |
| VMFLOAD | SYSGEN | Generate a new CP, CMS or RSCS module. |
| VSAMGEN | SYSGEN | Load and save the CMSVSAM and CMSAMS segments. |
| ZAP | Op Gd, SPG | Modify or dump LOADLIB, TXTLIB, or MODULE files. |

Figure 12. CMS Commands for System Programmers

Figure 13 contains an alphabetical list of the RSCS commands with a brief statement about the use of each command. RSCS commands are described in the VM/370 Remote Spooling Communications Subsystem (RSCS) User's Guide.

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

Figure 13. RSCS Command Summary

# Index

The entries in this Index are accumulative. They list additions to this publication by the following VM/370 System Control Program Products:

- VM/370 Basic System Extensions, Program Number 5748-XX8
- VM/370 System Extensions, Program Number 5748-XE1

However, the text within the publication is not accumulative; it only relates to the one SCP program product that is installed on your system. Therefore, there may be topics and references listed in this Index that are not contained in the body of this publication.


*
  command (CP)
    described 42
    summary 172
* operand
  of MESSAGE command (CP) 102
  of SPOOL command (CP) 141
*dumpid operand, VMDUMP command 169

#CP
  command (CP)
    commandline operand 43
    described 43
    summary 172
#CP command (CP), example of usage 44

A
abbreviation, use of 37
accounting, information, displaying 129
ACNT, command (CP), summary 172
ACNT operand, of SET command (CP) 129
ADSTOP
  command (CP)
    described 45
    hexloc operand 45
    OFF operand 45
    summary 172
AFFINITY operand, of SET command (CP) 129
affinity setting, relation to the logon process 99
ALL operand
  of CHANGE command (CP) 49
  of PURGE command (CP) 107
  of QUERY command (CP) 110
  of TRACE command (CP) 161
  of TRANSFER command (CP) 166
altering
  attributes of
    virtual printer file 49
    virtual punch file 49
    virtual reader file 49
  logical editing symbols 157
  virtual machine environment 60
annotating your console sheet 42
Any user, described 8
APL, control of translation tables 157
APL operand, of TERMINAL command (CP) 157
ASSIST operand, of SET command (CP) 129
ATTACH, command (CP), summary 172
attached processor, INDICATE command (CP) use for monitor of system load 85
attached processor application, use of SET ASSIST 129
attention
  handling, terminal control of 157
  interrupt
    entering from your virtual console 47
    for a virtual machine 126
ATTN
  command (CP)
    described 47
    summary 172
ATTN operand, of TERMINAL command (CP) 157
attributes
  modifying, for 3800 printer 141
  spool file, changing 49
  virtual device, modifying 141
AUTOCR option, of PARM operand, for IPL command (CP) 90
AUTOPOLL operand, of SET command (CP) 129

B
BACKSPAC, command (CP), summary 172
BATCH option, of PARM operand, for IPL command (CP) 90
BEGIN
  command (CP)
    described 48
    hexloc operand 48
    summary 172
BEGIN command (CP), used with the DISCONN command (CP) 72
blanks, as delimiters 3
block address
  used to load a system
    at IPL (5748-XX8) 88
    at IPL (5748-XE1) 88
BOTH operand, of TRACE command (CP) 161
braces, use of 38
brackets, use of 38
BRANCH operand, of TRACE command (CP) 161

branches, tracing 161
byte alignment, DISPLAY command 29
bytecount operand, VMDUMP command 169

C
CAW (Channel Address Word), displaying 74
CAW operand, of DISPLAY command (CP) 74
CCW operand, of TRACE command (CP) 161
CCW translation, controlling 129
CHANGE
  command (CP)
    ALL operand 49
    CHARS operand 49
    CLASS operand 49
    COPY* operand 49
    COPY operand 49
    described 49
    DIST operand 49
    FCB operand 49
    FLASH operand 49
    HOLD operand 49
    MODIFY operand 49
    NAME operand 49
    NOHOLD operand 49
    PRINTER operand 49
    PUNCH operand 49
    READER operand 49
    summary 172
changing
  extended control registers in your virtual machine 150
  filename of a spool file 49
  floating-point registers in your virtual machine 150
  general registers in your virtual machine 150
  Program Status Word (PSW) in your virtual machine 150
  spool class of a file 49
  spool file attributes 49
  spoolid of a file 49
  virtual storage locations 150
channel c operand, of DETACH CHANNEL command (CP) 69
CHANNELS operand
  of DEFINE command (CP) 60
  of QUERY command (CP) 110
channel-to-channel adapters, virtual, connecting 57
character
  set 4
  translation tables, APL, controlling 157
CHARDEL operand, of TERMINAL command (CP) 157
CHARS operand
  of CHANGE command (CP) 49
  of SPOOL command (CP) 141
class
  privilege
    commands assigned to each 8
    for CP commands 7
  spool, changing for a file 49
  virtual device spool, modifying 141
class Any, described 9
class G, described 9
CLASS operand
  of CHANGE command (CP) 49
  of ORDER command (CP) 105
  of PURGE command (CP) 107
  of QUERY command (CP) 110
  of SPOOL command (CP) 141
  of TRANSFER command (CP) 166
CLEAR operand
  of IPL command (CP) 88
  of SYSTEM command (CP) 153
clearing
  interrupts from a virtual machine 127
  storage to zeros 88,153
CLOSE
  command (CP)
    CONSOLE operand 53
    described 53
    DIST operand 53
    dsname operand 53
    fnft operand 53
    HOLD operand 53
    NAME operand 53
    NOHOLD operand 53
    PRINTER operand 53
    PUNCH operand 53
    PURGE operand 53
    READER operand 53
    summary 172
    vaddr operand 53
  HOLD restrictions 53
  NOHOLD restrictions 53
  restrictions 53
CLOSE operand, of SPOOL command (CP) 141
closing
  files 53
  TRACE output 33
  virtual devices 141
CMS commands, summary of 176
CMS commands for system programmers, summary of 177
code operand, of EXTERNAL command (CP) 84
command
  name 4
  operands 4
command environment
  CP 3
  defined 3
commandline operand
  of #CP command (CP) 43
  of CP command 59
commands
  ADSTOP command (CP), halting execution of virtual machine 27
  ATTACH command (CP), with dedicated devices 20
  ATTN command (CP), simulates real console function 24
  CHANGE command (CP), altering spool file attributes 16
  CLOSE command (CP), using with spool files 16
  CMS
    for system programmers 177
    summarized 176
  CP, summarized 172
  DEFINE command (CP)
    reconfiguring virtual machine 26
    temporary disks 14

DETACH CHANNEL command (CP), with dedicated channels 21
DETACH command (CP)
removing dedicated devices 20
temporary disks 14
DISPLAY command (CP), displaying virtual storage 28
DUMP command (CP), printing virtual storage 30
EXTERNAL command (CP), simulating interruptions 24
LINK command (CP), sharing virtual disks 14
MESSAGE command (CP), obtaining operator assistance 21
notational conventions 37
ORDER command (CP), reordering spool files 19
PURGE command (CP), deleting spool files 19
QUERY command (CP), determining device or file attributes 16
REQUEST command (CP), simulates real console function 24
RSCS, summarized 182
SPOOL command (CP), altering device attributes 17
STORE command (CP), altering virtual storage 31
summary of use 172
TRACE command (CP), follow virtual machine activities 33
used to control a terminal session, summary of 172
comments, how to write 4
connecting
remote terminals to virtual machines 70
virtual channel-to-channel adapters 57
console
input/output, terminating 141
sheet, annotation of 42
spool file, modifying 141
spooling, controlling 141
CONSOLE operand
of CLOSE command (CP) 53
of DEFINE command (CP) 60
of QUERY command (CP) 110
of SPOOL command (CP) 141
CONT operand, of SPOOL command (CP) 141
continuous reading of an entire file 141
control, functions, setting for your virtual machine 129
Control Program (CP), console function mode 26
controlling
a terminal session, summary of commands used 172
3270 display (5748-XX8) 160.1
3270 display (5748-XE1) 160.1
controlling virtual machine, during terminals session 11
COPY* operand
of CHANGE command (CP) 49
of SPOOL command (CP) 141
COPY operand
of CHANGE command (CP) 49
of SPOOL command (CP) 141
COUPLE
command (CP)
described 57
summary 172
userid operand 57
vaddr1 operand 57
vaddr2 operand 57
CP
command (CP)
commandline operand 59
described 59
summary 172
CP command environment 3
CP commands
belonging to each privilege class 8
entering 24
from CP console function mode 26
from virtual console read mode 26
while running 24
entering from the virtual machine environment 59
privilege classes, described 8
privilege classes for 7
text format of 41
CPUID operand, of QUERY command (CP) 110
creating disk space with DEFINE command (CP), precautions 60
CSW (Channel Status Word), displaying 74
CSW operand
of DISPLAY command (CP) 74
of TRACE command (CP) 161
CTCA operand, of DEFINE command (CP) 60

D
DASD operand, of QUERY command (CP) 110
DCP, command (CP), summary 172
debugging, programs 27
dedicated channel, detaching 69
dedicated 2305 restriction 60
DEFINE
command (CP)
CHANNELS operand 60
CONSOLE operand 60
CTCA operand 60
described 60
GRAF operand 60
LINE operand 60
PRINTER operand 60
PUNCH operand 60
READER operand 60
STORAGE operand 60
summary 172
TFB512 operand (5748-XX8) 60
TFB-512 operand (5748-XE1) 60
TIMER operand 60
T2305 operand 60
T2314 operand 60
T2319 operand 60
T3310 operand (5748-XX8) 60
T3310 operand (5748-XE1) 60
T3330 operand 60
T3340 operand 60
T3350 operand 60
T3370 operand (5748-XX8) 60
T3370 operand (5748-XE1) 60

1403 operand 60
1443 operand 60
2501 operand 60
2540P operand 60
2540R operand 60
3203 operand 60
3211 operand 60
3262 operand (5748-XX8) 60
3262 operand (5748-XE1) 60
3289E operand (5748-XX8) 60
3289E operand (5748-XE1) 60
3505 operand 60
3525 operand 60
3800 operand 60
DEFINE command (CP)
temporary disks 14
use with MSS devices 60
virtual addresses 60
defining
fixed-block mode devices
for the virtual machine (5748-XX8) 60
for the virtual machine (5748-XE1) 60
virtual devices 60
delimiters, in command lines 3
DETACH
command (CP)
described 66
summary 173
vaddr operand 66
DETACH CHANNEL
command (CP)
channel c operand 69
described 69
DETACH command (CP), temporary disks 14
detaching
dedicated channels 69
virtual devices 66
DEV operand, of TAG command (CP) 154
device-end interrupt pending for a virtual device, specifying 125
devices
virtual
linking to 91
simulating not ready status for 104
DIAL
command (CP)
described 70
summary 173
userid operand 70
vaddr operand 70
DIAL command (CP), device restriction 70
directory entry, controlling, virtual machine attributes 12
DISABLE, command (CP), summary 173
DISCONN
command (CP)
described 72
HOLD operand 72
summary 173
disconnecting
virtual console 72
virtual machine 97
DISPLAY
command (CP)
CAW operand 74
CSW operand 74
described 74
PSW operand 74
summary 173
DISPLAY command (CP), byte alignment 29
displaying
accounting information 129
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 (CP) 49
of CLOSE command (CP) 53
distribution code, for an output file 49
DMCP, command (CP), summary 173
DRAIN, command (CP), summary 173
dsname operand, of CLOSE command (CP) 53
DSS operand, VMDUMP command 169
DUMP
command (CP)
described 80
summary 173

E
ECHO
command (CP)
described 83
nn operand 83
summary 173
ECMODE operand, of SET command (CP) 129
editing symbols
controlling 157
logical, controlling use of 129
EMSG operand, of SET command (CP) 129
ENABLE, command (CP), summary 173
END operand, of TRACE command (CP) 161
entering
a CP command from the virtual machine environment 59
CP commands 3
EOF operand, of SPOOL command (CP) 141
error message handling, controlling 129
ESCAPE operand, of TERMINAL command (CP) 157
execution
halting at an instruction address 45
of a virtual machine, resuming 48
extended control mode, setting 129
extended control registers
changing your virtual machine 150
displaying 74
printing 80
EXTERNAL
command (CP)
code operand 84
described 84
summary 173
external interrupt, simulated 84
EXTERNAL operand, of TRACE command (CP) 161

F
FB-512 device information
  obtained with QUERY command (5748-XX8) 116
  obtained with QUERY command (5748-XE1) 116
FB-512 devices
  tracing CCW information (5748-XX8) 163
  tracing CCW information (5748-XE1) 163
FCB operand
  of CHANGE command (CP) 49
  of LOADVFCB command (CP) 95
  of SPOOL command (CP) 141
FILE operand, of TAG command (CP) 154
files
  changing the spool class of 49
  changing the spoolid for 49
  closing 53
  distribution code for 49
  holding before output processing 49
  holding from further processing 53,141
  holding via CLOSE 141
  purging 53,141
  purging via SPOOL 141
  reclaiming 166
  reordering closed spool files 105
  spool, changing filename for 49
  terminating processing of 53
  transferring to another user 141
  transmitting to a virtual reader 166
FILES operand, of QUERY command (CP) 110
first-level storage, requesting via DISPLAY command (CP) 76
FLASH operand
  of CHANGE command (CP) 49
  of SPOOL command (CP) 141
floating-point registers
  changing your virtual machine 150
  displaying 74
  printing 80
FLUSH, command (CP), summary 173
fnft operand, of CLOSE command (CP) 53
FOR operand, of SPOOL command (CP) 141
FORCE, command (CP), summary 173
FORMAT vmtype operand, VMDUMP command 169
forms controls for a virtual printer, specifying 95
FREE, command (CP), summary 173
FROM operand, of TRANSFER command (CP) 166

G
general registers
  changing in your virtual machine 150
  displaying 74
  printing 80
general user, described 8
GRAF operand
  of DEFINE command (CP) 60
  of QUERY command (CP) 110

H
HALT, command (CP), summary 173
halting, execution at an instruction address 45
handshaking, control of VM/VS 129
hexloc operand
  of ADSTOP command (CP) 45
  of BEGIN command (CP) 48
hexloc1 operand, VMDUMP command 169
hexloc2 operand, VMDUMP command 169
hiding your password 99
highlighting
  3270 terminal display (5748-XX8) 160
  3270 terminal display (5748-XE1) 160.1
HILIGHT operand
  of TERMINAL command (CP) (5748-XX8) 157
  of TERMINAL command (CP) (5748-XE1) 157
HOLD, command (CP), summary 173
HOLD operand
  of CHANGE command (CP) 49
  of CLOSE command (CP) 53
  of DISCONN command (CP) 72
  of LOGOFF command (CP) 97
  of SPOOL command (CP) 141
holding
  a file before output processing 49
  files by user hold status 141
  files from further processing 53,141

I
IMSG operand, of SET command (CP) 129
INDEX operand, for LOADVFCB command (CP) 95
INDICATE
  command (CP)
    attached processor load indication 85
    described 85
    LOAD operand 85
    summary 173
    USER operand 85
initial program load (IPL)
  automatic, suppression of 99
  of a virtual machine operating system 88
  stopping 88
input/output operations, tracing 161
INSTRUCT operand, of TRACE command (CP) 161
internal print file duplication, by 3800 printer 49
interpreting, Figure 4 8
interrupt codes
  X'1004', clock comparator 24
  X'1005', CPU timer 24
interruptions, simulating 24
interrupts
  clearing from a virtual machine 127
  device-end, specifying virtual 125
  external, simulating 84
  for a virtual machine 126
  range of external codes 84
  tracing 161
I/O devices, virtual, defining for your virtual machine 60
I/O operand, of TRACE command (CP) 161
IPL
  command (CP)
    CLEAR operand 88
    described 88

nnnnnn operand (5748-XX8) 88
nnnnnn operand (5748-XE1) 88
NOCLEAR operand 88
PARM operand 88
STOP operand 88
summary 173
vaddr operand 88
IPL command (CP), for named systems 89
ISAM operand, of SET command (CP) 129

K
keys
program function, controlling 129
storage
displaying 74
printing 80

L
line editing symbols, controlling 129
line length, controlling at your terminal 157
LINE operand, of DEFINE command (CP) 60
LINEDEL operand, of TERMINAL command (CP) 157
LINEDIT operand, of SET command (CP) 129
LINEND operand, of TERMINAL command (CP) 157
LINES operand, of QUERY command (CP) 110
LINESIZE operand, of TERMINAL command (CP) 157
LINK
command (CP)
described 91
password operand 91
summary 173
userid operand 91
vaddr operand 91
LINK command (CP)
batch job restrictions 91
passwords 91
2305 restrictions 91
linking, a device to your virtual machine 91
LINKS operand, of QUERY command (CP) 110
load key, simulation of via IPL command (CP) 88
LOAD operand, of INDICATE commmand 85
LOADBUF, command (CP), summary 173
loading, an operating system 21
loading a system
with a block address
at IPL (5748-XX8) 88
at IPL (5748-XE1) 88
LOADVFCB
command (CP)
described 95
FCB operand 95
INDEX operand 95
summary 174
vaddr operand 95
LOCATE, command (CP), summary 174
LOCK, command (CP), summary 174
logging off of your virtual machine 97
logging on your virtual machine 99
logical editing symbols
altering 157
controlling 157
controlling use of 129
logically connected terminal 70
LOGMSG operand, of QUERY command (CP) 123
LOGOFF
command (CP)
described 97
HOLD operand 97
summary 174
logon, affinity setting effect 99
LOGON
command (CP)
described 99
MASK operand 99
NOIPL operand 99
password operand 99
summary 174
userid operand 99
logon procedure, hiding your password during 99
lowercase operands (vaddr, userid, spoolid), described 8

M
MASK operand
of LOGON command (CP) 99
of TERMINAL command (CP) 157
masking
of passwords, controlling 157
password during logon 99
Mass Storage Systems (see MSS)
MESSAGE
command (CP)
* operand 102
described 102
msgtext operand 102
OPERATOR operand 102
summary 174
userid operand 102
message handling, error, handling 129
messages
controlling transmission of 129
sending to other users 102
MIGRATE, command (CP), summary (5748-XE1) 174
MODE operand, of TERMINAL command (CP) 157
MODIFY operand
of CHANGE command (CP) 49
of SPOOL command (CP) 141
modifying
spooling control options 141
storage, using ADSTOP command (CP) 45
virtual device attributes 141
MONITOR, command (CP), summary 174
MSG operand, of SET command (CP) 129
MSGNOH, command (CP), summary 174
msgtext operand
for MESSAGE command (CP) 102
of SMSG command (CP) 140
MSS, virtual volumes, defined with DEFINE command (CP) 60

N
NAME operand
of CHANGE command (CP) 49
of CLOSE command (CP) 53
named systems, IPL command (CP) for 88
NAMES operand, of QUERY command (CP) 123
nesting, described 38
NETWORK, command (CP), summary 174
nn operand, of ECHO command (CP) 83
nnnnnn operand
of IPL command (CP) (5748-XX8) 88
of IPL command (CP) (5748-XE1) 88
NOCLEAR operand, of IPL command (CP) 88
NOCONT operand, of SPOOL command (CP) 141
NOEOF operand, of SPOOL command (CP) 141
NOHOLD operand
of CHANGE command (CP) 49
of CLOSE command (CP) 53
of SPOOL command (CP) 141
NOIPL operand, of LOGON command (CP) 99
NORUN operand, of TRACE command (CP) 161
notational conventions 37
NOTERM operand, of SPOOL command (CP) 141
NOTRANS operand, of SET command (CP) 129
NOTREADY
command (CP)
described 104
summary 174
vaddr operand 104

O
OFF operand
of ADSTOP command (CP) 45
of SPOOL command (CP) 141
of TRACE command (CP) 161
operands, command 4
operands listed, described 8
operating system
for a virtual machine, passing parameters to 89
initial program load for 88
operating systems
loading 21
passing parameters at IPL 23
operator, sending messages to 102
OPERATOR operand, of MESSAGE command (CP) 102
ORDER
command (CP)
CLASS operand 105
described 105
PRINTER operand 105
PUNCH operand 105
READER operand 105
spoolid operand 105
summary 174
OS Indexed Sequential Access Method (ISAM) 129

P
PAGEX operand, of SET command (CP) 129
parameters, passing to a virtual machine operating system 88
PARM operand, of IPL command (CP) 88
passing parameters to your virtual machine operating system 89
password
controlling masking of 157
hiding during logon procedure 99
invalid format for commandline, precaution 99
with the LINK command (CP) 91
password operand
of LINK command (CP) 91
of LOGON command (CP) 99
PFnn operand
of QUERY command (CP) 110
of SET command (CP) 129
power off vs logoff, with 3270 terminal 97
print file requirements, for 3800 printer 51
printer , virtual, specifying forms controls for 95
printer files, virtual, altering the attributes of 49
PRINTER operand
of CHANGE command (CP) 49
of CLOSE command (CP) 53
of DEFINE command (CP) 60
of ORDER command (CP) 105
of PURGE command (CP) 107
of QUERY command (CP) 110
of SPOOL command (CP) 141
of TRACE command (CP) 161
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 (CP) 81
PRIV operand, of TRACE command (CP) 161
privilege class, commands belonging to each 8
privilege classes
for CP commands
defined 7
summarized 8
privileged instructions, tracing 161
processor, determining, attached or main 109
PROCESSOR operand, of QUERY command (CP) 110
program function keys, controlling 129
PROGRAM operand, of TRACE command (CP) 161
Program Status Word (PSW)
changing your virtual machine 150
displaying 74
printing 80
PSW operand, of DISPLAY command (CP) 74
PUNCH
operand
of CHANGE command (CP) 49
of PURGE command (CP) 107
of QUERY command (CP) 110
punch files, virtual, altering the attributes of 49
PUNCH operand
of CLOSE command (CP) 53
of DEFINE command (CP) 60
of SPOOL command (CP) 141

PUNCH operator, of ORDER command (CP) 105
PURGE
  command (CP)
    ALL operand 107
    CLASS operand 107
    described 107
    PRINTER operand 107
    PUNCH operand 107
    READER operand 107
    spoolid operand 107
    summary 174
PURGE operand
  of CLOSE command (CP) 53
  of SPOOL command (CP) 141
purging
  closed files 53
  spooled files 141
purging closed spool files 107

Q
QUERY
  command (CP)
    ALL operand 110
    CHANNELS operand 110
    CLASS operand 110
    CONSOLE operand 110
    CPUID operand 110
    DASD operand 110
    described 109
    FILES operand 110
    GRAF operand 110
    LINES operand 110
    LINKS operand 110
    LOGMSG operand 123
    NAMES operand 123
    PFnn operand 110
    PRINTER operand 110
    PROCESSOR operand 110
    PUNCH operand 110
    READER operand 110
    SET operand 110
    SPMODE operand (5748-XE1) 110
    spoolid operand 110
    STORAGE operand 110
    summary 174
    S370E operand (5748-XE1) 110
    TAPES operand 110
    TBL operand 110
    TERMINAL operand 110
    TIME operand 110
    UR operand 110
    USERS operand 123
    vaddr operand 110
    VIRTUAL operand 110
    VMSAVE operand (5748-XX8) 110
    VMSAVE operand (5748-XE1) 110
QUERY operand, of TAG command (CP) 154
QVM, command (CP), summary (5748-XE1) 174

R
reader files, virtual, altering the attributes of 49
READER operand
  of CHANGE command (CP) 49
  of CLOSE command (CP) 53
  of DEFINE command (CP) 60
  of ORDER command (CP) 105
  of PURGE command (CP) 107
  of QUERY command (CP) 110
  of SPOOL command (CP) 141
reading, an entire file continuously 141
read/write access, with LINK command (CP) 91
READY
  command (CP)
    described 125
    summary 174
    vaddr operand 125
real computer
  reset button, simulating 153
  restart button, simulating 153
reclaiming files, in order to purge 107
reclaiming transmitted files 166
reconfiguring, your virtual machine 26
registers
  extended control
    displaying 74
    printing 80
  floating-point
    displaying 74
    printing 80
  general
    displaying 74
    printing 80
Remote Spooling Communications Subsystem
  receiving files 149
  TAG command (CP) 154
  transmitting files 149
remote terminal, connecting to a virtual machine 70
removing a virtual device from your virtual machine 66
REPEAT, command (CP), summary 174
REQUEST
  command (CP)
    described 126
    performs same function as ATTN command 47
    summary 174
reserved IBM class, described 8
RESET
  command (CP)
    described 127
    summary 174
    vaddr operand 127
reset button, simulating 153
RESET operand, of SYSTEM command (CP) 153
response
  for ADSTOP command (CP) 46
  for DETACH CHANNEL command (CP) 69

for ORDER command (CP) 106
for PURGE command (CP) 108
for QUERY CPUID command (CP) 114
for QUERY FILES command (CP) 115
for QUERY LINKS command (CP) 119
for QUERY LOGMSG command (CP) 123
for QUERY NAMES command (CP) 124
for QUERY PFnn command (CP) 121
for QUERY PROCESSOR command (CP) 122
for QUERY READER, PRINTER, and PUNCH command (CP) 120
for QUERY SET command (CP) 113
for QUERY S370E command (CP) (5748-XE1) 122
for QUERY TERMINAL command (CP) 114
for QUERY TIME command (CP) 113
for QUERY USERS command (CP) 124
for QUERY VIRTUAL CHANNELS command (CP) 115
for QUERY VIRTUAL CONSOLE command (CP) 115
for QUERY VIRTUAL DASD command (CP) 116
for QUERY VIRTUAL GRAF command (CP) 115
for QUERY VIRTUAL LINES command (CP) 117
for QUERY VIRTUAL STORAGE command (CP) 119
for QUERY VIRTUAL TAPES command (CP) 116.1
for QUERY VIRTUAL UR command (CP) 117
for QUERY VIRTUAL vaddr command (CP) 119
for QUERY VMSAVE (5748-XX8) 122
for QUERY VMSAVE (5748-XE1) 122
for RESET command (CP) 127
for STORE command (CP) 152
responses
for #CP command (CP) 44
for CHANGE command (CP) 51
for CLOSE command (CP) 56
for COUPLE command (CP) 57
for CP command (CP) 59
for DEFINE command (CP) 65
for DETACH command (CP) 67
for DIAL command (CP) 70
for DISCONN command (CP) 72
for DISPLAY command (CP) 77
for DUMP command (CP) 82
for ECHO command (CP) 83
for INDICATE LOAD and USER 85
for LINK command (CP) 93
for LOGOFF command (CP) 97
for LOGON command (CP) 100
for MESSAGE command (CP) 103
for REWIND command (CP) 128
for SPOOL command (CP) 149
for SYSTEM command (CP) 153
for TAG command (CP) 156
for TRACE command (CP) 164
for TRANSFER command (CP) 166
for VMDUMP command (CP) 168
responses to CP commands, types of 41
responses vs system messages 41
restart button, simulating 153
RESTART operand, of SYSTEM command (CP) 153
resume
execution of your virtual machine 48
use of your virtual machine after disconnecting it 72
REWIND
command (CP)
described 128
summary 174
vaddr operand 128
rewinding a real tape 128
RSCS (see also Remote Spooling Communications Subsystem)
command summary 182
RUN operand
of SET command (CP) 129
of TRACE command (CP) 161

S
saved systems, IPL command (CP) for 88
SAVESYS, command (CP), summary 174
saving, virtual machine data 150
sending messages
to other users 102
to the operator 102
service representative, described 8
SET
command (CP)
ACNT operand 129
AFFINITY operand 129
ASSIST operand 129
AUTOPOLL operand 129
described 129
ECMODE operand 129
EMSG operand 129
IMSG operand 129
ISAM operand 129
LINEDIT operand 129
MSG operand 129
NOTRANS operand 129
PAGEX operand 129
PFnn operand 129
RUN operand 129
SMSG operand 129
STBYPASS operand (5748-XE1) 129
STMULTI operand (5748-XE1) 129
summary 174
TIMER operand 129
VMSAVE operand (5748-XX8) 129
VMSAVE operand (5748-XE1) 129
WNG operand 129
370E operand (5748-XE1) 129
SET operand, of QUERY command (CP) 110
shared segment, modification of 45,163
shared segments, modification of 152
sharing virtual disks 14
SHUTDOWN, command (CP), summary 175
simulating
not ready status for a virtual device 104
the reset button on a real computer 153
the restart button on a real computer 153
simulating interruptions 24
SIO operand, of TRACE command (CP) 161

SLEEP
command (CP)
described 139
summary 175
SMSG
command (CP)
described 140
msgtext operand 140
summary 175
userid operand 140
SMSG operand, of SET command (CP) 129
SPACE, command (CP), summary 175
specifying, device-end interrupt for a virtual device 125
SPMODE, command (CP), summary (5748-XE1) 175
SPMODE operand, of QUERY command (CP) (5748-XE1) 110
SPOOL
command (CP)
* operand 141
CHARS operand 141
CLASS operand 141
CLOSE operand 141
CONSOLE operand 141
CONT operand 141
COPY* operand 141
COPY operand 141
described 141
EOF operand 141
FCB operand 141
FLASH operand 141
FOR operand 141
HOLD operand 141
MODIFY operand 141
NOCONT operand 141
NOEOF operand 141
NOHOLD operand 141
NOTERM operand 141
OFF operand 141
PRINTER operand 141
PUNCH operand 141
PURGE operand 141
READER operand 141
START operand 141
STOP operand 141
summary 175
SYSTEM operand 141
TERM operand 141
TO operand 141
userid operand 141
vaddr operand 141
spool class
for a file, changing 49
modifying 141
SPOOL command (CP), use with TAG command 148
spool command options, set for a virtual device 53
spool file
altering TAG information 154
changing filename for 49
changing the attributes of 49
characteristics
changing 16
querying 16
setting 16
reordering 105
spoolid 147
TAG command (CP) 154
transmitting locally 147
transmitting to remote location 148,154
spoolid number 147
spoolid operand
of ORDER command (CP) 105
of PURGE command (CP) 107
of QUERY command (CP) 110
of TRANSFER command (CP) 166
spooling
control, options, modifying 141
operator, described 8
terminals session 15
virtual console 17
spooling output to a 3800 printer 147
SPTAPE
command
summary (5748-XX8) 176
summary (5748-XE1) 176
command (CP), summary (5748-XE1) 175
START, command (CP), summary 175
START operand, of SPOOL command (CP) 141
status
virtual machine, determining 109
words, tracing 161
STATUS operand
of STORE command (CP) 150
STORE command (CP), simulating hardware store 32
STBYPASS operand, of SET command (CP) (5748-XE1) 129
STCP, command (CP), summary 175
STMULTI operand, of SET command (CP) (5748-XE1) 129
STOP operand
of IPL command (CP) 88
of SPOOL command (CP) 141
stopping, initial program load (IPL) procedure 88
storage
clearing to zeros 88
keys, printing 80
locations, virtual, displaying 74
modification of 45
virtual, altering 31
storage keys, displaying 74
storage locations
virtual machine
altering 150
displaying 74
printing 80
STORAGE operand
of DEFINE command (CP) 60
of QUERY command (CP) 110
STORE
command (CP)
described 150
STATUS operand 150
summary 175
STORE command (CP), STATUS operand, simulate hardware store 32
storing
CPU status for virtual machine 150
virtual machine data 150

summary
  of CMS commands 176
  of CP commands 172
  of RSCS commands 182
suppressing, automatic IPL of a virtual machine operating system 99
SVC operand, of TRACE command (CP) 161
SYSTEM
  command (CP)
    CLEAR operand 153
    described 153
    RESET operand 153
    RESTART operand 153
    summary 175
system analyst, described 8
system messages vs CP command responses 41
SYSTEM operand
  of SPOOL command (CP) 141
  VMDUMP command 169
system operator, primary, described 8
system programmer, described 8
system resource operator, described 8
system resources, indication of 85
S370E operand, of QUERY command (CP) (5748-XE1) 110

T
TABCHAR operand
  of TERMINAL command (CP) 157
  of TERMINAL command (CP) (5748-XX8) 157
  of TERMINAL command (CP) (5748-XE1) 157
TAG
  command (CP)
    described 154
    DEV command 154
    FILE operand 154
    QUERY operand 154
    summary 175
TAG command (CP)
  altering TAG information 154
  transmitting files to remote location 154
  use with SPOOL command 148
tapes, rewinding 128
TAPES operand, of QUERY command (CP) 110
TBL operand, of QUERY command (CP) 110
temporary, virtual disks 14
temporary disk space, formatting and clearing 60
TERM operand, of SPOOL command (CP) 141
TERMINAL
  command (CP)
    APL operand 157
    ATTN operand 157
    CHARDEL operand 157
    described 157
    ESCAPE operand 157
    HILIGHT operand (5748-XX8) 157
    HILIGHT operand (5748-XE1) 157
    LINEDEL operand 157
    LINEND operand 157
    LINESIZE operand 157
    MASK operand 157
    MODE operand 157
    summary 175
    TABCHAR operand 157
    TABCHAR operand (5748-XX8) 157
    TABCHAR operand (5748-XE1) 157
    TEXT operand 157
TERMINAL operand
  of QUERY command (CP) 110
  of TRACE command (CP) 161
terminals
  console, disconnecting from your virtual machine 72
  control of attention handling 157
  input/output processing, controlling 157
  line length, controlling 157
  remote, connecting to a virtual machine 70
  session
    altering virtual storage 31
    changing spool file characteristics 16
    commands used to control, summary of 172
    controlling I/O functions 11
    controlling virtual machine 21
    debugging programs 27
    dedicated channels 21
    dedicated devices 19
    determining the length of time of 109
    displaying virtual storage 27
    entering CP commands 24
    logging on 21
    purging spool files 19
    querying spool file characteristics 16
    reordering spool files 19
    setting spool file characteristics 16
    simulating interruptions 24
    spooling 15
    store virtual machine status 32
    testing programs 26
    tracing virtual machine activity 33
    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 (CP) 83
terminating
  console input/output 141
  processing of files 53
testing
  programs 26
  terminal 83
TEXT, control of translation tables 157
text format, of CP commands 41
TEXT operand, of TERMINAL command (CP) 157
TFB-512 devices
  defining fixed-block mode (5748-XX8) 60
  defining fixed-block mode (5748-XE1) 60
TFB512 operand, of DEFINE command (CP) (5748-XX8) 60
TFB-512 operand, of DEFINE command (CP) (5748-XE1) 60
time, determining length for a terminal session 109
TIME operand, of QUERY command (CP) 110
timer, virtual, controlling 129

TIMER operand
  of DEFINE command (CP) 60
  of SET command (CP) 129
TO * operand, VMDUMP command 169
TO operand
  of SPOOL command (CP) 141
  of TRANSFER command (CP) 166
TO userid operand, VMDUMP command 169
TRACE
  command (CP)
    ALL operand 161
    BOTH operand 161
    BRANCH operand 161
    CCW operand 161
    CSW operand 161
    described 161
    END operand 161
    EXTERNAL operand 161
    INSTRUCT operand 161
    I/O operand 161
    NORUN operand 161
    OFF operand 161
    PRINTER operand 161
    PRIV operand 161
    PROGRAM operand 161
    RUN operand 161
    SIO operand 161
    summary 175
    SVC operand 161
    TERMINAL operand 161
TRACE output, closing 33
tracing
  fixed-block mode device information (5748-XX8) 163
  fixed-block mode device information (5748-XE1) 163
  virtual machine activity 33,161
TRANSFER
  command (CP)
    ALL operand 166
    CLASS operand 166
    described 166
    FROM operand 166
    spoolid operand 166
    summary 175
    TO operand 166
    userid operand 166
transferring
  files
    to a virtual reader 166
    to another user 141
translating, CCWs, controlling 129
transmitting
  spool files
    locally 147
    to remote locations 148
T2305 operand, of DEFINE command (CP) 60
T2314 operand, of DEFINE command (CP) 60
T2319 operand, of DEFINE command (CP) 60
T3310 operand
  of DEFINE command (CP) (5748-XX8) 60
  of DEFINE command (CP) (5748-XE1) 60
T3330 operand, of DEFINE command (CP) 60
T3340 operand, of DEFINE command (CP) 60
T3350 operand, of DEFINE command (CP) 60
T3370 operand
  of DEFINE command (CP) (5748-XX8) 60
  of DEFINE command (CP) (5748-XE1) 60

U
underscore, use of 38
UNLOCK, command (CP), summary 175
uppercase (letters and words) and special symbols, use of 38
UR operand, of QUERY command (CP) 110
USER operand
  of INDICATE command (CP) 85
  of QUERY command (CP) 123
userid operand
  of COUPLE command (CP) 57
  of DIAL command (CP) 70
  of LINK command (CP) 91
  of LOGON command (CP) 99
  of MESSAGE command (CP) 102
  of SMSG command (CP) 140
  of SPOOL command (CP) 141
  of TRACE command (CP) 166
using virtual forms control buffer
  with the 3289E printer (5748-XX8) 95
  with the 3289E printer (5748-XE1) 95

V
vaddr operand
  of CLOSE command (CP) 53
  of DETACH command (CP) 66
  of DIAL command (CP) 70
  of IPL command (CP) 88
  of LINK command (CP) 91
  of LOADVFCB command (CP) 95
  of NOTREADY command (CP) 104
  of QUERY command (CP) 110
  of READY command (CP) 125
  of RESET command (CP) 127
  of REWIND command (CP) 128
  of SPOOL command (CP) 141
vaddr1 operand, of COUPLE command (CP) 57
vaddr2 operand, of COUPLE command (CP) 57
VARY, command (CP), summary 175
virtual console
  disconnecting from your virtual machine 72
  read environment 26
  spooling 17,141
virtual devices
  closing 141
  defining 60
  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 60
  removing from your virtual machine 66
  simulating not ready status for 104
  specifying a device-end interrupt for 125
virtual disks, permanent 14
virtual machine
  activity, tracing 161
  configuration, altering 60
  determining status of 109
  device, linking to 91
  directory entry 12
  disconnecting 97
  execution, resuming 48

logging off of 97
logging on 99
operating system, passing parameters to 88
placing in a dormant state 139
reconfiguring 26
sending and processing special messages 140
setting control functions for 129
storing information from 150
tracing activity 33
virtual machine environment, entering a CP command from 59
VIRTUAL operand, of QUERY command (CP) 110
virtual printer , specifying forms controls for 95
virtual storage, altering 31
virtual storage locations
changing 150
printing 80
virtual timer, controlling 129
virtual unit record devices 15
virtual volumes (3330V), defining 60
VMDUMP command (CP)
*dumpid operand 169
bytecount operand 169
DSS operand 169
FORMAT vmtype operand 169
hexloc1 operand 169
hexloc2 operand 169
SYSTEM operand 169
TO * operand 169
TO userid operand 169
VMSAVE operand
of QUERY command (CP) (5748-XX8) 110
of QUERY command (CP) (5748-XE1) 110
of SET command (CP) (5748-XX8) 129
of SET command (CP) (5748-XE1) 129
VM/VS handshaking, control of 129

W
WARNING, command (CP), summary 175
WNG operand, of SET command (CP) 129
writing, comments in VM/370 4

Z
zeros
clearing storage to 153
clearing storage to via IPL 88
clearing storage to via SYSTEM command (CP) 153

1
1403 operand, of DEFINE command (CP) 60
1443 operand, of DEFINE command (CP) 60

2
2501 operand, of DEFINE command (CP) 60
2540P operand, of DEFINE command (CP) 60
2540R operand, of DEFINE command (CP) 60

3
3203 device
as DEFINE operand 60
LOADVFCB restrictions 95
3203 operand, of DEFINE command (CP) 60
3203 printer, virtual, specifying forms controls for 95
3211 operand, of DEFINE command (CP) 60
3211 printer, virtual, specifying forms controls for 95
3262 operand
of DEFINE command (CP) (5748-XX8) 60
of DEFINE command (CP) (5748-XE1) 60
3270 device message length 102,140
3270 terminal, power off vs logoff, precautions 97
3270 terminal display
highlighting features available (5748-XX8) 160
highlighting features available (5748-XE1) 160.1
3289E operand
of DEFINE command (CP) (5748-XX8) 60
of DEFINE command (CP) (5748-XE1) 60
3289E printer
loading a virtual forms control buffer (5748-XX8) 95
loading a virtual forms control buffer (5748-XE1) 95
3505 operand, of DEFINE command (CP) 60
3525 operand, of DEFINE command (CP) 60
370E operand, of SET command (CP) (5748-XE1) 129
3800 operand, of DEFINE command (CP) 60
3800 printer
modifying device attributes 141
modifying spool file attributes 51
print file duplication 49
spooling output to one 147
using CHANGE command (CP) 51
using SPOOL command 141
3850 Mass Storage Systems (see MSS)

IBM VM/370 CP Command
Reference for General Users
GC20-1820-4

READER'S
COMMENT
FORM

This manual is part of a library that serves as a reference source for systems analysts, programmers, and operators of IBM systems. 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? | ☐ | ☐ |
| • Did you find the material: | | |
| Easy to read and understand? | ☐ | ☐ |
| Organized for convenient use? | ☐ | ☐ |
| Complete? | ☐ | ☐ |
| Well illustrated? | ☐ | ☐ |
| Written for your technical level? | ☐ | ☐ |

• What is your occupation? _____

• How do you use this publication:

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

**Your comments:**

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

Thank you for your cooperation. No postage stamp necessary if mailed in the U.S.A. (Elsewhere, an IBM office or representative will be happy to forward your comments.)

## Reader's Comment Form

Cut or Fold Along Line

IBM VM/370 CP Command Reference for General Users

Printed in U.S.A.

GC20-1820-4

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**

IBM VM/370 CP Command Reference for General Users

Printed in U.S.A.

GC20-1820-4



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