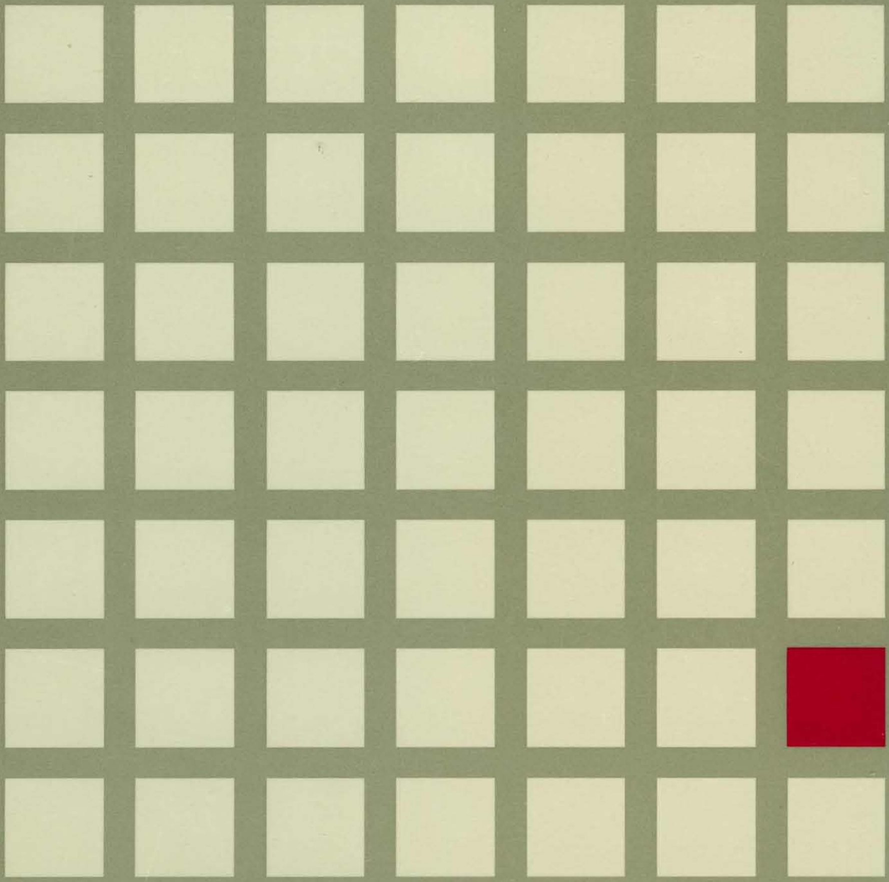
IBM

Virtual Machine/System Product

SX20-4400-05

Quick Reference

Release 6



SX20-4400-05

IBM Virtual Machine/System Product

Quick Reference

Release 6

Sixth Edition (July 1988)

This edition, SX20-4400-05, is a major revision of SX20-4400-04. It applies to Release 6 of the Virtual Machine/System Product, and to all subsequent releases unless otherwise indicated in new editions or Technical Newsletters. Changes are made periodically to the information herein; before using this publication in connection with the operation of IBM systems, consult the latest *IBM System/370, 30xx, 4300, and 9370 Processors Bibliography* GC20-0001, for the editions that are applicable and current.

Summary of Changes

For a detailed list of the changes, refer to the "Summary of Changes" on page 363.

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

Ordering Publications

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

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

© Copyright International Business Machines Corporation 1983, 1984, 1985, 1986, 1988. All rights reserved.

Preface

This publication is a quick reference for all users of VM/SP, from general users to experienced system programmers. It contains all the CP, CMS, GCS, TSAF, AVS, IPCS, and RSCS commands that are documented in the rest of the VM/SP library.

As its name suggests, this book is for quick reference. If you need in-depth explanation of commands, please refer to the pertinent books in the VM/SP Library listed in the bibliography at the end of this book. If you are a new user of VM/SP and need more help, you will find that the *VM/SP CMS Primer* and the *VM/SP CMS Primer for Line-Oriented Terminals* give detailed, easy-to-follow guidance on learning to use the VM/SP system.

The first section of the book is a collection of reference information about communicating with VM/SP and interpreting its responses, and working with two of its components, CP and CMS. The rest of the book consists of VM/SP commands and service aids. The complete format of each command (identical to those found in VM/SP reference manuals) is given, plus a brief description of its function. This section also explains the syntax used in the command formats, both in this book and in the rest of the VM/SP library.

This publication is part of a set of reference summaries that may be ordered as a group using order number SBOF-3255.

Note: The user privilege classes referred to throughout this book are IBM-defined classes. If your installation restructures the classes, see your installation administrator.

Contents

Chapter 1. VM/SP Reference Information **1**
Command Format Notation 1
Messages and Return Codes 4
Online HELP Facility 16
CMS File Identifiers 19
Naming Shared File System (SFS) Directories 30
Full-Screen CMS Default Settings 31
CP Command Privilege Classes 35

Chapter 2. CMS, CP, RSCS, TSAF, AVS, IPCS and GCS Commands **37**

Summary of Changes **363**
New Commands for Release 6 363
Changed Commands for Release 6 364

Bibliography **367**
Prerequisite Publications 367
Corequisite Publications 367

Index **369**

Chapter 1. VM/SP Reference Information

Command Format Notation

The command formats in this book (and in the entire VM/SP library) are shown using certain standard notational conventions.

Use of Symbols

When entering commands, enter the following exactly as they appear in the command format:

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

The following symbols distinguish operands and command syntax only. Do **not** enter them when entering commands:

braces { }

enclose a stack of choices, one of which **must** be selected. If a stack of choices is enclosed by neither brackets or braces, braces are assumed.

brackets [ ]

enclose a stack of choices, one of which **may** be selected. If brackets are nested, the outermost operand (surrounded by one pair of brackets) is the highest level of nesting. It must be selected in order to select the next lower-level operand nested within it, and so forth.

underscore ______

indicates a default option. The system chooses this option when the user does not specify one.

vertical bar |

separates operand alternatives within brackets and braces.

ellipsis ...

means that the preceding item may be repeated.

Use of Case

Uppercase

The uppercase letters in a command, keyword operand or option represents the minimum truncation that the system accepts.

Lowercase

Lowercase letters, words, and symbols appearing in *italics* are user- or system-supplied variables. The following table gives the meaning of selected variables used in this book.

Table 1 (Page 1 of 3). **Selected Variables Used in Command Formats**

Variable	Meaning
a...	alphabetic or numeric information
applid	application identifier
arg	argument
bbcchh	bin, cylinder, and head
cc	cylinder number
cchhr	cylinder, head, record
char	character
col	column
compid	component identifier
cm	command code, in hexadecimal
convid	VTAM conversation identifier
cpuid	central processing unit identification

Table 1 (Page 2 of 3). **Selected Variables Used in Command Formats**	
Variable	**Meaning**
csw	channel status word
cuu	virtual device address. Also: *vdev*
cyl	cylinder
date	system date
ddname	data definition name
devclass	class of IBM device
devname	mnemonic name for an IBM device type
devtype	IBM device type
dirid	SFS directory identifier
dsname	dataset name
dumpid	dump identifier
execname	file name of an exec
exectype	file type of an exec
ext	extension
fileid	file identifier
filepoolid	file pool identifier
fm	file mode
fn	file name
ft	file type
hexloc	hexadecimal location
imagelib	3800 printer image library
langid	language identifier
libname	library name
linkid	link identifier
lprt	logical printer
lrecl	logical record length
luname	logical unit name
membername	library member name
mode	mode letter, or mode letter and mode number
mmm	module name code

Table 1 (Page 3 of 3). **Selected Variables Used in Command Formats**

Variable	Meaning
n...	decimal information
nodeid	node of a user
prefix	prefix subcommand or macro
psw	Program Status Word
qual	qualifier
range	range (of addresses or registers)
rdev	real device address (formerly 'raddr' or 'cuu')
recfm	record format
reg	register
rid	resource identifier (formerly 'resid')
spoolid	spool file identification
svc	supervisor call number
sysname	system name
vdev	virtual device address (formerly 'vaddr' or 'cuu')
vname	virtual screen name
volid	volume serial number
wname	window name
x...	hexadecimal information
yyyy	reason code (GCS abend)

Messages and Return Codes

Message Format:

Messages from VM consist of a message identifier (for example, DMKCQG020E) and text. The identifier distinguishes each message, and the text describes a condition that has occurred or requests a response from the user.

The **message identifier** consists of four fields: a prefix, a module code, a message number, and a severity code. Its format is:

XXXMMM###S

where:

xxx	is the prefix. Messages are prefixed as follows: • CP messages — DMK • CMS messages — DMS • IPCS messages — DMM • GCS messages — CSI • TSAF messages — ATS • AVS messages — AGW.
mmm	is the module code, three characters that indicate which module generated the message. This field is usually an abbreviation of the name of the module in which the error occurred.
### or ####	is the message number, three or four digits that are associated with the condition that caused the message to be generated.
S	is the severity code, a letter that indicates what kind of condition caused the message. The definition of the severity codes depends on the nature of the routine producing the message.

The following table summarizes the message severity codes for each of the six VM/SP components. More complete information can be found in *VM/SP System Messages and Codes*. The *VM/SP System Messages Cross-Reference* lists messages grouped by command, module code, number and text.

VM/SP Reference Information

Table 2. **Types and Meanings of Severity Codes by Component**

Prefix	Codes	Meaning
DMK (CP Commands)	A I W E	Immediate action required Information only Warning *or* System Wait Error
DMS (CMS Commands)	R I W E S T	Response awaited from user Information only Warning *or* System Wait Error Severe Error Termination error
DMM (IPCS Commands)	R I E S	Response awaited from user Information only Error Severe error
CSI (GCS Commands)	R I E W S T	Response awaited from user Information only Warning Error Severe error Terminating error
ATS (TSAF Commands)	R I W E S T	Response awaited from user Information only Warning Error Severe error Terminating error
AGW (AVS Commands)	I W E S	Information only Warning Error Severe error

Return Codes

If a condition arises during execution of a command that results in the display of an error message, the command causes a nonzero return code to be displayed.

CMS Return Codes

The following table shows the codes which may be returned by CMS commands.

Note: Commands that call program products produce return codes set by that program. They may have the same numbers as CMS codes, but they have been redefined by the program product in operation.

Table 3 (Page 1 of 3). **Return Codes Produced by CMS**

Return Code	Meanings
-0001	• No CP command with this name was found. (The CP error code of +1 is converted by CMS to -0001 for commands entered from the virtual console.)
-0002	• An attempt was made to execute a CMS command while in CMS subset mode, which would have caused the module to be loaded in the user area (LOADMOD error code 32).
-0003	• No CMS command with this name was found.
-0004	• The LOADMOD failed (for example, there was an error on the module).
-0005	• A LOADMOD was attempted with the wrong environment (for example, the module was generated by the GENMOD command with the OS option and LOADMOD was attempted with DOS=ON specified).
-0006	• An attempt was made to invoke a CMS function or macro from the command line (or from a REXX EXEC via and ADDRESS CMS command, or from and EXEC 2 EXEC via an &PRESUME &SUBCOMMAND CMS.
0	• Normal.
1	• Device disconnected.

Table 3 (Page 2 of 3). **Return Codes Produced by CMS**

Return Code	**Meanings**
1	• Top or bottom of virtual screen reached.
3	• Virtual screen, window, or queue already exists.
3	• Data, field, or scroll amount is truncated.
4	• List or queue is empty.
4	• The user did not specify all the conditions necessary to execute the command as intended. Execution of the command continues; however, the result may or may not be as the user intended.
8	• Device errors occurred for which a warning message is issued, or errors have been introduced into the output file.
12	• Errors were found in the input file.
13	• No space available.
14	• No reserved or data area.
20	• An invalid character is in the file identifier. Valid characters are: 0-9, A-Z, a-z, $, @, #, +, -(hyphen), :(colon), _(underscore). • Window name of "*" or "=" not allowed.
24	• The user did not specify the command line correctly. • CMS virtual screen or window cannot be deleted.
28	• An error occurred while trying to access, or manipulate, a user's files. • Virtual screen, window, or queue not defined.
32	• The user's file is not in the expected format, or does not contain the expected information, or an attempt was made to execute a LOADMOD command while in CMS subset mode. • Invalid position specified.
36	• An error for which the user is responsible occurred on one of the user's devices. For example, a disk is in read-only status, and needs to be in write status so that a file can be written on it. • Window not connected or displaying virtual screen **or**. • No field to write data/color/exthi/PSset.

Table 3 (Page 3 of 3). **Return Codes Produced by CMS**

Return Code	**Meanings**
40	• A functional error for which the user is responsible occurred during execution of the command. • User failed to supply all the necessary conditions for executing the command. • End-of-file, end-of-tape (where applicable).
41	• Insufficient storage was available to execute the command.
88	• A CMS system restriction prevented execution of the command. • Function requested is an unsupported feature. • Device requested is an unsupported device. • TTY device.
100	• Input/output or input/output device error.
104	• A functional error for which the system is responsible occurred during execution of the command. • Insufficient storage.
256	• All unexpected errors for which the system is responsible; that is, terminal error messages. • Request rejected by IUCV.

The CMS **DASD Dump Restore (DDR) command** produces several return codes of its own:

Table 4. **Return Codes Produced by the CMS DDR Command**

1	Invalid file name, or file not found.
2	Error in executing the program.
3	Flagged DASD (Direct Access Storage Device) track.
4	Permanent tape or DASD I/O error.
1*xx*[1]	Error in the PRINTIO routine.
2*xx*[1]	Error in the CONREAD routine.
3*xx*[1]	Error in the RDBUF routine.
4*xx*[1]	Error in the TYPLIN routine.

[1] *xx* is the CMS routine return code.

CP Return Codes and Error Message Numbers

The result of entering a CP command may be either:

Ready; (or Ready(0);)
which indicates successful execution.

or

Ready(*nnnnn*);
which indicates an error. If the contents of *nnnnn* is a CMS return code, then the error occurred in CMS; If *nnnnn* contains a CP message number, the error occurred in CP.

Return Codes from CP commands correspond directly to the message numbers. CP error messages are divided into several categories according to this message number. The following table shows the function related to each group of error numbers and the category of users that receives each one.

Table 5. **CP Error Message Numbers**

Numbers	Related Function	Received By
001-349	CP commands and console functions	General Users
350-399	Nucleus loading	Primary System Operators
400-424	Paging	
425-449	Spooling	
450-474	Dispatching and service routines	
475-499	Directory routine	
500-549	Input/output error recovery	
550-599	Input/output error recording	
600-649	Machine check recovery	
650-699	Reserved for IBM use only	
700-729	DDR (dump restore) service program	System Support Personnel
730-749	FMT (format) service program	
750-799	DIR (directory) service program	
800-849	Reserved for IBM use only	
850-899	DMM (VMFDUMP) service routine	
900-999	Checkpoint, warm start, dump initialization	

IPCS Command Return Codes

The following table following figure shows the return codes produced by IPCS commands.

Table 6. **Return Codes Produced by IPCS Commands**

Code	Meaning
4	An incorrectly entered parameter.
8	System failure; a read/write error or an invalid internal parameter.
100	CMS error reading a file.
200	CMS error writing a file.
500	CMS error writing to the printer.
12	IPCS processing error.

XEDIT Command Return Codes

The following table shows the return codes produced by the XEDIT command.

Table 7 (Page 1 of 5). **Return Codes Produced by the XEDIT Command**

Code	Meaning(s)
-3	• Unknown command. • Invalid from environment other than EXEC 2 or REXX.
-2	• Invalid subset command.
-1	• Incorrect operands specified in the PARSE macro.
0	• Normal; Parsing was successful. • "N" lines were inserted.

Table 7 (Page 2 of 5). **Return Codes Produced by the XEDIT Command**

Code	Meaning(s)
1	• TOF or EOF reached (and displayed) during execution or change. • No change (SPLTJOIN issued at TOF or EOF). • Duplicate name defined. • Valid only in display mode. • Column pointer outside restored zone settings. • Only one file edited. • Parsing incomplete - scanned line does not match PARSE macro format. • No action taken - cursor will be set outside screen. • Out of zone definition during execution. • Partial delete because EOF or TOF reached reached during execution. • File has been filed, and was the only one edited. • Overlapping groups of lines. • Target line within lines to move. • No line(s) changed or cursor not on valid data field. • Total number of lines or columns exceeds physical screen size.
2	• Target line not found. • Name does not exist for OFF function.

Table 7 (Page 3 of 5). **Return Codes Produced by the XEDIT Command**

Code	Meaning(s)
3	• Terminal is not a display terminal. • Truncated or spilled. • Invalid placement of cursor or subcommand. • SORT cannot be used when a file is edited in UPDATE or extended mode. • Operand or subcommand is valid only for display terminal. • File already exists. • RGTLEFT valid in display mode only. • No PRESERVE has been issued. • Pool of deleted lines is empty. • "QUERY POINT *" issued, but no symbolic names defined. • Macro not currently in storage. • LOAD has already been issued. • Subcommand is not valid in extended mode, or records truncated.
4	• Insufficient storage available. • No lines changed. • Each logical screen must contain at least 5 lines and 20 columns. • Line is not reserved. • Lrecl must be lower than 65536 for recfm V. • File already in storage. • Too many control characters defined. • Invalid when issued from prefix macro. • Macro is in use - do not purge. • No change occurred (string1 not found).
5	• Invalid or missing operand, string, or (line) number.
6	• Subcommand rejected in the profile because of LOAD error. • QUIT issued in macro.
7	• Error building the update file.

Table 7 (Page 4 of 5). **Return Codes Produced by the XEDIT Command**

Code	Meaning(s)
8	• Prefix area contains pending subcommand or macro. • I/O error. • Modifications lost because PA key pressed while message pending.
12	• Disk defined in file mode is read-only. • File has been changed (use QQUIT to QUIT anyway).
13	• Disk is full.
16	• EXEC 2 variable greater than 256 characters.
20	• Invalid character in file name or file type.
24	• Invalid file mode. • Invalid parameters or options. • Invalid columns defined.
28	• Source file not found (UPDATE MODE). • Library not found (MEMBER option). • Specified profile macro does not exist. • File XEDTEMP CMSUTI already exists. • File name already exists.
32	• Error during updating process. • Record "firstrec" is beyond end of file. • File is not a library. • Library has no entries. • File is not fixed, with 80 character records.
36	• Disk not accessed yet.
40	• No list given.
80	• Unsupported OS data set.
81	• Unsupported OS data set.
82	• Unsupported OS data set.
83	• Unsupported OS data set.
88	• File is too large, cannot fit into storage. • Previous MACLIB function not found.

Table 7 (Page 5 of 5). **Return Codes Produced by the XEDIT Command**

Code	Meaning(s)
100	• Error reading/writing file to disk. • Error from rdbuf. • Error occurred while creating the file.
104	• No storage is available.
nn	• Command's return code specified as operand. • Same as repeated subcommand's return codes. • Return code of CMS or CP command. • Return code of subcommand or macro or from subcommand following LOCATE command.
any number >10	• Standard CMS HELP command return codes.

Online HELP Facility

The VM/SP online HELP facility contains information about:

- Commands
  - AVS, CMS, CP, IPCS, GCS and TSAF
- Subcommands
  - EDIT, XEDIT, SRPI, and IPCS
- Macros
  - CMS assembler language
- Routines
  - From the VMLIB callable services library (CSL)
- Messages
  - CMS, CP, TSAF, AVS, GCS, IPCS

Note: The following program products provide command and message HELP:

  - RSCS
  - PVM
  - SQL/DS

- Control Statements and Instructions
  - EXEC and EXEC2, REXX

Entering

```
help
```

shows you the various general subjects covered by HELP, and how to use it to perform some specific system tasks. Entering

```
help task
```

shows you a listing of all the subjects for which HELP is available.

Note: If in XEDIT you will immediately be shown HELP for XEDIT. You can also directly access HELP for a command by entering:

HELP.[*component*] [*command*]

HELP can contain three layers of information (*brief*, *detailed*, and *related* layers) about commands, each more complete and detailed than the one before. Access a specific layer directly by adding it as an option to the above command.

Understanding Command Formats in HELP

Online, braces appear as less than (<) and greater than (>) characters while brackets appear as parentheses, bars, and plus signs.

VM/SP Reference Information

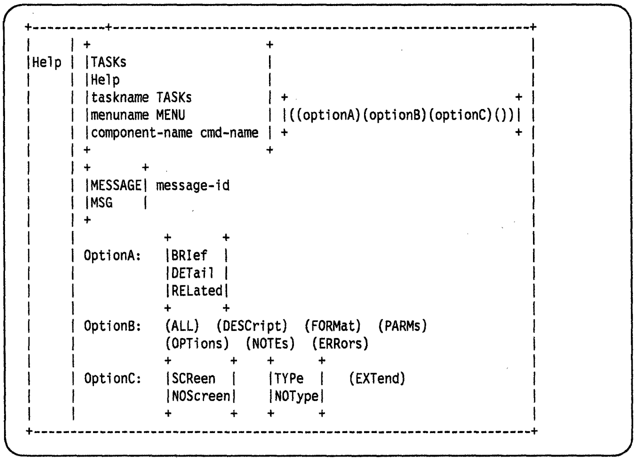
The command format in a reference manual:

Help	**TASKs** **Help** *taskname* **TASKs** *menuname* **MENU** *component-name* *cmd-name* [([optionA] [optionB] [optionC] [)]] **MESSAGE** *message-id* **MSG** **OptionA:** **BRIef** **DETail** **RELated** **OptionB:** [**ALL**] [**DESCript**] [**FORMat**] [**PARMs**] [**OPTions**] [**NOTEs**] [**ERRors**] **OptionC:** **SCReen** / **NOScreen** **TYPe** / **NOType** [**EXTend**]

The command format in online Help:

Note: To get the following example on your screen, type

`help cms help (format`



```
+----------+---------------------------------------------------------+
|     | +                        +                                   |
|Help | |TASKs                   |                                   |
|     | |Help                    |                                   |
|     | |taskname TASKs          | +                               + |
|     | |menuname MENU           | |((optionA)(optionB)(optionC)())| |
|     | |component-name cmd-name | +                               + |
|     | +                        +                                   |
|     | +       +                                                    |
|     | |MESSAGE| message-id                                         |
|     | |MSG    |                                                    |
|     | +                                                            |
|     |            +       +                                         |
|     | OptionA:   |BRIef  |                                         |
|     |            |DETail |                                         |
|     |            |RELated|                                         |
|     |            +       +                                         |
|     | OptionB:   (ALL)  (DESCript)  (FORMat)  (PARMs)              |
|     |            (OPTions)  (NOTEs)  (ERRors)                      |
|     |            +        +    +      +                            |
|     | OptionC:   |SCReen  |    |TYPe  |   (EXTend)                 |
|     |            |NOScreen|    |NOType|                            |
|     |            +        +    +      +                            |
+--------------------------------------------------------------------+
```

Getting Help for System Messages

You can find out what a VM/SP message means directly by entering:

```
HELP [message identifier]
```

For example, to get Help on system message DMSACC059E, enter:

```
====> help dms059e
```

or

```
====> help dmsacc059e
```

CMS File Identifiers

File Names

CMS files can have any file name and file type you wish, subject to the following rules:

- The file name and file type can each be from one to eight characters.
- The valid characters are A-Z, a-z, 0-9, $, #, @, +, - (hyphen), : (colon), and _ (underscore).

Note: Lowercase letters within a file ID are valid for use within the CMS file system. However, some CMS commands do not support file IDs that contain lowercase letters.

Specifying Subsets of Files with Special Characters

Some CMS commands allow you to use two special characters in the *fn* and *ft* operands, if you want to specify a subset of your files rather than just one file. These commands are:

- CREATE ALIAS
- CREATE LOCK
- DELETE LOCK
- FILELIST
- GRANT AUTHORITY
- LISTFILE
- QUERY ALIAS
- QUERY AUTHORITY
- QUERY LOCK
- RELOCATE
- REVOKE AUTHORITY

The special characters are * (asterisk) and % (percent), where:

*	represents any number of character(s). As many asterisks as required can appear *anywhere* in a file name or file type. However, the total number of characters, including the asterisks, may not exceed eight. For example, if you enter: `filelist *d* *file*` you are requesting that the list contain all files on your disk or directory accessed as A whose file name contains "d" and whose file type contains "file."
%	is a place-holding character for any single character. As many percent symbols as necessary may appear anywhere in a file name or file type.

Reserved File Types

The following file types are reserved for use by both CMS commands and the language processors:

In addition, there are special file types reserved for use by the language processors, which are IBM program products. For details, consult the appropriate program product manuals.

Table 8 (Page 1 of 9). **Reserved File types**

File type	**Command**	**Usage**	**File name**	**Format RECFM | LRECL**	**Contents**
AMSERV	AMSERV	Input	fn	F 80	Input control statements for Access Method Services
ASM3705	ASM3705 GEN3705	Input Output	fn fn(nn)	F 80 F 80	3704/3705 assembler source statements
ASSEMBLE	ASSEMBLE	Input	fn	F 80	Assembler language source statements
AUXxxxx	UPDATE XEDIT	Input	fn	F 80	Auxiliary update file
BASDATA	BASIC execution	Execution time files	fn	V 255	User input and output files
BASIC	BASIC	Input	fn	V 156	BASIC language source statements
CMSUT1	READCARD COPYFILE RECEIVE SENDFILE DISK LOAD TAPE LOAD UPDATE INCLUDE LOAD MACLIB EDIT TAPPDS XEDIT	Intermediate work files	READCARD COPYFILE RECEIVE SENDFILE DISK TAPE fn DMSLDR DMSLDR DMSLBM EDIT TAPPDS XEDIT	F 80	
CNTRL	UPDATE XEDIT	Input	fn	F 80	Control file update
COBOL	COBOL	Input	fn	F 80	COBOL source statements

Table 8 (Page 2 of 9). **Reserved File types**

File type	Command	Usage	File name	Format RECFM | LRECL	Contents
COPY	MACLIB SSERV	Input Output	fn fn	F 80	COPY control cards and macro definitions A book from a DOS/VS source library
CSLCNTRL	CSLGEN	Input	fn	F 80	Routine names, TEXT files, template files, and other CSL control files that are to be used in building the library.
CSLLIB	CSLGEN CSLLIST RTNLOAD	Output Input/Output Input	fn fn fn	F 80	Callable services library (CSL), generated by CSLGEN, for use on DASD.
CSLSEG	CSLGEN CSLLIST RTNLOAD	Output Input/Output Input	fn fn fn	F 80	Callable services library (CSL), generated by CSLGEN, for use in a logical saved segment.
DIRECT	DIRECT	Input	fn	F 80	User directory entries
DOSLIB	DOSLIB DOSLKED FETCH GLOBAL	Input Input Output Input	fn fn fn fn	V 1024	CMS/DOS phase library
DOSLNK	DOSLKED	Input	fn	F 80	Linkage editor control statements for input to CMS/DOS linkage editor

Table 8 (Page 3 of 9). **Reserved File types**

File type	Command	Usage	File name	Format RECFM | LRECL	Contents
ESERV	ESERV	Input	fn	F 80	Input control statements for ESERV program
EXEC	EXEC EXEC2 REXX LISTFILE GEN3705 LISTIO	Input Input Input Output Output Output	fn fn fn CMS fn $LISTIO	V 130 V 256 V none	EXEC statements
EXPAND	EXPAND	Input	fn		Control records that expand object files
FORTRAN	FORTGI FORTHX GOFORT TESTFORT	Input	fn	V 80	FORTRAN source statements
FREEFORT	GOFORT	Input	fn	V ≤81	FREEFORM FORTRAN source statements
FTnnF001	FORTRAN execution	Input/Output	fn		User input and output files
GCS	EXEC	Input	fn	V 130	EXEC statements
GLOBALV	GLOBALV DEFAULTS	Input/Output	fn Initial Session Lasting	F/V ≤5201 V ≤520 V ≤520	Collection of named variables
GROUP	GROUP	Output	fn	F 80	Group Control System (GCS) data block entries used to describe a GCS virtual machine group
HELPAVS	HELP	Input	fn	F/V 79	Input files for HELP facility
HELPCMS	HELP	Input	fn	V 79	Input files for HELP facility

Table 8 (Page 4 of 9). **Reserved File types**

File type	Command	Usage	File name	Format RECFM | LRECL		Contents
HELPCMSQ	HELP	Input	fn	V	79	Input files for HELP facility
HELPCMSS	HELP	Input	fn	V	79	Input files for HELP facility
HELPCP	HELP	Input	fn	V	79	Input files for HELP facility
HELPCPOT	HELP	Input	fn	V	79	Input files for HELP facility
HELPCPQU	HELP	Input	fn	V	79	Input files for HELP facility
HELPCPSE	HELP	Input	fn	V	79	Input files for HELP facility
HELPEDIT	HELP	Input	fn	V	79	Input files for HELP facility
HELPEXEC	HELP	Input	fn	V	79	Input files for HELP facility
HELPEXC2	HELP	Input	fn	V	79	Input files for HELP facility
HELPGROU	HELP	Input	fn	V	79	Input files for HELP facility
HELPHELP	HELP	Input	fn	V	79	Input files for HELP facility
HELPIPCS	HELP	Input	fn	V	79	Input files for HELP facility
HELPMACR	HELP	Input	fn	V	79	Input files for HELP facility
HELPMENU	HELP	Input	fn	V	77	Input files for HELP facility
HELPMSG	HELP	Input	fn	V	79	Input files for HELP facility
HELPPREF	HELP	Input	fn	V	79	Input files for HELP facility
HELPPVM	HELP	Input	fn	V	79	Input files for HELP facility

Table 8 (Page 5 of 9). **Reserved File types**

File type	Command	Usage	File name	Format RECFM | LRECL		Contents
HELPQUER	HELP	Input	fn	V	79	Input files for HELP facility
HELPREXX	HELP	Input	fn	V	79	Input files for HELP facility
HELPROUT	HELP	Input	fn	V	79	Input files for HELP facility
HELPRSCS	HELP	Input	fn	V	79	Input files for HELP facility
HELPSET	HELP	Input	fn	V	79	Input files for HELP facility
HELPSQLD	HELP	Input	fn	V	79	Input files for HELP facility
HELPSRPI	HELP	Input	fn	V	79	Input files for HELP facility
HELPTASK	HELP	Input	fn	V	105	Input files for HELP facility
HELPTSAF	HELP	Input	fn	V	79	Input files for HELP facility
HELPXEDI	HELP	Input	fn	V	79	Input files for HELP facility
LISTING	ASSEMBLE	Output	fn	F	121	COBOL processor output used as input to SOURCE subcommand of TESTCOB
	ASM3705	Output	fn			
	ESERV	Output	fn			
	GOFORT					
	FORTGI					
	FORTHX					
	COBOL	Output	fn			
	PLIC					
	PLICR					
	PLIOPT	Output	fn			
	TESTCOB	Input	fn	V	121	
LKEDIT	LKED	Output	fn	F	121	Listing
LOGFILE	SET LOGFILE	Output	fn vscreen name	V	none	Log of data written to virtual screen.

Table 8 (Page 6 of 9). **Reserved File types**

File type	**Command**	**Usage**	**File name**	**Format RECFM | LRECL**	**Contents**
LOADLIB	LKED ZAP	Output Input	fn fn	F ≤260	3704/3705 control program load modules
MACLIB	GLOBAL MACLIB MACLIST	Library Input/Output Input/Output	fn fn fn		Macro definitions (dictionary and members)
MACRO	ESERV MACLIB	Input Output	fn fn	F 80	Macro definitions
MAP	DOSLIB	Output	libname	F 80	Library map
	DOSLKED	Output	fn		DOS/VS linkage editor map
	DSERV	Output	DSERV		Directory information from DOS/VS libraries
	INCLUDE	Output	LOAD		Module map
	LOAD	Output	LOAD		Module map
	MACLIB	Output	fn		Library Map
	TXTLIB	Output	fn		Library Map
	TAPE	Output	fn		Library Map
MEMO				F 80	
MODULE	GENMOD LOADMOD MODMAP	Output Input Input	fn fn fn	V	Nonrelocatable object file
NAMES	NAMEFIND NAMES	Input/Output	userid	V 255	Information about users in communication
NETLOG	RECEIVE SENDFILE	Logging	userid	V 255	Records logging transmission of files sent or received
NOTE	NOTE	Input/Output	userid	V 132	Creates a note to be sent to others.

Table 8 (Page 7 of 9). **Reserved File types**					
File type	**Command**	**Usage**	**File name**	**Format RECFM | LRECL**	**Contents**
NOTEBOOK	RECEIVE SENDFILE	Input	userid	V 132	Notes sent to or received by you
PLI or PLIOPT	PLIOPT PLIC PLICR	Input Input Input	fn fn fn	F	PL/I source statements
PROC	PSERV	Output	fn	F 80	A procedure from the DOS/VS procedure library
REPOS	GENMSG	Input	DMKMES DMSMES	F 80	Source statements for message repositories.
RTABLE	PROP	Input	fn	V 72	Routing table for Programmable Operating Facility.
SCRIPT	SCRIPT	Input	fn	V 132	Input to SCRIPT processor
SYMDMP	FCOBOL	Output	fn	V 512	DOS/VS COBOL DEBUG file for SYMDMP option
SYNONYM	SYNONYM	Reference	fn	F 80	Command name synonyms
SYSUT1	ASM3705	Work	fn		
SYSUT2	ASSEMBLE	Work	fn		
SYSUT3	COBOL LKED PLIOPT	Work Work Work	fn fn fn		

Table 8 (Page 8 of 9). **Reserved File types**

File type	Command	Usage	File name	Format RECFM | LRECL	Contents
SYSUT4	COBOL	Work	fn	F 80	Used as input to TESTCOB
	LKED				
	PLIC				
	PLICR				
	TESTCOB	Input		512	
TEMPLATE	CSLGEN	Input	fn	F 80	Template information for parameters of CSL routines
TESTFORT	TESTFORT	Output	fn	VB 125	Processor printed output
TEXT	ASSEMBLE	Output	fn	F 80	Object code
	ASM3705	Output	fn		3704/3705 source code and JCL statements
	COBOL	Output	fn		
	DOSLKED	Input	fn		
	GEN3705	Output	fn (Ln)		Object code
					Object code
					Linkage editor control statements for 3704/3705 control programs
	INCLUDE	Input	fn		
	LKED	Input	fn		
	LOAD	Input	fn		
	PLIOPT	Output	fn		Object code
	TXTLIB	Input	fn		Object code and LKED control cards
	GOFORT	Output	fn		
	FORTGI	Output	fn		
	FORTHX	Output			
	RSERV	Output			Object code
	TEXTFORT	Input	fn		Object code
					Object code
					Object file
TXTlangid	GENMSG	Output	fn	F 80	Object code for language files.
TXTLIB	GLOBAL	Library	fn		Object decks (dictionary and members)
	TXTLIB	Output	fn		

Table 8 (Page 9 of 9). **Reserved File types**

File type	**Command**	**Usage**	**File name**	**Format RECFM | LRECL**	**Contents**
UPDATE	UPDATE	Input	fn	F 80	UPDATE control cards
UPDLOG	UPDATE	Input	fn	F	UPDATE log
VSBASIC	VSBASIC	Input	fn	F ≤256	VSBASIC language source statements
VSBDATA	VSBDATA	Execution time files	fn	V ≤140	VSBASIC user input/output files
UPDTxxxx	UPDATE	Input	fn	F 80	UPDATE control statements
XEDIT	XEDIT	Input	fn	V 255	EXEC/XEDIT statements
ZAP	ZAP ZAPTEXT	Input	fn	F 80	Control records that modify or dump files

File Modes

The file mode *letter* indicates the CMS shared file system directory on the CMS minidisk on which the file resides. The file mode *number* indicates the access mode of the directory or minidisk. The following table summarizes the uses of each file mode.

For more information see the *VM/SP CMS User's Guide*.

Table 9 (Page 1 of 2) **CMS File Modes**

File Mode Number	**Description**
0	No users may access unless they have read/write access to minidisk.
1	For reading and writing; the default file mode.

Table 9 (Page 2 of 2). **CMS File Modes**

File Mode Number	Description
2	Similar to 1; usually assigned to files shared by users linked to a common minidisk.
3	Erased after being read. Should not be used with execs (which may be erased before execution).
4	OS simulated data set format, created by OS macros in programs running in CMS.
5	Similar to 1; can be used as a label, to maintain logical groups.
6	File written back to previous location on minidisk (update-in place attribute of a CMS file) rather than a new one; only applies to files located on 512-. 1K-, 2K-, or 4K-byte block formatted minidisks.

Naming Shared File System (SFS) Directories

The Shared File System, an extension of the CMS file system, provides a means of organizing files into a "tree" structure of multi-level directories. A Shared File System directory name (also called *dirname*) has the following format:

filepoolid:userid.[dir1.dir2.dir3.dir4.dir5.dir6.dir7.dir8]

where:

filepoolid
is the name of the file pool, which must be followed by a colon.

userid
is your user ID and also the name of the top directory. It must be followed by a period.

dir1 through *dir8*
are the names of the subdirectories (maximum eight) under that top directory. Subdirectory names can be up to 16 characters long; the first character must be alphabetic, but the rest can be A-Z or 0-9. Two or more subdirectories may have the same name as long as each

of the subdirectories has a different parent directory. Subdirectory names must be separated by periods.

Specifying Directory Identifiers

SFS commands that accept a directory identifier (*dirid*) will accept a full directory name, a file mode, or plus (+) and minus (-) file mode letter notation to specify a directory. A file mode consists of a letter and number in commands that refer to a file or set of files; it consists of a letter only in commands that refer to an entire directory or minidisk.

Instead of using the entire directory name, plus and minus file mode letter notation lets you use a plus sign (+) before the file mode letter of a specific directory to move down one level lower in the hierarchy, or a minus (-) sign to move up one level.

In the above example, if *dir1,dir2 and dir3* are accessed as file mode letters B, C and E respectively, specifying file mode −C in a command would refer to *dir1*, +C would refer to *dir3*, and so forth. See the *VM/SP CMS Command Reference* or the *VM/SP CMS User's Guide* for more information.

Full-Screen CMS Default Settings

Full-screen CMS allows you to customize your session by tailoring virtual screens and the windows used to look at them. The following tables describe the default virtual screens and windows available in full-screen CMS. The route definition table shows how VM/SP messages are routed through virtual screens. For information about overriding these defaults, see the *VM/SP CMS User's Guide*.

Virtual Screen Default Settings

Default virtual screens are SYSTEM and TYPE virtual screens. This means that data is moved to the virtual screen when it is updated, and that it is retained when the system abnormally terminates (abends) or when an HX (halt execution) command is entered.

Table 10. **Default Virtual Screens**

Virtual Screen	**Number of Lines**	**Number of Columns**	**Reserved Top Lines**	**Reserved Bottom Lines**	**Default Color**	**Options**
WM	1	Physical screen size	0	5	White	NOPROTECT
STATUS	1	Physical screen size	0	0	White	PROTECT
NETWORK	16	70	2	0	Blue	PROTECT
WARNING	4	70	2	0	Red	PROTECT
MESSAGE	20	70	2	0	White	PROTECT
CMS	120	Physical screen size	2	5	Green	NOPROTECT

NOTES:

PROTECT You cannot type into the window(s) connected to the virtual screen because the data is protected.

NOPROTECT You can type into the window(s) connected to virtual screen; the data is not protected.

Message Routing with Full-Screen CMS

When SET FULLSCREEN is ON, the various message classes are routed to virtual screens according to the following table. For information on overriding these defaults, see the ROUTE command in the *VM/SP CMS Command Reference*.

Table 11. **Default Settings for Message Routing**

Message Class	**Virtual Screen**	**Options**
CMS	CMS	NOALARM NONOTIFY
CP	CMS	NOALARM NONOTIFY
MESSAGE	MESSAGE	ALARM NOTIFY
WARNING	WARNING	ALARM NOTIFY
SCIF	MESSAGE	NOALARM NONOTIFY
NETWORK	NETWORK	NOALARM NOTIFY

Window Default Settings

Default windows are SYSTEM windows. This means they are retained when the system abnormally terminates (abends) or when an HX (halt execution) command is entered.

Although the WM window is a default window is a default window, it is not defined when you enter full-screen CMS. It is defined when you enter the POP WINDOW WM command, press the PA1 key, or when it is automatically displayed on your screen.

Table 12. **Default Windows**

Window	Lines	Columns	Psline	Pscol	Options
STATUS	1	Physical screen size	-1	1	FIXED NOBORDER NOPOP NOTOP
CMS	Physical screen size	Physical screen size	1	1	FIXED BORDER NOPOP TOP
NETWORK	8 (max.)	71	-12	7	VARIABLE BORDER NOPOP TOP
WARNING	6 (max.)	71	3	3	VARIABLE BORDER POP TOP
MESSAGE	8 (max.)	71	11	3	VARIABLE BORDER POP TOP
WM	5	Physical screen size	-1	1	FIXED BORDER NOPOP NOTOP
CMSOUT	8	75	9	3	VARIABLE BORDER POP TOP

NOTES:

Pscr	Size of the physical screen.
Psline	Line on the physical screen where the upper (when psline is positive) or lower (when psline is negative) corner of the window will be placed.
Pscol	Column on the physical screen where the upper left corner oι the window will be placed.

FIXED	Window has constant number of lines.
VARIABLE	Number of lines in window may vary from zero to the maximum, depending on how much scrollable data is to be displayed.
BORDER	Window borders are displayed when possible.
NOBORDER	Window borders not displayed.
POP	Window is displayed on top of all other windows when the virtual screen that the window is showing is updated.
NOPOP	Window stays in same position (in ordered list of windows) when the virtual screen that the window is showing is updated.
TOP	Window may qualify as the topmost window.
NOTOP	Window cannot qualify as a topmost window.

CP Command Privilege Classes

Each CP command has one or more user privilege classes associated with it. The privilege classes govern access to the commands. There are seven user classes (plus “class ANY”), whose commands perform seven general types of functions.

The following table shows the eight privilege classes, their associated function codes, and major tasks that each class of user can perform.

VM/SP Reference Information

Table 13. **CP Privilege Classes**

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

Chapter 2. CMS, CP, RSCS, TSAF, AVS, IPCS and GCS Commands

The rest of this book contains CMS, CP, RSCS, TSAF, AVS, IPCS and GCS commands, and VM/SP service aids. Each command is briefly described and its format shown. For detailed explanations of commands, refer to the VM/SP reference manuals pertaining to the command type. (The command type, underlined and in bold print, is located directly beneath each command name.)

AVS	*Connectivity Planning, Administration, and Operation,* SC24-5378
CMS	*CMS Command Reference,* SC19-6209 *Application Development Guide for CMS,* SC24-5286 *Administration,* SC24-5285 *EXEC 2 Reference,* SC24-5219 *Installation Guide,* SC24-5237 *System Facilities for Programming,* SC24-5288 *CMS Shared File System Administration,* SC24-5367
REXX	*System Product Interpreter User's Guide,* SC24-5238 *System Product Interpreter Reference,* SC24-5239
XEDIT	*System Product Editor Command and Macro Reference,* SC24-5221
CP	*CP General User Command Reference,* SC19-6211 *CP System Command Reference,* SC24-5402
GCS	*Group Control System Command and Macro Reference,* SC24-5250
IPCS	*Interactive Problem Control System Guide and Reference,* SC24-5260

RSCS	*Operation and Use, SH24-5058* **Note:** RSCS commands must be preceded by RSCS if entered by the console operator, or by SMSG RSCSvmid if entered by a system-authorized alternative operator.
TSAF	*Connectivity Planning, Administration, and Operation, SC24-5378* **Note:** TSAF commands are either entered by the TSAF virtual console or by the secondary user of the TSAF virtual machine. If entered by the secondary user, commands must be preceded by SEND USERID,. where USERID is that of the disconnected TSAF virtual machine.

---

*

CMS,CP Class Any, GCS, RSCS

Permits comments.

```
*               anycomment
```

#CP

CP Class Any

Executes a CP command while in a virtual machine command environment without first signaling attention to get to the CP command environment.

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

ACCESS

CMS

Allows you to access minidisks or SFS directories with a file mode letter.

```
ACcess  ┌ dirid fm [ /ext]                           [(optionsA...[)]] ┐
        │                                                              │
        │ vdev  fm  [/ ext [fn   [ft   [fm  ]]]]     [(optionsB...[)]] │
        └                  [*    [*    [*   ]]]]                       ┘

        OptionsA: [NOPROF] [NODISK]
        OptionsB: [NOPROF] ┌ERASE    ┐ [NODISK]
                           │SAVEONLY │
                           └NOSAVE   ┘
```

ACCESS

GCS

Identifies CMS or VSAM disks that an application will use.

Unlike the CMS ACCESS command, you cannot specify options, and you cannot have an 800 byte blocksize.

```
ACcess  ┌cuu mode[ /ext [fn [ft [fm ]]]]┐
        └191  A                         ┘
```

ACNT

CP Class A

Creates accounting records.

```
ACNT      ⎧userid1   [userid2 ...]⎫
          ⎨ALL [CLOSE]            ⎬
          ⎩CLOSE                  ⎭
```

ADD LINK

TSAF

Identifies a communication link to TSAF when the TSAF virtual machine is running. Only the TSAF virtual console or the secondary user of the TSAF virtual machine can issue this command.

```
ADD     LINK vdev
```

ADSTOP

CP Class G

Halts the execution of the virtual machine.

```
ADSTOP ⎧hexloc⎫
       ⎩OFF   ⎭
```

AGW ACTIVATE GATEWAY

AVS

Defines a gateway LU in the TSAF collection to VTAM and VM.

```
AGW ACTIVATE GATEWAY  gateway {GLOBAL                  }
                              {PRIVATE [USERID userid] }
```

AGW CNOS

AVS

Sets the session limit and contention winner polarity values for communications between a gateway (local LU) and a remote LU.

```
AGW CNOS   gateway remotelu modename seslimit conwin conlose   [DRAIN  ]
                                                               [NODRAIN]
```

AGW DEACTIVATE CONV

AVS

Deactivates a conversation on a gateway.

```
AGW DEACTIVE CONV   gateway convid
```

AGW DEACTIVATE GATEWAY

AVS

Deactivates an activated gateway.

```
AGW DEACTIVE GATEWAY    gateway [FORCE]
```

AGW QUERY

AVS

Displays information about various settings and conditions of the AVS application.

```
AGW QUERY  ⎧ GATEWAY ⎡gateway⎤                   ⎫
    Q      ⎪         ⎣ALL    ⎦                   ⎪
           ⎪ CNOS  ⎡remotelu ⎡AT gateway⎤⎤       ⎪
           ⎪       ⎢         ⎣AT ALL    ⎦⎥       ⎪
           ⎪       ⎢ALL AT gateway       ⎥       ⎪
           ⎨       ⎣ALL                  ⎦       ⎬
           ⎪ CONV  ⎡GATEWAY gateway  ⎤           ⎪
           ⎪       ⎢REMOTELU remotelu⎥           ⎪
           ⎪       ⎢USERID userid    ⎥           ⎪
           ⎪       ⎣ALL              ⎦           ⎪
           ⎪ ETRACE                              ⎪
           ⎪ ITRACE                              ⎪
           ⎩ ALL                                 ⎭
```

AGW QUIESCE

AVS

Ends AVS when the last existing conversations have completed.

```
AGW QUIESCE
```

AGW SET ETRACE

AVS

Enables and disables external tracing.

Note: External tracing will only be in effect if internal tracing is set on.

```
AGW SET    ETRACE {ON }
                  {OFF}
```

AGW SET ITRACE

AVS

Enables and disables internal tracing.

Note: Must be set on to enable external tracing.

```
AGW SET    ITRACE {GATEWAY gateway}{ON }
                  {ALL            }{OFF}
```

AGW START

AVS

Calls an AVS application.

```
AGW START   [nnnn][ETRACE]
            [40  ]
```

AGW STOP

AVS

Ends an AVS application immediately. All existing conversations through AVS will be deactivated.

AGW STOP

ALARM VSCREEN

CMS

Sounds the terminal alarm the next time the display is refreshed.

ALARM **VSCreen** *vname*

AMSERV

CMS

Uses access method services utility functions to create, alter, list, copy, delete, import, or export VSAM catalogs and data sets.

```
AMserv     fn1  [fn2]   [(options...[)]]
                [fn1]

           Options:  [PRINT]  [TAPIN {18n }]  [TAPOUT {18n }]
                              [      {TAPn}]  [       {TAPn}]
```

APAR

IPCS

Generates APAR documentation for submission to IBM. The documentation can either be printed or dumped to tape.

```
APAR
```

ASMGEND

CMS

Regenerates the VM/SP assembler command modules (system programmers only).

```
ASMGEND
```

ASM3705

CMS

Assembles 370x source code.

Note: All of the options of the 3705 XF assembler are supported and may be used with the ASM3705 command, with the exception of ALIGN | NOALIGN and TEST | NOTEST.

```
ASM3705    fn  [(options ...)]

           Options:

           [XREF (FULL ) ]  [RENT  ]   [DECK  ]  [LOAD  ]
           [XREF (SHORT) ]  [NORENT]   [NODECK]  [NOLOAD]
           [NOXREF       ]

           [LIST  ]    [LINECOUN (55)]  [PRINT  ]
           [NOLIST]    [LINECOUN (nn)]  [DISK   ]
                                        [NOPRINT]
```

ASSEMBLE

CMS

Assembles assembler language source code.

```
Assemble    fn      [( options... [)] ]
            Listing Control Options:
             [ALOGIC  ] [ESD  ] [FLAG  (nnn)] [LINECOUN (nn)]
             [NOALOGIC] [NOESD] [FLAG  (0)  ] [LINECOUN (55)]

             [LIST  ] [MCALL  ] [MLOGIC  ] [RLD  ] [LIBMAC  ]
             [NOLIST] [NOMCALL] [NOMLOGIC] [NORLD] [NOLIBMAC]

             [XREF  (FULL) ] [PRINT  ]
             [XREF  (SHORT)] [NOPRINT]
             [NOXREF       ] [DISK   ]
            Output Control Options:
             [DECK  ] [OBJECT  ] [TEST  ]
             [NODECK] [NOOBJECT] [NOTEST]
            SYSTERM Options:
             [NUMBER] [STMT  ] [TERMINAL]
             [NONUM ] [NOSTMT] [NOTERM  ]
            Other Assembler Options:
             [ALIGN  ]   [BUFSIZE (MIN)]    [RENT  ]
             [NOALIGN]   [BUFSIZE (STD)]    [NORENT]
                         [BUFSIZE (MAX)]

             [YFLAG  ]   [SYSPARM (string)]  [WORKSIZE (2048K)  ]
             [NOYFLAG]   [SYSPARM ()      ]  [WORKSIZE (nnnnnK)]
                         [SYSPARM (?)     ]
```

ASSGN

CMS

Assigns or unassigns a CMS/DOS system or programmer logical unit for a virtual I/O device.

```
ASSGN     SYSxxx  {Reader    }        [(options... [ ) ] ]
                  {PUnch     }
                  {PRinter   }
                  {Terminal  }
                  {TAP  [n]  }
                  {     [1]  }
                  {mode      }
                  {IGN       }
                  {UA        }

          Options: [UPCASE  ] [7TRACK] [TRTCH a] [DEN den]
                   [LOWCASE ] [9TRACK]
```

ATTACH

CP Class B

Attaches a real device to a specified user or to the system.

```
ATTach   { raddr [TO] {userid  [AS] vaddr   [R[/O]]}  [3330V] [VOLID volid] }
         {            {SYSTEM  [AS] volid          }                        }
         {            {*                           }                        }
         {                                                                  }
         { {raddr ...    } [TO] {userid} [R[/O]] [3330V]                    }
         { {raddr-raddr  }      {*     }                                    }
         {                                                                  }
         { L addr  [TO]  {userid}  [AS]  vaddr                              }
         {               {*     }                                           }
         {                                                                  }
         { CHANnel c  [PROC nn]  [TO]  {userid}                             }
         {                             {*     }                             }
```

ATTN

CP Class G

Makes attention interruption pending.

```
ATTN
```

AUDIT

CMS

Starts or stops file pool server security audit trace processing. (File pool server operator only.)

```
AUDIT  ⎧ ON   ⎧PARTIAL⎫  ⎫
       ⎪      ⎩ALL    ⎭  ⎪
       ⎨                 ⎬
       ⎪ OFF  ⎡CLOSE  ⎤  ⎪
       ⎩      ⎣NOCLOSE⎦  ⎭
```

AUTOLOG

CP Class A and B

Logs on any virtual machine defined in the VM/370 directory.

```
AUTOLOg    userid password  [variable data]
```

B

CMS Border Command

Scrolls the window backward.

```
B
```

BACKSPAC

CP Class D

Restarts or repositions a current spool file.

Printer Format:

```
BAckspac      { raddr }  [ File              ]
              { lprt  }  [ pages    [EOF ]   ]
                         [   1               ]
```

Punch Format:

```
BAckspac          raddr    [File]
```

BACKSPAC

RSCS

Restarts or repositions in a backward direction the file currently being transmitted. This command is for RJE, 3270P, SNA3270P, and MRJE type links.

```
BAckspac      [linkid]    [ File ]
                          [ nnn  ]
```

BACKUP

CMS

Starts a backup of the control data while multiple user mode processing continues. (File pool server operator only.)

```
BACKUP
```

BEGIN

CP Class G

Starts the execution of a virtual machine.

```
Begin          [hexloc ]
```

C

CMS Border Command

Clears the window of scrollable data.

```
C
```

CATCHECK

CMS

Allows a CMS VSAM user (with or without DOS set ON) to use the VSE/VSAM Catalog Check Service Aid to verify a complete catalog structure.

```
CATCHECK     [catname          ]
             [catname/password ]
```

CHANGE

CP Class D

Alters the attributes of a closed spool file.

```
                                                       ┌                                   ┐1,2
                                                       │ CHars name0  [[CHars] name1]      │
                                                       │              [[CHars] name2]      │
                                                       │              [[CHars] name3]      │
                                                       │ CLass c2                          │
                                                       │ COpy [*] nnn                      │
                                                       │ DIst distcode                     │
                [userid ] {Reader }  {CLass c1    }    │ FCB name 2                        │
CHange          [SYSTEM ] {Printer}  {FORM form1  }   <  FLash name nnn                     >
                [  *    ] {PUnch  }  {DEST dest1  }    │ FORM form2                        │
                                     {spoolid     }    │ DEST dest2                        │
                                     {ALL         }    │ [HOLD  ]                          │
                                                       │ [NOHOLD]                          │
                                                       │ MOdify name [n] 2                 │
                                                       │ [SYS  ]                           │
                                                       │ [NOSYS]                           │
                                                       │ UNCONV                            │
                                                       └                                   ┘
                                                            [NAme {fn [ft] }]
                                                            [     {dsname  }]
```

[1] One of these options must be chosen; however, more than one may be specified and they may be in any sequence.

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

CHANGE

CP Class G

Alters the attributes of a closed spool file.

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

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

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

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

CHANGE

RSCS

Alters one or more attributes of an inactive spool file.

General User Format:

```
CHange       [*]spoolid       options ...
```

Operator Format:

```
CHange       [ linkid ]   spoolid       options ...

[ PRIority  nn ]
[ CLass  c ]
[ COpy  [* ] nnn ]
[ DIst  distcode ]

[ HOld    ]
[ NOHold  ]

[ FLash  name nnn ]
[ MOdify  name [trc ]]
[ CHars  name1   [.. CHars   name4 ]]
[ FCB  name ]
[ FOrm   cccccccc ]

[ DEST { cccccccc } ]
[      { OFF      } ]

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

CLEAR VSCREEN

CMS

Erases data in the virtual screen by overwriting the data buffer with nulls.

```
CLEAR VSCreen       vname
```

CLEAR WINDOW

CMS

Scrolls past all data in the virtual screen to which the window is connected so that no data is displayed in the data area of the window.

```
CLEAR WINdow    [ wname ]
                [   =   ]
```

CLOSE

CP Class G

Terminates spooling operations on a virtual reader, printer, or punch.

```
Close   ( [Reader                    ]                                              )
        ( [vaddr   [HOld  ]          ]                                              )
        ( [        [NOHold]          ]                                              )
        (                                                                           )
        { [CONsole [PUrge                                                       ]]  }
        { [Printer [FORM form                                                   ]]  }
        { [PUnch   [DEST dest                                                   ]]  }
        { [vaddr   [                                                            ]]  }
        ( [        [[HOld  ][DIst distcode][NAme {fn [ft] }                     ]]  )
        ( [        [[NOHold]               [     {dsname  }                     ]]  )
```

CMD

RSCS

Forwards a command line to a remote system for execution.

```
CMD      nodeid     [ command text ]
```

CMDCALL

CMS

Converts EXEC 2 extended plist function calls to CMS extended plist command calls.

```
CMDCALL      [cmd [operand1 [operand2 ... operandn ]]]
```

CMSBATCH

CMS

Calls the CMS batch facility, creating a virtual machine running in batch mode.

```
CMSBATCH     [sysname ]
```

CMSGEND

CMS

Generates a new CMS module from updated TEXT files.

```
CMSGEND      fn   [CTLCMS   ]  [MODE  fm]
                  [CTLALL   ]  [A       ]
                  [NOCLEAR  ]
                  [MAP      ]
                  [NOINV    ]
```

CMSSERV

CMS

Starts IBM Enhanced Connectivity Facilities communications between your VM/SP host system and your work station (IBM Personal Computer).

```
CMSSERV    [ ( options... [ ) ] ]

           options:   [ CUT ]
                      [ DFT ]
```

COMMANDS

CP Class Any

Lists the commands and diagnose codes you are authorized to use.

```
COMMands
```

COMPARE

CMS

Compares records in CMS disk files.

```
COMpare    fileid1   fileid2    [ (option... [ ) ] ]

               Option:  [ COL    mmm [nnn] ]
                        [ COL    mmm-nnn   ]
```

CONVERT

IPCS

Converts VM/370 Symptom Summary files and PRBnnnnn to the format required by VM/SP IPCS.

```
CONVERT     fn [ft [fm]]
```

CONVERT COMMANDS

CMS

Converts a CMS file containing Definition Language for Command Syntax (DLCS) statements into an internal form for the parsing facility.

```
CONVert COMmands   [fn  [ft     [fm]]]  [(options... [)]]
                        [DLCS]  [* ]

                   Options:  [SYStem]  [CHeck              ]
                             [USER  ]  [OUTmode      [fm]  ]
                             [ALL   ]  [             [* ]  ]

                             [STACK     [FIFO]]
                             [          [LIFO]]
                             [FIFO            ]
                             [LIFO            ]
```

CONVIPCS

IPCS

Converts PVM Release 2 or RSCS Release 3 help files to format required for VM/SP IPCS usage.

```
CONVIPCS
```

CONWAIT

CMS

Causes the program to wait until all pending terminal I/O is complete.

CONWAIT

COPYFILE

CMS

Copies CMS files from one minidisk to another, one SFS directory to another, or between minidisks and directories.

```
COPYfile   fileidi1   [fileidi2... ] [fileido ] [(options... [ )] ]
           Options:

           [Type   ]   [NEWDate]   [NEWFile]   [PRompt   ]
           [NOType ]   [OLDDate]   [REPlace]   [NOPRompt ]

           [FRom    recno    ]   [FOR     numrec   ]   [SPecs  ]
           [FRLabel xxxxxxxx ]   [TOLabel xxxxxxxx ]   [NOSPecs]

           [OVly  ]   [RECfm {F}]   [Lrecl  nnnn ]   [TRUnc  ]
           [APpend]   [      {V}]                    [NOTRunc]

           [PAck  ]   [FIll c  ]   [EBcdic]   [UPcase ]   [TRAns ]
           [UNPack]   [FIll hh ]              [LOwcase]
                      [FIll 40 ]

           [SIngle]
```

COUPLE

CP Class G

Connects virtual channel-to-channel adapters.

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

CP

CMS

Enters CP commands from the CMS environment.

```
CP        [commandline]
```

CP

CP Class Any

Permits execution of CP commands within your privilege class.

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

CP

RSCS

Executes a command line as a VM/370 Control Program (CP) console function without leaving the RSCS command environment (for RSCS operator only).

```
CP      command text
```

CPQUERY

RSCS

Requests status information from CP, similar to a VM/370 CP QUERY command.

```
          ⎧ CPUid            ⎫
          ⎪ CPLEVEL          ⎪
          ⎪ INDicate         ⎪
CPQuery   ⎨ LOGmsg           ⎬
          ⎪ Names            ⎪
          ⎪ Time             ⎪
          ⎩ Users  [userid]  ⎭
```

CPTRAP

CP Class C

Creates a file of trace table and CP and virtual machine interface records in the order they happen for problem determination.

```
CPTrap  ID trapid  ┌SET ┌trapset┐┐
                   └    └NULL   ┘┘

┌TYPE ┌DATA  ┌LOC hexloc instruction                                                          ┐┐┐
│     │      │DL ┌┌addr┐                ┌  ┌dddd┐                ┐              ┐             ││││
│     │      │   ││Gn  │ [%[%]... ]     │+ │Gn  │  [%[%]... ]    │ ...[.lll]    │ [= name ]   ││││
│     │      │   ││Xn  │                │- │Xn  │                │              │             ││││
│     │      │   │└    ┘                └  └    ┘                ┘              │             ││││
│     │      │   │┌Gn┐:m                                                        │             ││││
│     │      │   │└Xn┘                                                          │             ││││
│     │      └   └                                                              ┘             ┘│││
│     │                                                                                        │││
│     │ GT   ┌ALLOWid  userid        ┌3D ┐┐                                                    │││
│     │      │GRoupid  group-name    │3E ││                                                    │││
│     │      │                       │ALL││                                                    │││
│     │      │                       └   ┘│                                                    │││
│     │      │3D                          │                                                    │││
│     │      │3E                          │                                                    │││
│     │      │ALL                         │                                                    │││
│     │      └                            ┘                                                    │││
│     │                                                                                        │││
│     │ IO   ┌DEVice   raddr [-raddr][raddr [-raddr]... ]   ┐                                  │││
│     │      │                                              │                                  │││
│     │      │USER     ┌userid┐                             │                                  │││
│     │      │         └ALL   ┘                             │                                  │││
│     │      │IOData   ┌nnnn┐                               │                                  │││
│     │      │         └0   ┘                               │                                  │││
│     │      └                                              ┘                                  │││
│     │                                                                                        │││
│     │ TTable ┌INTable  ┌ALL       ┌ON ┐       ┐                         ┐                    │││
│     │        │         │          └OFF┘       │                         │                    │││
│     │        │         │typenum   ┌ON ┐       │                         │                    │││
│     │        │         └          └OFF┘...    ┘                         │                    │││
│     │        │INFile   ┌ALL       ┌ON ┐                              ┐  │                    │││
│     │        │         │          └OFF┘                              │  │                    │││
│     │        │         │          ┌ON                             ┐  │  │                    │││
│     │        │         │          │OFF                            │  │  │                    │││
│     │        │         │typenum   │Vmblok nnnnnn                  │  │  │                    │││
│     │        │         │          │DEVaddr {raddr|vaddr}...       │  │  │                    │││
│     │        │         │          │COde nnnn                      │  │  │                    │││
│     │        │         │          └                               ┘  │  │                    │││
└     └        └         └                                             ┘  ┘                    ┘┘┘
```

(format continued on the next page)

CPTRAP

(format continued from the previous page)

```
    TO    {userid}  [WRAP nnnnn]
          {  *   }

    CLOSE

ENable   ┌[ID [t1[t2...tx]] [SET s1 [s2...sy ]]]┐
DISAble {│                                      │}
DROp     └ALL                                   ┘

STOP

[other operands] DISPlay
```

CREATE ALIAS

CMS

Places an additional name for a file in a specified directory.

```
CREate ALIas       {fn1} {ft1} dirid1 {fn2} {ft2} {dirid2} [(options...[)]]
                   { * } { * }        { = } { = } {  =   }

                   Options:  ┌TYPe                ┐
                             │NOType              │
                             │STACK ┌FIFO┐        │
                             │      └LIFO┘        │
                             │LIFO                │
                             └FIFO                ┘
```

CREATE DIRECTORY

CMS

Creates an SFS directory.

```
CREate DIRectory     dirid
```

CREATE LOCK

CMS

Creates an explicit lock on a file or a directory.

```
CREate LOCk     [fn   ft]  dirid  {SHAre    }  {SESsion}   [(options...[)]]
                [*    * ]         {EXClusive}  {LASting}
                                  {UPDate   }

                Options:  [TYPe               ]
                          [NOType             ]
                          [STACK [FIFO]       ]
                          [      [LIFO]       ]
                          [LIFO               ]
                          [FIFO               ]
```

CREATE NAMEDEF

CMS

Assigns a temporary name for a user which can be used by a program, instead of a file name and file type or a fully-qualified directory name.

```
CREate NAMedef   {fn ft}   namedef    [(options...[)]]
                 {dirid}

                 Options:  [REPlace]
```

CSLGEN

CMS

Builds a callable services library from control files, text files and template files.

```
CSLGEN    {DASD} library FROM fn [ft       [dirid]] [(options... [)]]
          {SEG }                 [CSLCNTRL [  *  ]]

                options:

                [TO  dirid]     [ REPlace   ]
                [TO  A    ]     [ NOREPlace ]
```

CSLLIST

CMS

Lists information about all members of a specified callable services library, with the ability to issue certain commands (RTNLOAD, RTNMAP, RTNSTATE, or RTNDROP) from the list's command area.

```
CSLList     libname    [(options... [)]]

            Options:   [IN fm            ]
                       [IN dirid         ]
                       [SEGment segid    ]
                       [ Append ]
                       [PROFile fn]
```

A **Special command** that can be used in the CSLLIST environment, followed by it's description, is:

```
EXECUTE   [Cursor]  [command]
          [lines ]
```

Issues CP/CMS commands (or EXECs) that make use of files displayed by CSLLIST.

CURSOR VSCREEN

CMS

Positions the cursor on a specified line and column in a virtual screen.

```
CURsor VSCreen    vname  line  col    [(options [)]]

                  Options:   [Reserved]
                             [Data    ]
```

D

CMS Border Command

Drops the window.

```
D
```

DCP

CP Class C and E

Displays real processor storage on the terminal.

```
DCP       ┌                                            ┐
          │ ┌                 ┐  ┌                    ┐ │
          │ │ ML hexloc 1     │  │ {-} [hexloc 2 ]    │ │
          │ │ NL hexloc 1     │  │ {:} [END      ]    │ │
          │ │ MT hexloc 1     │  │                    │ │
          │ │ NT hexloc 1     │  │                    │ │
          │ │  M hexloc 1     │  │                    │ │
         {  │  N hexloc 1     │  │ {.} [bytecount]    │ │ }
          │ │  L hexloc 1     │  │     [END      ]    │ │
          │ │  T hexloc 1     │  │                    │ │
          │ │    hexloc 1     │  │                    │ │
          │ │       L         │  │                    │ │
          │ │       T         │  │                    │ │
          │ │       0         │  │                    │ │
          │ └                 ┘  └                    ┘ │
          └                                            ┘
```

DCSSGEN

CMS

Builds the CMS installation saved segment (CMSINST).

```
DCSSGEN   fn   ft   fm   [segname ]
                         [CMSINST ]
```

DDR

CMS

Performs backup, restores, and copies operations for entire DASD volumes or minidisks.

```
DDR        [fn  ft [fm]]
                   [* ]
```

I/O Definition Statements:

```
INput       cuu  type  [  volser  ]   [(options...)]
OUTput                 [  altape  ]
                       [ SCRATCH  ]

                Options:

            [SKip nn]  [MOde 6250]  [REWIND]  COmpact
            [SKip 0 ]  [MOde 1600]  [UNload]
                       [MOde  800]  [LEave ]
                       [MOde  38K]
```

SYSPRINT Control Statement:

```
SYsprint      [ cuu  ]
              [ CONS ]
              [ 00E  ]
```

Function Control Statements:

```
DUmp  [FTr    ] [block1  [TO ]  [block2    [Reorder ]  [To ] [block3 ]]]
      [OLDFORM] [cyl1           [cyl2                        [cyl3   ]]]
                [CPvol                                                 ]
                [ALL                                                   ]
                [NUcleus                                               ]
```

DDR

```
COpy      ⎡block1    [TO ]  ⎡block2   [Reorder]  [To] ⎡block3⎤ ⎤ ⎤
REstore   ⎢cyl1             ⎣cyl2                     ⎣cyl3  ⎦ ⎦ ⎥
          ⎢CPvol                                                ⎥
          ⎢ALL                                                  ⎥
          ⎣NUcleus                                              ⎦
```

Note: The FTr option is valid only with the DUMP control statement.

PRINT/TYPE Function Statements:

```
PRint     cyl1 [hh1 [rr1]]  [To   cyl2 [hh2 [rr2] ]]  [(options...)]]
TYpe

          block1     [To   block2]    [(options...)]

                    Options:

                  [Hex][Graphic][Count]
```

DEBUG

CMS

Displays state of virtual machine at time of abend.

```
DEBUG
```

DEFAULTS

CMS

Sets or displays default options for various commands.

```
DEFAULTS  ⎡Set  command options...  ⎤
          ⎣List [command ]          ⎦
```

DEFINE

CP Class A and B

Redefines the status of a 3330V volume.

```
DEFine            {Sysvirt}     raddr1    [-raddr2]
                  {Virtual}
```

DEFINE

CP Class G

Reconfigures the user's virtual machine or channel operating mode.

```
DEFine   { CONsole  }
         { CTCa     }
         { Printer  }
         { PUnch    }
         { Reader   }
         { TIMer    }
         { 1403     }
         { 1443     }
         { 2501     }
         { 2540P    }  [As]  vaddr
         { 2540R    }
         { 3088     }
         { 3203     }
         { 3211     }
         { 3262     }
         { 3289E    }
         { 3505     }
         { 3525     }
         { 4245     }
         { 4248     }

         { 3800   }                          [2Wcgm]  [BTS]  [DAtck  ]
         { 3800-1 } [As]  vaddr  [Size ww ll] [4Wcgm]  [CFS]  [NODatck]
         { 3800-3 }

           CHANnels  [As]   {SEL}
                            {BMX}

           GRAF    cuu     [3036]
                           [3066]
                           [3138]
                           [3148]
                           [3158]
                           [3270]

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

           Printer  [As]   vaddr  [1403  ]
           PRT                    [1443  ]
                                  [3203  ]
                                  [3211  ]
                                  [3262  ]
                                  [3289E ]
                                  [3800  ]
                                  [3800-1]
                                  [3800-3]
                                  [4245  ]
                                  [4248  ]
```

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

DEFINE

RSCS

Temporarily adds a new link definition to the RSCS link table, or temporarily alters an existing link definition (for RSCS operator only).

```
DEFine     linkid   [{ ASTart   }]
                    [{ NOASTart }]

                    [CLass c]
                    [DP dpriority]
                    [KEEP holdslot]
                    [LINE vaddr]
                    [LOGMode logmodename]
                    [LUName luname]

                    [       { Priority } ]
                    [ Queue { Fifo     } ]
                    [       { Size     } ]

                    [{ RETry   }]
                    [{ NORETry }]

                    [      { ASCII     } ]
                    [      { LISTPROC  } ]
                    [      { MRJE      } ]
                    [      { NJE       } ]
                    [ TYPE { RJE       } ]
                    [      { SNANJE    } ]
                    [      { SNARJE    } ]
                    [      { SNA3270P  } ]
                    [      { 3270P     } ]

                           [Parm [parameters ...]]
```

DEFINE VSCREEN

CMS

Creates a virtual screen.

```
DEFine VSCreen   vname  lines  cols  rtop  rbot   [(optionA optionB optionC optionD[)]]

                 OptionA:  [TYPe  ]
                           [NOType]

                 OptionB:  [PRotect  ]   [High  ]
                           [NOProtect]   [NOHigh]

                 OptionC:  [color]  [exthi]  [psset]

                 OptionD:  [USer  ]
                           [SYstem]
```

DEFINE WINDOW

CMS

Creates a window.

```
DEFine WINdow    wname  lines  cols  psline  pscol   [(options [)]]

                 Options:  [VARiable]  [BORder  ]  [POP  ]
                           [FIXed   ]  [NOBorder]  [NOPop]

                           [TOP  ]     [USer  ]
                           [NOTop]     [SYstem]
```

DELETE **RSCS**

Temporarily deletes a link definition from the RSCS link table. (For RSCS operator only.)

```
DELete     linkid
```

DELETE ADMINISTRATOR **CMS**

Removes administrator authority for the specified Shared File System file pool, from the specified user ID. (For use by file pool administrator only.)

```
DELete ADMinistrator   { userid   } [ filepoolid: ] [ (options... [ )] ]
                       { nickname }

                       Options:   [ TYPe                 ]
                                  [ NOType               ]
                                  [ STACK  [ FIFO ]      ]
                                  [        [ LIFO ]      ]
                                  [ LIFO                 ]
                                  [ FIFO                 ]
```

DELETE LINK **TSAF**

Removes a communication link from the TSAF table of communication links when the TSAF virtual machine is running. Only the TSAF virtual console or the secondary user of the TSAF virtual machine can issue this command.

```
DELETE   LINK   vdev
```

DELETE LOCK

CMS

Releases the explicit lock placed on a file or directory by the CREATE LOCK command.

```
DELete LOCk   [fn  ft]  dirid     [(options...[)]]
              [ *   *]

              Options:  [FROM  userid]
                        [TYPe               ]
                        [NOType             ]
                        [STACK  [FIFO]      ]
                        [       [LIFO]      ]
                        [LIFO               ]
                        [FIFO               ]
```

DELETE NAMEDEF

CMS

Deletes the temporary name given to a user by the CREATE NAMEDEF command, and makes it no longer usable by a program.

```
DELete NAMedef     {namedef}
                   {   *   }
```

DELETE PUBLIC

CMS

Removes the connect authority given to public on the ENROLL PUBLIC command. (For use by file pool administrator only.)

```
DELete PUBlic          [filepoolid:]
```

DELETE USER

CMS

Removes a user from the specified file pool. (For use by file pool administrator only.)

```
DELete USEr        { userid   } [filepoolid:] [(options... [ )]]
                   { nickname }

                   Options:  [ TYPe                 ]
                             [ NOType               ]
                             [ STACK [FIFO]         ]
                             [       [LIFO]         ]
                             [ LIFO                 ]
                             [ FIFO                 ]

                             [ CONFirm              ]
                             [ NOCONFirm            ]
```

DELETE VSCREEN

CMS

Removes a virtual screen definition.

```
DELete VSCreen     vname
```

DELETE WINDOW

CMS

Removes a window definition.

```
DELete WINdow      wname
```

DESBUF

CMS

Clears the program stack and the terminal input buffers.

```
DESBUF
```

DETACH

CP Class B

Removes a real device from the system or from a specific user.

```
DETach   ⎧ ⎧raddr      ⎫  [FROM]  ⎧userid⎫  [UNload]  ⎫
         ⎪ ⎨raddr...   ⎬          ⎨SYSTEM⎬  [LEave ]  ⎪
         ⎪ ⎩raddr-raddr⎭          ⎩  *   ⎭            ⎪
         ⎨                                            ⎬
         ⎪ L addr   [FROM ]  {userid}                 ⎪
         ⎪                   {  *   }                 ⎪
         ⎪ CHANnel c   [PROC nn] [FROM]  {userid}     ⎪
         ⎩                               {  *   }     ⎭
```

UNLoad and LEave can be used with tape devices only.

DETACH

CP Class G

Removes a virtual device from the virtual machine.

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

DIAL

CP Class Any

Attaches a terminal device to a multiple access system.

```
Dial            userid   [vaddr ]
```

DIRECT

CMS

Sets up VM/SP directory entries.

```
DIRECT    [fn     [ft        [fm]]]  [(EDIT )]
          [USER   [DIRECT    [* ]]]
```

Control Statements:

Account number [*distribution*]

Defines an account number and a distribution identification.

ACIgroup *groupname*

Defines a user as a member of an access control group. If used, must precede first device statement (CONSOLE, MDISK, DEDICATE, LINK or SPOOL).

CLass *classes*

Defines up to 32 user classes assigned to a user. If used, must immediately follow USER control statement and USER control statement must have an asterisk (*) in its class field.

Console *vaddr* *devtype* [*class*] [*userid*]

Specifies the virtual console.

```
Dedicate  ⎧NETwork  vaddr  resource                                      ⎫
          ⎨         vaddr  ⎧rdev  [VOLID] [volser]  [3330V]  [R/O]⎫      ⎬
          ⎩                ⎩      [VOLID]  volser   [3330V]  [R/O]⎭      ⎭
```

Specifies that a real device is to be dedicated to this user.

```
DIRectory  cuu  devtype  volser  [alt-cuu]
```

Defines the device on which the directory is allocated. Must be the first statement.

```
INclude  profilename
```

Specifies the name of a PROFILE entry to be used as part of this USER entry. If used, must directly follow USER control statement.

```
Ipl  iplsys  [PARM  data]
```

Contains the name of the system to be loaded for the user when they log on. If used, must follow USER control statement, and precede first device statement (CONSOLE, MDISK, DEDICATE, LINK or SPOOL).

```
IUCV  ⎡userid                                       ⎤ PRIORITY  [MSGLIMIT  limit]
      ⎢resource id                                  ⎥
      ⎢* CRM                                        ⎥
      ⎢* IDENT  ⎡RESANY⎤ ⎡LOCAL ⎤  [REVOKE]         ⎥
      ⎢         ⎣resid ⎦ ⎣GLOBAL⎦                   ⎥
      ⎢*CCS                                         ⎥
      ⎢*SIGNAL                                      ⎥
      ⎢*LOGREC                                      ⎥
      ⎢*SPL                                         ⎥
      ⎢ALLOW                                        ⎥
      ⎣ANY                                          ⎦
```

Defines an authorization for establishment of a communication path with another virtual machine or a CP system service. If used, must follow USER control statement or another OPTION control statement, and precede first device statement (CONSOLE, MDISK, DEDICATE, LINK or SPOOL).

```
Link  userid  vaddr1  [vaddr2  [mode]]
```

Makes a device belonging to another user ID available to this virtual machine at logon.

```
Mdisk  cuu  devtype  ⎧cylr  cyls  volid  [mode [pr [pw [pm]]]]⎫
                     ⎨T-DISK  cyls                             ⎬
                     ⎩blkr  blks                               ⎭
```

Assigns DASD area which becomes the user's minidisk.

```
Option    [Realtimer]  [Ecmode]  [CONceal]  [Isam]  [Virt=real]
          [Acct]       [Svcoff]  [BMX] [CPUID bbbbbb]
          [Affinity nn] [VMsave] [STFirst]  [370E]
          [Maxconn]    [MIH]     [DIAG98]   [COMSRV]
          [Lang langid] [VCUNOSHR]
```

Selects specific options. If used, must follow USER control statement or another OPTION control statement, and precede the first device statement (CONSOLE, MDISK, DEDICATE, LINK or SPOOL).

```
Profile  profilename
```

Specifies the start of a PROFILE entry in the source directory. If used, must follow DIRECTORY control statements and precede USER control statements.

```
SCReen  area  {color  [hilight]  hilight  [color  ]}  .....
              {       [NONe   ]           [DEFault]}
```

Defines the color and extended-highlight options for the user terminal. If used, must follow USER control statement, and precede first device statement (CONSOLE, MDISK, DEDICATE, LINK or SPOOL).

```
SPEcial  vaddr  devtype  [IBM | TELE]
```

Specifies the I/O units available to the user (may or may not be connected to real or virtual devices). If used, must follow USER control statement, and precede first device statement (CONSOLE, MDISK, DEDICATE, LINK or SPOOL).

```
Spool  cuu  devtype  [class [ww [ll  [2WCGM] [CFS] [DATCK  ] ]]
                                     [4WCGM] [BTS] [NODATCK]
```

Specifies the unit record device to be spooled.

```
User  userid  pass  [stor  [mstor  [cl  [pri  [le  [ld   [cd   [es  ]]]]]]]]
                                              [ON  [ON   [ON   [ON  ]]]]
                                              [OFF [OFF  [OFF  [OFF ]]]]
```

Defines a virtual machine and creates a VM/SP directory entry. If * is used, the USER control statement should be followed immediately by the CLASS control statement.

DIRLIST

CMS

Lists directories of a specified directory structure in a fullscreen environment.

```
DIRList     [dirid]     [(options...[)]]

            Options:  ┌ACCessed┐     [PROFile fn]
                      └ALL     ┘
                                    ┌DIRlist fn┐
                      [APPend]      └NODirlist ┘
```

Special Commands which can be used in the DIRLIST environment, followed by their descriptions, are:

```
AUThlist    [fn ft] dirid          [(options...[)]]

            Options:   [ REFresh ]
```

Displays the authority that the issuer has for the specified file or directory, and also shows a list of all the users authorized for that file or directory if the issuer is the owner.

```
DISCARD     ┌fn ft ┌fm   ┐┐
            │      └dirid┘│
            └dirid        ┘
```

Erases a file displayed in the list.

```
EXECUTE     ┌Cursor┐  [command]
            └lines ┘
```

Issues CP/CMS commands (or EXECs) that make use of files displayed by DIRLIST.

DISABLE

CMS

Disables a storage group or file space for write access (SHARE) or all access (EXCLUSIVE). When all access is prohibited (EXCLUSIVE), the storage group minidisks may also be detached. DISABLE may be done on behalf of another user by specifying the *userid*. (File pool server operator only.)

```
DISABLE   { GROUP group-num { SHARE     } { DETACH   } } [ FOR owner ]
          {                 { EXCLUSIVE } { NODETACH } }
          {                                            }
          { FILESPACE userid { SHARE     }             }
          {                  { EXCLUSIVE }             }
```

DISABLE

CP Class A and B

Prevents communication lines from accessing VM/SP.

```
DISAble     { raddr...           }
            { L addr...          }
            { SNA  [userid]      }
            { ALL                }
```

DISABLE

RSCS

Deactivates a switched telecommunications port (a dialed telephone line).

```
DISAble     { cuu }
            { ALL }
```

DISCONN

CP Class Any

Disconnects the terminal from the VM/SP system while the virtual machine continues operation.

```
DISConn          [HOld ]
```

DISCONN

RSCS

Places RSCS in disconnect mode and optionally directs RSCS operator console output to another virtual machine (for RSCS operator only).

```
DISConn          [{ LOG   } [userid ]]
                 [{ NOLog }          ]
```

DISK

CMS

Performs disk-to-card and card-to-disk operations for CMS files. Can be used with files residing on minidisk or in directories.

```
DISK     {DUMP    fn  ft [fm]          }
         {LOAD    [( options... [)]]   }

         Options: [Fullprompt]  [Replace  ] [OLDDate]
                  [Minprompt ]  [NOReplace]
                  [NOPrompt  ]
```

DISKMAP

CMS

Summarizes the MDISK statements in the CP directory in order to show gaps and overlaps in minidisk assignments.

```
DISKMAP   fn   [ft    ]
               [DIRECT]
```

DISPLAY

CP Class G

Displays storage locations (second-level only), registers, program status word, channel address word, and channel status word.

```
Display  {  [  hexloc 1]   {-}  [hexloc 2]                  }
         {  [K hexloc 1]   {:}  [END     ]                  }
         {  [L hexloc 1]                                    }
         {  [T hexloc 1]                                    }
         {                                                  }
         {                 {.}  [bytecount]                 }
         {                      [END      ]                 }
         {                                                  }
         {  G reg 1     [ {-} [reg 2]          ]            }
         {  Y reg 1     [ {:} [END  ]          ]            }
         {  X reg 1     [                      ]            }
         {              [ {.} [regcount]       ]            }
         {              [     [END     ]       ]            }
         {                                                  }
         {  Psw                                             }
         {  CAW                                             }
         {  CSW                                             }
```

DLBL

CMS

In CMS/DOS, defines DOS and CMS sequential disk files for program I/O; identifies DOS files and libraries; defines and identifies VSAM catalogs, clusters, and data spaces; identifies VSAM, DOS, or CMS files uses for VSAM program I/O and access method service functions.

In CMS, defines a VSE file name or VSAM *ddname* and relates that name to a CMS file.

```
DLBL   [ ddname {fm   } [CMS fn   ft    ] [(option A  option B[)]] ]
       [        {DUMMY} [CMS FILE ddname]                          ]
       [                                                           ]
       [ ddname {fm   } [DSN qual1 [.qual2...qualn] ]              ]
       [        {DUMMY} [DSN qual1 [qual2...qualn ] ]              ]
       [                [DSN  ?                     ]              ]
       [                                                           ]
       [                      [(optionA  optionB  optionC[)]]      ]
       [                                                           ]
       [ {ddname} CLEAR                                            ]
       [ {  *   }                                                  ]

       Option A:   [SYS xxx]

       Option B:   [PERM]   [CHANGE  ]
                            [NOCHANGE]

       Option C:   [VSAM]   [EXTENT]   [CAT catdd]   [BUFSP nnnnn]
                            [MULT  ]
```

DLBL

GCS

Defines VSAM files used for program I/O.

```
DLBL  ┌ddname  mode  ┌DSN qual1 [[.] qual2....qualn]┐[optionB  optionC[)]]┐
      │              └DSN ?                         ┘                     │
      │                                                                   │
      │ddname  CLEAR                                                      │
      └   *                                                               ┘

                              optionB:        optionC:
                              [PERM]          [VSAM]
                                               ‾‾‾‾
                             ┌ CHANGE   ┐     [MULT]
                             │ ‾‾‾‾‾‾   │
                             └ NOCHANGE ┘     [CAT catdd]

                                              [BUFSP nnnnnn]
```

DMCP

CP Class C or E

Dumps any area of System/370 real storage to a spool device.

```
DMCP   ┌┌ML hexloc 1┐┌                        ┐┐
       ││NL hexloc 1││ ┌-┐  ┌hexloc 2┐        ││
       ││MT hexloc 1││ └:┘  └END     ┘        ││
       ││NT hexloc 1││                        ││  [* dumpid ]
       <│ M hexloc 1││                        │>
       ││ N hexloc 1││                        ││
       ││ L hexloc 1││ {.}  ┌bytecount┐       ││
       ││ T hexloc 1││      └END      ┘       ││
       ││   hexloc 1││                        ││
       └└     0     ┘└                        ┘┘
```

DOSGEN

CMS

Builds the CMSDOS physical saved segment.

```
DOSGEN    vstor    segname
```

DOSLIB

CMS

Deletes, compacts, or lists information about the phases of a CMS/DOS phase library.

```
DOSLIB  ⎧ DEL  libname  phasename1   [...phasenamen] ⎫
        ⎨ COMP  libname                              ⎬
        ⎩ MAP  libname  [(options... [)]]            ⎭

        Options:   ⎡ TERM  ⎤
                   ⎢ DISK  ⎥
                   ⎣ PRINT ⎦
```

DOSLKED

CMS

Link-edits CMS TEXT decks or object modules from a VSE relocatable library and places them in executable form in a CMS/DOS phase library.

```
DOSLKED    fn  ⎡ libname ⎤   [(options... [)]]
               ⎣ fn      ⎦

           Options:   ⎡ DISK  ⎤
                      ⎢ PRINT ⎥
                      ⎣ TERM  ⎦
```

DRAIN

CP Class D

Stops spooling activity on the specific device after the current file is finished spooling.

```
DRain      [ Reader    ]
           [ Printer   ]
           [ PUnch     ]
           [ raddr ... ]
           [ lprt      ]
           [ ALL       ]
```

DRAIN

RSCS

Deactivates an active communication link after the current file has finished being transmitted.

```
DRain      [ linkid ]
```

DROP WINDOW

CMS

Moves a window down in the order of displayed windows.

```
DROP WINdow   { wname }  [ n ]
              {   =   }  [ * ]
              {  WM   }
```

DROPBUF

CMS

Eliminates a program stack buffer.

```
DROPBUF        n
```

DSERV

CMS

Displays information contained in the VSE core image, relocatable, source, procedure, and transient directories.

```
DSERV    { CD  [PHASE {name [nn]}] } [d2...dn ] [(options...[)]]
         {                  [12]   }
         { RD                      }
         { SD                      }
         { PD                      }
         { TD                      }
         { ALL                     }

         Options:    [DISK ]     [SORT]
                     [TERM ]
                     [PRINT]
```

DUMP

CP Class G

Dumps virtual machine registers, program status word, and storage to the virtual printer.

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

ECHO

CP Class G

Returns data directly to the terminal a specified number of times.

```
ECho          [nn]
              [1 ]
```

EDIT

CMS

Calls the VM/SP System Product Editor in CMS editor (EDIT) compatibility mode to create or modify a file residing on a minidisk or in an SFS directory.

```
Edit      fn  ft  [fm]     [(options...[)]]
                   *

          Options:   [LRECL nn]     [NODISP]
```

ENABLE

CMS

Reinstates use of a storage group or file space. (File pool server operator only.)

```
ENABLE          {GROUP      group-num}  [FOR  owner]
                {FILESPACE  userid   }
```

ENABLE

CP Class A and B

Activates communication lines.

```
ENable        { raddr...           }
              { L addr...          }
              { SNA  [userid ]     }
              { ALL                }
```

ENABLE

RSCS

Activates a switched telecommunications port.

```
                      [        { ALL }]
ENable    { cuu }     [ TRace  { LOG }]    [TO  userid  [nodeid ]]
          { All }     [        { OFF }]
```

ENROLL ADMINISTRATOR

CMS

Adds a file system administrator to the specified file system file pool. (This authority is temporary, and may only be used by a file pool administrator.)

```
ENRoll ADMinistrator   {userid  }  [filepoolid:]  [(options... [ )]]
                       {nickname}

              Options:   [NOType              ]
                         [TYPe                ]
                         [                    ]
                         [STACK  [FIFO]       ]
                         [       [LIFO]       ]
                         [                    ]
                         [LIFO                ]
                         [FIFO                ]
```

ENROLL PUBLIC

CMS

Gives connect authority for an SFS file pool to all users.
(For use by file pool administrator only.)

```
ENRoll PUBlic          [filepoolid:]
```

ENROLL USER

CMS

Enrolls a user in the specified SFS file pool. (For use by file pool administrator only.)

```
ENRoll USEr      { userid   } [filepoolid:]  [(options... [ )]]
                 { nickname }

                 Options:   [BLOcks nnnnnnnnnn]
                            [STOrgroup nnnnn]
                            [ NOType               ]
                            [ TYPe                 ]
                            [ STACK [FIFO]         ]
                            [       [LIFO]         ]
                            [ LIFO                 ]
                            [ FIFO                 ]
```

ERASE

CMS

Deletes CMS files from a user's minidisk or SFS directory.

```
ERASE    ⎧ ⎧fn⎫⎧ft⎫ ⎡fm   ⎤ [(optionA...  [)]]           ⎫
         ⎪ ⎩* ⎭⎩* ⎭ ⎢dirid⎥                              ⎪
         ⎨          ⎣*    ⎦                              ⎬
         ⎪                                               ⎪
         ⎩ dirid             [(optionA optionB...  [)]]  ⎭

         OptionA:   ⎡Type                ⎤
                    ⎢Notype              ⎥
                    ⎢STACK  ⎡FIFO⎤       ⎥
                    ⎢       ⎣LIFO⎦       ⎥
                    ⎢FIFO                ⎥
                    ⎣LIFO                ⎦

         OptionB:   ⎡FILes  ⎤
                    ⎣NOFiles⎦
```

ESERV

CMS

Displays, punches or prints an edited (compressed) macro from a VSE source statement library (E sublibrary).

```
ESERV    fn
```

ESTATE, ESTATEW

CMS

ESTATE verifies the existence of a CMS file on a minidisk or in an SFS file pool. ESTATEW verifies the existence of a file on a read/write file mode.

$$\begin{array}{l}\textbf{STATE}\\ \textbf{STATEW}\\ \textbf{ESTATE}\\ \textbf{ESTATEW}\end{array}\qquad \left\{\begin{array}{c} fn\\ *\end{array}\right\} \left\{\begin{array}{c} ft\\ *\end{array}\right\} \left[\begin{array}{c} fm\\ *\\ -\end{array}\right]$$

ETRACE

CMS

Starts or stops file pool server external trace processing. (File pool server operator only.)

$$\textbf{ETRACE}\quad \left\{\begin{array}{c}\textbf{ON}\\ \textbf{OFF}\end{array}\right\}$$

ETRACE **GCS**

Enables or disables the recording of events in a spool file for a virtual machine or virtual machine group.

```
ETrace   ⎧⎡⎡DSP    ⎤⎡     ⎤⎤⎫
         ⎪⎢⎢EXT    ⎥⎢     ⎥⎥⎪
         ⎪⎢⎢FRE    ⎥⎢     ⎥⎥⎪
         ⎪⎢⎢GET    ⎥⎢     ⎥⎥⎪
         ⎪⎢⎢I/O    ⎥⎢     ⎥⎥⎪
         ⎪⎢⎢PRG    ⎥⎢     ⎥⎥⎪
         ⎪⎢⎢SIO    ⎥⎢ OFF ⎥⎥⎪ [GRoup]
         ⎨⎢⎢SSS    ⎥⎢     ⎥⎥⎬
         ⎪⎢⎢SVC    ⎥⎢     ⎥⎥⎪
         ⎪⎢⎢SYN    ⎥⎢     ⎥⎥⎪
         ⎪⎢⎣GTrace ⎦⎢     ⎥⎥⎪
         ⎪⎢ [ALL]   ⎢     ⎥⎥⎪
         ⎪⎣         ⎣     ⎦⎦⎪
         ⎪ [END]            ⎪
         ⎩                  ⎭
```

EXEC **CMS**

Executes special procedures made up of frequently used sequences of commands.

```
[EXec]     fn       [args...]
```

EXEC **RSCS**

Executes a sequence of commands contained in a CMS exec file that is accessible to the RSCS virtual machine.

```
EXec      filename   [arguments]
```

EXEC 2

CMS

Calls EXEC 2 files.

[EXec] *fn* [*args...*]

EXEC 2 Predefined Variables:

&

Initializes to its own name. This variable is automatically initialized or maintained.

&0

Initializes to the first word of the command string passed to the EXEC 2 interpreter. This variable is automatically initialized or maintained.

&1 &2 ...

Initializes to the arguments arg1 arg2 ... since they themselves are arguments and then are passed to the EXEC 2 file.

&ARGSTRING

Initializes to the argument string passed to the EXEC 2 file. This variable is treated as a single literal string.

EXEC 2

&BLANK

Assigns the value of a blank.

&CMDSTRING

Initializes to the untranslated command string passed to the EXEC 2 file.

&COMLINE

Initializes to zero and keeps the line number of the last EXEC 2 file issued command or subcommand.

&DATE

Evaluates true date (primary meridian -- GMT) in the form: YY/MM/DD. **See also** &TIME, below.)

&DEPTH

Keeps number of user-defined function and subroutine invocations to which return has not yet been made.

&FILEMODE

Initializes to third qualifier of EXEC 2 file.

&FILENAME

Initializes to first qualifier of EXEC 2 file.

&FILETYPE

Initializes to second qualifier EXEC 2 file.

&FROM

Initializes to zero and keeps line number of last executed &GOTO statement of EXEC 2 file.

&LINE, &LINENUM

Keeps current line number of EXEC 2 file.

&LINK

Keeps line number from which the currently executing user-defined function or subroutine was called, or is zero.

&N, &INDEX

Keeps the number of EXEC 2 arguments set. (See &1, &2, ... &n)

EXEC 2

&RC, &RETCODE

Initializes to zero, and keeps return code from last EXEC 2 issued command or subcommand.

&TIME

Evaluates true time-of-day (primary meridian -- GMT) in the form: HH:MM:SS. (**See also** &DATE, above.)

Note: An asterisk (*), a hyphen (-), or an ampersand (&) starting a command must be given as an argument.

EXEC 2 Control Statements:

&ARGS [*word1* [*word2* ...]]

Assigns word1, word2, ...wordn to arguments &1, &2, ... &n and discards previously set arguments.

```
&BEGPRINT   [ n      [ k ] ]
&BEGTYPE    [ label  [ * ] ]
            [ 1      [ - ] ]
            [ *            ]

line1
line2
. . .
```

Prints line1, line2, ... linen, truncated at column k if necessary. Does not remove surplus blanks or replace any EXEC 2 variables.

```
&BEGSTACK   ┌ n ┌ k  ┌FIFO┐┐┐
            │ * │ *  └LIFO┘┘│
            │   └ -         │
            │ label         │
            └ l             ┘
line1
line2
...
```

Places line1, line2, ... linen, in the program stack, truncated at column k if necessary. Does not remove surplus blanks or replace any EXEC 2 variables.

```
&BUFFER   n ┌comment┐
          * └       ┘
```

Discards lookaside buffer and its contents, then creates a new lookaside buffer for either designation.

```
&CALL   line-number ┌arg1 [arg2 ...]┐
        label       └               ┘
```

Calls the routine located at the specified label or line number and creates a new generation of the EXEC 2 arguments &1, &2, ..., &n initialized to arg1, arg2, ..., argn. Control is returned via the &RETURN statement.

```
&CASE   ┌U [comment]┐
        │M          │
        └           ┘
```

Translates any lowercase alphabetic character to uppercase or allows mixed cases. If U or M is not specified, the current setting is not changed.

EXEC 2

&COMMAND *word1* [*word2* ...]

Issues the command made up of word1, word2,..., each with one space between.

&DUMP *ARGS*
[*VAR* [*S*] [*var1 var2* ...]]

Prints a line for each &1, &2, ..., &n argument or variables var1, var2, ..., varn.

&ERROR *action*

Sets the action to be automatically taken on return from any command(s) or subcommand(s) that has a nonzero return code.

&EXIT [*return-code* [*comment*]
0]

Stops execution of the EXEC 2 file and yields the given numeric return code within the host system acceptable range.

&GOTO *line-number* [*comment*]
label

Transfers control to the designated line number or to the line with the label. The first character of label must be a hyphen.

```
&IF  word1  = | EQ              ┌ word2  executable statement ┐
           X = | NE             │                             │
           < = | LT             │                             │
           < = | X> | LE | NG   │                             │
           > | GT               │                             │
           > = | X < | GE | NL  └                             ┘
```

Executes the given executable statement if the condition is satisfied; otherwise, proceeds to next statement.

```
&LOOP  n      m
       label  *
              WHILE condition
              UNTIL condition
```

Loops through the designated operands until specified condition is satisfied.

Note: When condition is given, the operands are the same as given in the &IF statement.

```
&PRESUME  [ &COMMAND                          ]
          [ &SUBCOMMAND  environment          ]
```

(&COMMAND is underlined as the default.)

Presumes that any statement without a beginning ampersand is to be issued to CMS or to the designated subcommand environment.

```
&PRINT   [word1  [word2 ...]]
&TYPE
```

Prints or types a line containing the operand(s) each separated by one blank, or prints or types a blank line if no operand appears.

```
&READ     ┌ n                                ┐
          │ 1                                │
          │ *                                │
          │ ARGS                             │
          │ VAR[S] ┌var1  [var2 ...] ┐       │
          │        └ *    [* ...]    ┘       │
          └ STRING  var                      ┘
```

Reads from the console stack (if stack is not empty); otherwise, reads from the console the number of lines indicated, or assigns values as designated.

```
&RETURN       [word] [comment]
```

Returns control to the most recently called subroutine to which no return has as yet been made.

```
&SKIP     ┌ n  [comment] ┐
          └ 1            ┘
```

Skips the designated number of lines dependent on whether it is a positive or negative number. If it is equal to zero, control goes to the next line. If it is negative, control goes to the statement that precedes the &SKIP statement.

```
&STACK    [ [FIFO] [word1  [word2 ...]] ]
          [ [LIFO]                      ]
```

Places a line in the program stack that contains word(s) that are separated by one space or stacks a null line if no words are given.

```
&SUBCOMMAND        environment   [word1  [word2 ...]]
```

Issues the designated subcommand comprised of word1, word2, ..., separated by one space, to the appropriate environment.

```
&TRACE   ┌ ON    ┌output - action┐ ┐
         │ ERR   │               │ │
         │ ALL   │               │ │
         │ OFF   │               │ │
         │ *     └               ┘ │
         └ -                       ┘
```

Traces commands and subcommands as indicated by the trace setting. Information obtained is passed to the destination determined by output action.

Note: Initial trace setting is OFF. Default is asterisk (*), which means current setting remains in effect. Initially, output action is set to &PRINT.

```
&TRUNC       ┌ k  [comment] ┐
             └ *            ┘
```

Sets the truncation column to k or the maximum value (*). If no argument is shown, the previous setting stays in effect.

```
&UPPER       ARGS
             VAR [S]   [var1   [var2 ...]]
```

Translates any lowercase alphabetic characters to uppercase in the values of &1, &2, ... &n or the values of var1, var2, ..., varn.

EXEC 2 Predefined Functions:

&CONCATENATION OF [*word1* [*word2 ...*]]
&CONCAT OF

Concatenates the word(s) with no intervening space into a single word. If no word(s) appear, a null line results.

&DATATYPE OF [*word*]
&TYPE OF

Yields the value NUM if word represents a valid signed or unsigned number; otherwise, the value is CHAR.

&DIVISION OF *dividend divisor*
&DIV OF

Yields a numeric value representing the integral part of the division of the dividend by the divisor.

&LEFT OF *word j*

Left justifies word of length j. Truncates or pads with blanks on the right-hand side.

&LENGTH OF [*word*]

Gives either the number of characters in word or zero if word is not given.

&LITERAL OF [*string*]

Gives the literal string beginning with character after blank following OF and ending with the last nonblank character.

Note: Any leading or embedded blanks are retained and search for replacement variables is suppressed.

&LOCATION OF *needle* [*haystack*]

Searches haystack for first occurrence of needle and gives the starting position number, or gives a zero when there is no matching string, or needle exceeds length of haystack, or a word is not given.

&MULTIPLICATION OF *i* *j* [*k* ...]
&MULT OF

Yields numeric value that results from the multiplying of given numeric signed or unsigned words.

Note: There must be at least two of these.

&PIECE OF *word* *i* $\begin{bmatrix} j \\ * \\ - \end{bmatrix}$
&SUBSTR OF

Extracts part of word starting at character i for length j or to end of word.

Note: Value of i must be numeric positive and j must not be negative.

&POSITION OF *word* [*word1* [*word2* ...]]

Compares and tries to match word with word1, word2, ... If match occurs, gives numeric value of position of matching word. If no match is made or if there is no word(s) with which to compare, the result is zero.

&RANGE OF *stem i j*

Yields a string made up of words made by appending numbers to the stem ranging from i to j with one blank between each or, if i is greater than j, yields a null string.

Note: Appended numbers are stripped of any plus sign or redundant leading zeros.

&RIGHT OF *word j*

Right-justifies word of length j. Truncates or pads with blanks on left-hand side.

&STRING OF [*string*]

Gives the string beginning with character after blank following OF and ending with last nonblank character.

&TRANSLATION OF
&TRANS OF *word1* [*word2* [*word3*]]

Compares each character in word1 with word2. If a match is found, the position of that matching character in word1 is replaced with the character in the same position from word3.

```
&TRIM OF     [word]
```

Removes trailing blanks in a word. If word is not given, result is a null line.

```
&WORD OF       [word1  [word2 ...]]  i
```

Gives the ith word in the list of words unless the number given is zero, or exceeds the number of words in the list.

The format of the **EXEC 2 User-Defined Function**, followed by its description, is:

```
line-number OF   [arg1 [arg2 ...] ]
label OF
```

Calls the given function by transferring control to the given line number or label and creates a new generation of EXEC 2 arguments &1, &2, ... &n initialized to arg1, arg2, ... argn. Control is returned via the &RETURN statement.

EXECDROP

CMS

Purges storage-resident execs.

```
EXECDrop   {execname}  [exectype]  [(options...[)]]
EXDrop     {   *    }  [   *_   ]

           Options:  [User   ]
                     [SYstem ]
                     [SHared ]
```

EXECIO

CMS

Does I/O operations between a device and the program stack or a variable.

Note: Parsing of the EXECIO command differs from that of other CMS commands in that it involves handling of strings that may contain embedded blanks, parenthesis, other special characters, and words of more than eight characters. Therefore, if a right parentheses is used to mark the end of an EXECIO option, it must be preceded by at least one blank character. A right parenthesis cannot be used to mark the end of the STRING option.

```
EXECIO  {lines} (DISKR fn  ft [fm [linenum]]    [([FINIs]  options [a] [b]]           [)] )
        { *   } |CARD  [(                                  options [a] [b]]           [)] |
                |CP    [(                                  options [a] [b]  [d]] [e]] [)] |
                |DISKW fn  ft  fm [linenum                                                |
                {          [recfm [lrecl ]]]  [([FINIs]    options [b] [c]  [d]]      [)] }
                |PUNCH[(                                   options [b] [c]  [d]]      [)] |
                |PRINT [([CC{code  }]]                     options [b] [c]  [d]]      [)] |
                |      [ [  {DATA  }]]                                                    |
                (EMSG  [(                                  options [b] [c]  [d]]      [)] )
```

Option formats:

(a)

```
[FInd /chars /  ]    [Zone {n1  n2}]    [LIFO]      [SKip]
[LOcate /chars /]          {1   * }     [FIFO]
[Avoid / chars /]           -   -
```

(b)

```
[Margins {n1  n2}]   [STRIP]   [NOTYPE]   [STEm xxxxn]
         {1   * }                         [VAR  xxxx ]
          -   -
```

(c)

```
[CAse {U}]
      {M}
       -
```

(d)

```
[STring  xxx...]
```

(e)

```
[BUFfer  length]
```

EXECLOAD

CMS

Loads execs into storage.

```
EXECLoad   {fn ft }[fm [execname[exectype]]]   [(options...[)] ]
EXLoad

           Options:    [User    ]     [Push]
                       [SYstem  ]
```

EXECMAP

CMS

Lists storage-resident execs and displays execs in saved segments.

```
EXECMap   [execname [exectype ]]        [(options...[ )]]
EXMap     [   *     [   *     ]]

          Options:  [User ] [SYstem] [SHared]

                    [SEGment  {segname} ]
                    [         {   *   } ]
                    [NOSEGment          ]

                    [STACK [FIFO] ]
                    [      [LIFO] ]
                    [FIFO         ]
                    [LIFO         ]
```

EXECOS

CMS

Resets the OS and VSAM environments under CMS without returning to the interactive environment.

```
EXECOS      [cmd [operand1  [operand2..operandn  ]]]
```

EXECSTAT

CMS

Obtains the status of the specified exec. The status is returned in the form of a return code in register 15 as follows:

```
EXECStat    {execname }{exectype }
EXStat      { *       }{ *       }
```

EXECUPDT

CMS

Produces an updated version of a System Product Interpreter source program.

```
EXECUPDT   fn [ft     [fm]]  [(options...[)]]
              [EXEC   [* ]]

     Options:

     [CTL fn 1][HISTory  ][COMPress  ][COMMents  ]
               [NOHISTory][NOCOMPress][NOCOMMents]

     [ETMODE]   [SID  ]   [NOUPdate]
                [NOSID]
```

EXIT

RSCS

Enables or disables one or more specified user exits.

```
EXIT      { nnn }   [ ON  ]
          { ALL }   [ OFF ]
```

EXPAND

CMS

Adds space to a program in object deck form.

```
EXPAND    fn1 [ft1 [fm1 [fn2 [ft2 [fm2 ]]]]] [ (options: ... [ ) ] ]

          Options:
          [ INPUT  filename             ] [ PRINT   ]
          [ CSECT  csect    SIZE  size  ] [ NOPRINT ]
```

EXTERNAL

CP Class G

Simulates an external interruption condition on the virtual machine and returns control to that machine.

```
EXTernal     [ code ]
             [ 40   ]
```

F

CMS Border Command

Scrolls the window forward.

```
F
```

FETCH

CMS

Fetches a CMS/DOS or VSE executable phase.

```
FETch    phasename    [(options...[)]]

         Options:  [START] [COMP] [ORIGIN hexloc]
```

FILEDEF **CMS**

Defines an OS *ddname* and relates that *ddname* to any device supported by CMS or to a file residing in an SFS directory.

```
FIledef  [ {ddname} {Terminal   [(optionA  optionB  optionE [)]]                          } ]
         [ {nn    } {PRinter    [(optionA optionB  OPTCD J[)]]                           } ]
         [ {*     } {                                                                    } ]
         [          {PUnch      [(optionA optionB [)]]                                   } ]
         [          {Reader     [(optionA optionB [)]]                                   } ]
         [          {DISK   [fn    ft     [fm]][(optionA optionB  optionC [)]]           } ]
         [          {       [FILE  ddname [A1]]                                          } ]
         [          {                                                                    } ]
         [          {DISK [[fn    ft    ][fm]] {DSN ?              }                     } ]
         [          {     [[FILE  ddname][A1]] {DSN qual1 qual2 ...}                     } ]
         [          {                          {DSN qual1.qual2 ...}                     } ]
         [          {           [(optionA optionB optionC[)]]                            } ]
         [          {                                                                    } ]
         [          {DISK   vaddr                                                        } ]
         [          {DUMMY      [(optionA optionB [)]]                                   } ]
         [          {TAPn [LABOFF                                                 ]      } ]
         [          {     [BLP  [n]                                               ]      } ]
         [          {     [SL   [n][VOLID volid] [( DISP MOD  optionF[)]]         ]      } ]
         [          {     [SUL  [n][VOLID volid]                                  ]      } ]
         [          {     [NL   [n]                                               ]      } ]
         [          {     [NSL  filename                                          ]      } ]
         [          {     [(optionA  optionB  optionD [)]]                               } ]
         [          {GRAF   vdev   [(optionA [)]]                                        } ]
         [          {CLEAR                                                               } ]
```

FILEDEF

CMS

```
FIledef   OptionA:
             [PERM]  [CHANGE  ]
                     [NOCHANGE]

          OptionB:

             [RECFM a]  [LRECL nnnnn] [BLOCK nnnnn  ]
                                      [BLKSIZE nnnnn]

          OptionC:

             [KEYLEN nnn] [XTENT  nnnnn][LIMCT nnn] [OPTCD a][DISP MOD]
                          [        50  ]

             [MEMBER membername]  [CONCAT]  [DSORG  {PS}]
                                            [       {PO}]
                                            [       {DA}]

          OptionD:

             [7TRACK  ]   [TRTCH a] [DEN den ]   [ LEAVE]   [NOEOV]
             [9TRACK  ]
             [18TRACK ]

             [ALT  {TAPn}]
             [     {vdev}]

          OptionE:

             [UPCASE  ]
             [LOWCASE ]

          OptionF:

             [SYSPARM{(string)} ]
             [       {(?)     } ]
```

FILEDEF

GCS

Defines CMS format files and spool files.

```
FIledef  ┌           ┌PRinter [(optionA OPTCDj[) ]]                          ┐┐
         │           │PUnch [(optionA[)] ]                                   ││
         │           │Reader [(optionA[)]]                                   ││
         │{ddname}   │DISK [fn    ft      ] [fm] [(optionA optionB[)]]       ││
         │{*     }   │     [FILE  ddname  ] [A1]                             ││
         │           │DUMMY[(optionA[)]]                                     ││
         └           └CLEAR                                                  ┘┘

                                        option A:        option B:
                                        [PERM]           [DISP MOD]

                                        [CHANGE   ]      [DSORG PS]
                                        [NOCHANGE ]

                                        [RECFM a]

                                        [LRECL nnnnn]

                                        [BLOCK nnnnn  ]
                                        [BLKSIZE nnnnn]
```

Note: These operands work in the same manner as in the CMS FILEDEF command. However, only the operands and options shown are allowed. (For RECFM only F, FA, FB, FBA, U, UA, V, VA, and VBA are allowed.)

FILELIST

CMS

Lists information about CMS files in an SFS directory or a minidisk, with the ability to edit and issue commands from the list.

```
FILEList     [fn [ft [fm]]]        [(options...[)]]

        Options:  [Append]  [ Filelist   ]  [PROFile fn]
                            [ Nofilelist ]

                  [ALLfile ]  [ STAts  ]
                  [AUTHfile]  [ SHAre  ]
                              [ SEArch ]
```

Special Commands that can be used in the FILELIST environment, followed by their descriptions, are:

```
ALIalist      fn ft dirid          [(options...[)]]

        Options:  [ REFresh ]
```

Displays a list of users that have an alias to a specified file and lists the number of aliases each user has to the file.

```
AUTHlist     [fn ft] dirid         [(options...[)]]

        Options:  [ REFresh ]
```

Displays the authority that the issuer has for the specified file or directory, and also shows a list of all the users authorized for that file or directory if the issuer is the owner.

```
EXECUTE    [Cursor]  [command]
           [lines ]
```

Issues CP/CMS commands (or EXECs) that make use of files displayed by FILELIST.

```
DISCARD    [fn ft [fm   ]]
           [      [dirid]]
           [             ]
           [dirid        ]
```

Erases from disk a file displayed in the list.

FILEPOOL BACKUP

CMS

Backs up all data in a storage group and all associated file pool catalog data. The resultant file can be used as input to the FILEPOOL RESTORE command in case of data loss.

```
FILEPOOL BACKUP     group-number  [filepoolid:]  [(ACK | NOACK)]
```

FILEPOOL CLEANUP

CMS

Corrects any storage group or administration machine problems caused by a catastrophic failure of a FILEPOOL BACKUP or FILEPOOL RESTORE command.

```
FILEPOOL CLEANUP     group-number  [filepoolid:]
```

FILEPOOL FORMAT AUDIT

CMS

Formats the security audit data created by file pool server processing, and puts it in a file that may be printed or displayed.

FILEPOOL FORMAT AUDIT

FILEPOOL RESTORE

CMS

Loads the copy of a specified file pool storage group created by FILEPOOL BACKUP, thereby restoring the storage group (and the assoc information for all user IDs) to the status they had when that restore file was created.

FILEPOOL RESTORE *group-number* [*filepoolid:*] [**(ACK | NOACK)**]

FILESERV BACKUP

CMS

Starts a file pool server in dedicated maintenance mode to back up the *control data*. This includes the contents of the POOLDEF file, the control minidisk, and the catalog storage group.

FILESERV BACKUP

FILESERV DEFAUDIT

CMS

Adds, changes or deletes the assignment of the security audit output file for a file pool.

```
                        ┌TAPE  vadr                          ┐
FILESERV  DEFAUDIT     {DISK  filename filetype [filemode]   }
                        └DELETE                              ┘
```

FILESERV DEFBACKUP

CMS

Adds, deletes or changes the assignment of the control data backup file for the file pool.

```
                        ┌TAPE  vadr                          ┐
FILESERV  DEFBACKUP    {DISK  filename filetype [filemode]   }
                        └DELETE                              ┘
```

FILESERV GENERATE

CMS

Defines and initializes a new CMS Shared File System file pool.

```
FILESERV GENerate      [filename filetype [filemode]]
```

FILESERV LIST

CMS

Displays the contents of the file pool catalogs.

FILESERV LIST

FILESERV LOG

CMS

Formats and updates the file pool log minidisks. Allows you to reformat the logs, change their sizes and/or locations.

FILESERV LOG *vadr1 vadr2*

FILESERV MINIDISK

CMS

Adds one or more minidisks to one or more storage groups in a file pool.

FILESERV MINIDISK *filename filetype* [*filemode*]

FILESERV MOVEUSER

CMS

Moves all the file pool data for a user to a different storage group within the same file pool.

```
FILESERV MOVEUSER     userid groupnumber
```

FILESERV REGENERATE

CMS

Expands the file pool control minidisk without affecting the user storage group data.

```
FILESERV REGENERATE     vadr    [MAXDISKS   nnnnn]
                                [MAXUSERS   nnnnn]
```

FILESERV REORG

CMS

Deletes unused file pool catalog entries for each user and reorganizes the file pool catalogs to insure optimum use of catalog index space.

```
FILESERV REORG
```

FILESERV START

CMS

Calls file pool server processing to support access to a file pool from other virtual machines (referred to as *multiple user mode*).

```
FILESERV START
```

FINIS

CMS

Closes an open file on a minidisk or in a file pool.

```
FINIS        fn   ft   [fm]
             *    *    [*̲ ]
```

FLUSH

CP Class D

Halts and immediately purges or holds the current spool file.

```
Flush        {raddr}  [ALL]  [HOld]
             {lprt }
```

FLUSH

RSCS

Halts processing of a file currently being transmitted on a link. The file is either purged or held.

General User Format:

```
Flush      [*] spoolid    [ALL ]
                          [HOld]
```

Operator Format:

```
Flush      [linkid]  {spoolid}   [ALL ]
                     {*      }   [HOld]
```

FORCE

CMS

Rolls back any uncommitted file pool changes made by a user ID and severs the user ID's connection to the file pool server. (File pool server operator only.)

```
FORCE USER    userid [ALL]
```

FORCE

CP Class A

Forces logoff of the named user.

```
FORCE         userid
```

FORCE

RSCS

Immediately deactivates an active link, without quiescing file transfer. (For RSCS operator only).

```
FORCE      linkid
```

FORMAT

CMS

Prepares minidisks in CMS fixed block format.

```
FORMAT   vdev   fm     [nocyl ]   [(options...[)]]
                       [noblk]

         Options:

         [ Blksize   [ 512  ] ]  [Noerase] [Label] [Recomp ]
         [           [ 800  ] ]
         [           [ 1024 ] ]
         [           [ 2048 ] ]
         [           [ 4096 ] ]
         [           [   1K ] ]
         [           [   2K ] ]
         [           [   4K ] ]
```

FORMAT/ALLOCATE

Service Aid

Formats, allocates, and labels direct access volumes for paging, spooling, and CP file residence.

Format Service Aid Control Statements:

- Format Function

```
FORMAT    , devadr , devtype , volser , startadr , endadr , wrtver
```

- Allocate Function

```
ALLOCATE  , devadr , devtype , volser
TEMP , startadr , endadr
PERM , startadr , endadr
TDSK , startadr , endadr
DRCT , startadr , endadr
OVRD , startadr , endadr
PAGE , startadr , endadr
DUMP , startadr , endadr
END
```

- Label Function

```
FORMAT    , devadr , devtype , volser ,   LABEL
```

FREE

CP Class D

Releases previously held user spool files.

```
FRee                  [Printer]
           userid     [PUnch  ]
                      [ALL    ]
```

FREE

RSCS

Resumes transmission on a communication link previously in HOLD status.

```
FRee        [ linkid ]
```

FWDSPACE

RSCS

Causes the file currently being processed to be repositioned in a forward direction. This command is for RJE, 3270P, SNA3270P, and MRJE type links.

```
FWdspace     [ linkid ]  [nnn ]
```

GDUMP

GCS

Produces a copy of the contents of your virtual machine's storage.

```
GDUMP     [hexlocl1] [ {-} [hexloc2] ] [TO  *     ] [DSS] [FORMAT type ]
          [   0    ] [ {:} [END    ] ] [TO  userid]       [        GCS ]
                     [   bytecount   ]
```

GENDIRT

CMS

Fills in auxiliary module directories.

```
GENDIRT      directoryname  [targetmode [sourcemode]]
```

GENIMAGE

CMS

Presents input control file to the OS utility program IEBIMAGE. Creates text files used by the 3800 printer.

```
GENIMAGE     [ fn    ] [ ft   ][ fm ] [ sfn      ] [ sft     ] [ sfm ]
             [ SYSIN ] [ FILE ][ *  ] [ SYSPRINT ] [ LISTING ] [ A1  ]
```

GENMOD

CMS

Generates nonrelocatable CMS files (MODULE files).

```
Genmod    [fn  [MODULE  [fm]]]  [(options...[)]]
                        [A1]

     Options:  [FROM  entry 1][TO  entry 2 ]

               [MAP   ] [STR  ]  [OS ]
               [NOMAP ] [NOSTR]  [DOS]
                                 [ALL]
               [CLEAN  ] [SYSTEM]
               [NOCLEAN]

               [AMODE 24 ]   [RMODE 24 ]
               [AMODE 31 ]   [RMODE ANY]
               [AMODE ANY]

               [370]    [XA]
```

GENMSG

CMS

Converts a message repository file, made via XEDIT, into an internal form. Each record is read from the input file, its syntax is checked, and it is placed in an output file in a form the message processor can use.

```
GENMSG    fn ft fm applid [langid ]  [(options ...[)]]

     Options:

        [CP]  [Dbcs   ]  [List  ]  [Xref  ]
              [NODbcs ]  [NOList]  [NOXref]

        [Object  ]   [Margin nn]
        [NOObject]   [Margin 72]
```

GENSERVE

CMS

Builds CMS Shared File System (SFS) file pool server load modules.

```
GENSERVE    [module1 module2... [(ctlfile]]
```

GENTSAF

CMS

Builds the RUNTSAF module and creates a TSAF load map.

```
GENTSAF    [loadlist  [ctlfile ]]
           [ATSLOAD   [ATSSP  ]]
```

GEN3705

CMS

Generates and EXEC file that assembles and link-edits the 370x control program.

```
GEN3705        fn  ft  [fm]  [(options...)]
           Options:
              [RUN  ]  [SAVE  ]
              [NORUN]  [NOSAVE]
```

GET VSCREEN

CMS

Writes data from a CMS file to the specified virtual screen.

```
GET VSCreen     vname  fn  ft  [fm  [fromrec  [numrec  ]]]
                                *    1          *
```

GIVE

CP Class B

Transfers control of a dedicated tape drive to another virtual machine. (Tape drive must be dedicated to the virtual machine that gives the command.)

```
GIVE    vaddr1 [TO] userid [AS] vaddr2  [ RETurn   ]
                                        [ NORETurn ]
                                        [ R/O      ]
```

GLOBAL

CMS

Identifies specific CMS libraries to be searched for macros, copy files, missing subroutines, LOADLIB modules, or DOS executable phases.

```
GLobal   { MACLIB  }  [libname1...libname63]
         { TXTLIB  }
         { DOSLIB  }
         { LOADLIB }
         { CSLLIB  }
```

GLOBAL

GCS

Defines the CMS load libraries you want searched for modules.

```
GLobal    LOADLIB  [libname1... libname63]
```

GLOBALV

CMS

Sets, maintains, and retrieves a collection of named variables.

```
           ┌                                                                          ┐
GLOBALV    │ INIT                                                                     │
           │ SELECT [group   ]                                                        │
           │        [UNNAMED ]                                                        │
           │ ┌                    ┐ SET  ┌                                       ┐      │
           │ │                    │ SETS { name1 [value1] [name2  value2]...   }      │
           │ │                    │ SETP └                                       ┘      │
           │ │                    │                                                   │
           │ │                    │ SETL  ┌                 ┐                         │
           │ │  SELECT {group   } │ SETLS │                 │                         │
           │ │         {UNNAMED } │ SETSL { name   [value]  }                         │
           │ │                    │ SETLP │                 │                         │
           │ │                    │ SETPL └                 ┘                         │
          {  │                    │                                                   }
           │ │                    │ LIST [name1   [name2 ]... ]                       │
           │ │                    │                                                   │
           │ │                    │ STACK  {name1   [name2 ]... }                     │
           │ └                    ┘ STACKR {                    }                     │
           │ ┌ SELECT {group   }  ┐ PUT  ┌                      ┐                     │
           │ │        {UNNAMED }  │ PUTS { name1 [name2 ]...    }                     │
           │ │                    │ PUTP └                      ┘                     │
           │ │                    │                                                   │
           │ └                    ┘ GET [name1 [name2 ]...]                           │
           │ SELECT {group   }        PURGE                                           │
           │        {UNNAMED }                                                        │
           │ GRPLIST                                                                  │
           │ GRPSTACK                                                                 │
           │ PURGE                                                                    │
           └                                                                          ┘
```

Note: Although this command may be used in CMS EXECs, it is designed for use with EXEC 2 or REXX EXECs. For restrictions and precautions on its use, see the *CMS Command Reference*.

GRANT AUTHORITY

CMS

Authorized other users to read and/or modify one or more of your SFS directories or the files within those directories.

```
GRAnt AUThority  [fn  ft]  dirid  TO  {userid  }   [(options...[)]]
                 [*   * ]             {nickname}
                                      {PUBlic  }

                 Options:  [REAd ]  [TYPe          ]
                           [WRIte]  [NOType        ]
                                    [STACK [FIFO]  ]
                                    [      [LIFO]  ]
                                    [LIFO          ]
                                    [FIFO          ]
```

GRANT ADMIN

CMS

Gives a user file pool administration authority. It will remain in effect until it is explicitly removed or the file pool server processing ends. (File pool server operator only.)

```
GRANT ADMIN  userid
```

GROUP

CMS

Builds a GCS configuration file.

```
GROUP   systemname
```

GTRACE

RSCS

Provides additional tracing through VM/GCS.

GTRace $\left[\begin{array}{l}\underline{\textbf{ON}}\\ \textbf{OFF}\end{array}\right]$

H

CMS Border Command

Hides the window.

H

HALT

CP Class A

Stops any active channel program on the real device specified.

HALT *raddr*

HB

CMS Immediate Command

Halts the execution of CMS batch virtual machine at the end of the current job.

HB

HELP

CMS

Displays information about VM/SP, including:

- Commands: AVS, CMS, CP, IPCS, TSAF
- Subcommands: EDIT, XEDIT, SRPI, IPCS
- Macros: CMS assembler language
- Routines: CSL
- Messages: CMS, CP, TSAF, AVS, GCS, IPCS
- Control Statements and Instructions: EXEC, EXEC2, REXX

Note: The following program products available for installation also provide command and message HELP:

- RSCS, PVM and SQL/DS

```
Help          ┌                            ┐
              │ TASKs                      │
              │ Help                       │
              │ taskname   TASKs           │ [([optionA][optionB ][optionC ][ )]]]
              │ menuname  MENU             │
              │ component-name  cmd-name   │
              └                            ┘

              [ MESSAGE ] message-id
              [ MSG     ]

              OptionA:  [ BRIef   ]
                        [ DETail  ]
                        [ RELated ]

              OptionB:  [ALL] [DESCript] [FORMat] [PARMs]
                        [OPTions] [NOTEs] [ERRors]

              OptionC:  [ SCReen   ]  [ TYPe   ]    [EXTend]
                        [ NOScreen ]  [ NOType ]
```

HELPCONV

CMS

Converts a script file into an acceptable form to be used by the HELP facility.

```
HELPCONV    filename  filetype  [filemode]
                                [   *    ]
```

HI

CMS Immediate Command

Causes all currently executing System Product Interpreter or EXEC 2 programs or macros to terminate execution without destroying the environment (as HX would).

```
HI
```

HIDE WINDOW

CMS

Prevents the specified window from being displayed and connects the window to a virtual screen.

```
HIDE WINdow    [wname [ON  vname [ line  col]]]
               [  =                           ]
```

HO

CMS Immediate Command

Halts the current CMS tracing operation.

```
HO
```

HOLD

CP Class D

Defers processing of specified spool output.

```
HOLD          userid   [Printer]
                       [PUnch  ]
                       [ALL    ]
```

HOLD

RSCS

Suspends file transmission on an active link without deactivating the link.

```
HOld       [ linkid ]    [ IMMED ]
```

HT

CMS Immediate Command

Halts displaying at the terminal.

```
HT
```

HX

CMS Immediate Command

Halts execution of the current CMS command or program.

HX

HX

GCS

Halts execution of all programs and commands active in a virtual machine.

HX

IDENTIFY

CMS

Displays or stacks userid, nodeid, rscsid, date, time, time zone, and day of the week.

```
IDentify    [(options...[)]]

            Options:  ┌ STACK  ┌FIFO┐┐
                      │        └LIFO┘│
                      │              │
                      │ FIFO         │
                      │ LIFO         │
                      └ TYPE         ┘
```

IMAGELIB

CMS

Reads the control file created by GENIMAGE and loads files into the specified named system.

```
IMAGELIB    namesys
```

IMAGEMOD

CMS

Allows changes to the 3800 named systems.

```
IMAGEMOD   {GEN}  libname [modname [modname ...]]
           {ADD}
           {REP}  libname [modname [modname ...]]
           {DEL}
            MAP   libname [(options) ]

                 Options:    [TERM ]
                             [PRINT]
                             [DISK ]
```

IMMCMD

CMS

Establishes or cancels Immediate commands from within an exec.

```
IMMCMD   {SET   }
         {CLEAR }     name
         {QUERY }
         {STATUS}
```

INCLUDE

CMS

Brings additional TEXT files into virtual storage and establishes linkages.

```
INclude    fn...  [(options...[)]]
           Options:

           [CLEAR  ] [RESET{entry}] [ORIGIN {hexloc}]
           [NOCLEAR] [     {  *  }] [       {TRANS }]

           [MAP  ]   [TYPE  ]   [INV  ] [REP  ]
           [NOMAP]   [NOTYPE]   [NOINV] [NOREP]

           [AUTO  ]  [LIBE  ]   [START] [SAME]
           [NOAUTO]  [NOLIBE]

           [DUP  ]   [NORLDsave] [HIST  ]
           [NODUP]   [RLDsave  ] [NOHIST]
```

INDICATE

CP Class A

Displays the use of and contention for major system resources.

```
INDicate       FAVORed
```

INDICATE

CP Class E

Displays the use of and contention for major system resources.

```
INDicate    ┌ FAVORed                 ┐
            │ I/O                     │
            │ LOAD                    │
            │                         │
            │ PAGing      ┌ WAIT ┐    │
            │             └ ALL  ┘    │
            │                         │
            │ Queues                  │
            │                         │
            │ USER        ┌   *    ┐  │
            └             └ userid ┘  ┘
```

INDICATE

CP Class G

Displays the use of and contention for major system resources.

```
INDicate      ┌ LOAD ┐
              └ USER ┘
```

INIT

RSCS

Initiates RSCS operations. It must be the first RSCS command issued after the RSCS module is loaded into storage. No other RSCS commands will be accepted until INIT is completed.

```
INIT
```

IPCSDUMP

IPCS

Moves a dump file from the virtual reader to a CMS file, associates the map with the dump file, collects information to include in the problem report, renames unassigned CPTRAP files using the IPCS problem number for the dump being processed, and creates a symptom record.

```
IPCSDUMP
```

IPCSPRT

IPCS

Formats and prints dumps and CPTRAP files.

```
IPCSPRT  ┌HELP                                       ┐
         │[PRB] nnnnn [fm] [DUMP    [options...]]    │
         │            [A ] [CPTRAP              ]    │
         │TRP nnnnn [CPTRAP] [options...]            │
         └SPOOL  spoolid [CPTRAP] [options... ]      ┘

         options:

              [SUMMARY] [ENTRY  ] [PROMPT  ]
              [NOSUM  ] [NOENTRY] [NOPROMPT]

              [HEX   ]  [NOFORM]
              [FORMAT]  [NOREAL]
                        [NOVIRT]
                        [NOHEX ]
                        [NOMAP ]
```

IPCSPRT subcommands:

Note: IPCS subcommands can fall into the following functional categories:

- **COMMON** - All dumps and CPTRAP files
- **DUMP** - All dumps
- **CPTRAP** - CPTRAP files
- **CP** - CP dump
- **CMS** - CMS dump
- **GCS** - GCS dump
- **TSAF** - TSAF dump
- **AVS** - AVS dump
- **SFS** - SFS dump
- **PVM** - PVM dump
- **RSCSNET** - RSCS Version 1 dump
- **RSCSV2** - RSCS Version 2 dump
- **CICSVM** - CICS/VM dump.

The functional category of each subcommand is shown in brackets after the subcommand's description.

END

Ends the IPCSPRT session and returns to CMS. [CPTRAP]

FORMat

Enables trace entries to be printed in their long format, usually multiple lines per trace entry. [CPTRAP]

IPCSPRT

HELP	**IPCS**	*subcommand*
	IPCS	**MENU**
	DUMP	**MENU**
	CPTRAP	**MENU**
	MESSAGE	*message-id*
	MSG	*message-id*

Calls the CMS HELP facility to display information about IPCSPRT and its subcommands. [CPTRAP]

HEX

Enables trace entries to be printed in their short format, usually one line per trace entry. [CPTRAP]

HX

Ends the IPCSPRT session and returns to CMS. [CPTRAP]

PROCESS

Exits the IPCSPRT subcommand environment and begins creating the output. [CPTRAP]

QUIT

Ends the IPCSPRT session and returns to CMS. [CPTRAP]

```
SELect   ┌ALL      [ON ]                                                            ┐
         │         [OFF]                                                            │
         │                                                                          │
         │┌typenum TRAPId    trapid   ┌     ┌TRAPId    trapid ┐┐ ┐                 │
         ││        TRAPSet   trapset  │     │TRAPSet   trapset││ │                 │
         ││        Vmblok    address  │     │Vmblok    address││ │                 │
         ││        DEVaddr   nnnn     │ AND │DEVaddr   nnnn   ││...│...            │
         ││        COde      code     │     │COde      code   ││ │                 │
         ││        MACHtype  mtype    │     │MACHtype  mtype  ││ │                 │
         ││        USERid    userid   │     │USERid    userid ││ │                 │
         ││        ECODE     ecode    └     └ECODE     ecode  ┘┘ │                 │
         ││                                                      │                 │
         ││        OFF                                           │                 │
         │└                                                      ┘                 │
         │                                                                          │
         │DEVice  ┌ALL                   ┐ [ON   ]┐                                 │
         │        │{nnnn[-nnnn]...}      │ [ERROR]│                                 │
         └        └                      ┘ [OFF  ]┘                                 ┘
```

Specifies which trace entry types to print. [CPTRAP]

```
TIMESPAN    starttime   endtime
```

Designates a range of entries in the CPTRAP file using time stamp as an index value. [CPTRAP]

IPCSSCAN

IPCS

Allows interactive viewing of dumps and CPTRAP files.

```
IPCSSCAN   ┌HELP                                 ┐
           │[PRB]nnnnn  [fm]  [DUMP  ]           │
           │            [A ]  [CPTRAP]           │
           │TRPnnnnn [CPTRAP]                    │
           └SPOOL spoolid [CPTRAP]               ┘
```

IPCSSCAN subcommands:

Note: IPCS subcommands can fall into the following functional categories:

- **COMMON** - All dumps and CPTRAP files
- **DUMP** - All dumps
- **CPTRAP** - CPTRAP files
- **CP** - CP dump
- **CMS** - CMS dump
- **GCS** - GCS dump
- **TSAF** - TSAF dump
- **AVS** - AVS dump
- **SFS** - SFS dump
- **PVM** - PVM dump
- **RSCSNET** - RSCS Version 1 dump
- **RSCSV2** - RSCS Version 2 dump
- **CICSVM** - CICS/VM dump.

The functional category of each subcommand is shown in brackets after the subcommand's description.

(null line)

Reissues the previous CHAIN, LOCATE, or SCROLL subcommand. [COMMON]

?

Displays last subcommand entered. [COMMON]

+ *number*
−

Adjusts the address pointer and reissues the DISPLAY command. [DUMP]

&name [*subcommand*]
&

Creates a table of frequently used subcommands which may be called by another name, or to call a subcommand by its other name. [COMMON]

Aregs

Displays the registers, clocks, PSW, CSW, and CAW for the attached non-IPL processor. [CP]

ARIoblok *raddr*

Displays the RCHBLOK, RCUBLOK, and RDEVBLOK for the specified device attached to the non-IPL processor in an MP configuration. [CP]

BOTtom

Positions the user at the bottom of the CPTRAP file. [CPTRAP]

C

Displays the control registers for the failing processor. [CP]

CHain *fromhexloc increment endval*

Verifies the chain of homogeneous control blocks that start at the specified location. [DUMP]

CMS

Enters the CMS subset environment. [COMMON]

CMSPoint

Displays the formatted contents of pointers from CMS NUCON. [CMS]

CORtable *hexloc*

Displays page status and the formatted contents of the CORTABLE entry for the hexadecimal location specified. [CP]

Display $\left\{\begin{matrix} \textit{hexloc} \\ \textit{hexloc\%} \\ \textit{Thexloc} \\ \textit{Thexloc\%} \end{matrix}\right\}$ [*nnnn*]

Displays areas in the dump. The actual address or an indirect address may be specified. [DUMP]

DOSPoint

Displays the formatted contents of five pointers used by DOS simulation. [CMS]

DOWN $\left[\begin{matrix} \underline{1} \\ n \end{matrix}\right]$

Moves a specified number of entries toward the bottom of the CPTRAP file and displays the current entry. [CPTRAP]

DUMPID

Displays identification information concerning the data type being viewed and , when viewing dumps, the dump ID information associated with the dump. [COMMON]

END

Ends the IPCSSCAN session and returns to CMS. [COMMON]

```
FDISPlay   ⎧ ALL                    ⎫
           ⎪ COLLect                ⎪
           ⎪ LINKCtl   ⎧ BSC  ⎫     ⎪
           ⎪           ⎨ CTCa ⎬     ⎪
           ⎪           ⎪ ELAN ⎪     ⎪
           ⎨           ⎩ TLAN ⎭     ⎬
           ⎪ LINKDef                ⎪
           ⎪ NEIGhbor               ⎪
           ⎪ PATH                   ⎪
           ⎪ RESOurce               ⎪
           ⎪ ROUTing                ⎪
           ⎩ SERVice                ⎭
```

Displays data control blocks, tables, and arrays important to the TSAF virtual machine. [TSAF]

FORMat

Displays trace entries in their long format, usually multiple lines per trace entry. [CPTRAP]

G

Displays the set of general purpose registers (GPRs) in the failing processor or virtual machine. [DUMP]

```
GDISPLAY  { SGB                                        }
          {       { TASK    }                          }
          { GCB   { TRANS   }                          }
          {       { STATUS  }                          }
          {       { STORAGE }                          }
          {       { ALL     }                          }
          {                                            }
          { CVB  caddr                                 }
          { RLU  caddr                                 }
          { GWB  caddr                                 }
          { SCB  caddr                                 }
          { MAPN mname                                 }
          { MAPA maddr                                 }
          { GWBPTRS                                    }
          {          { *                      }        }
          {          { gateway-name  [ CVB ]  }        }
          {          {               [ RLU ]  }        }
```

Displays control blocks important to the AVS virtual machine and displays the MAPN and MAPA information for APPC/VM VTAM Support. [AVS]

```
HELP      [ IPCS      subcommand  ]
          [ IPCS      MENU        ]
          [ DUMP      MENU        ]
          [ CPTRAP    MENU        ]
          [ MESSAGE   message-id  ]
          [ MSG       message-id  ]
```

Calls the CMS HELP facility to display information about IPCSSCAN and its subcommands. [COMMON]

HEX

Displays trace entries in their short format, usually one line per trace entry. [CPTRAP]

HX

Ends the IPCSSCAN session and returns to CMS. [COMMON]

IDENTIFY

Displays identification information about the dump or CPTRAP file being viewed and, when viewing dumps, the dump ID information associated with the dump. [COMMON]

IPCSMAP

Adds an IPCS map to the dump being viewed. [DUMP]

IUcv

Displays all entries in the IUCV path table. [GCS, AVS, RSCSV2]

Locate
Locate Up { *string* | *X'string* } *start* *end* [*increment* | *1*]

Searches the dump for a particular string of data. [DUMP]

Locate
Locate Up { *string* | *X'string* } *start* [*end* [*increment* | *1*]] [*options*]

options:

[**DATA** | **HEADer** | **BOTH**] [**ALL** | **SELECT**]

Searches the CPTRAP file for a particular string of data and displays the entry in which a match was found. [CPTRAP]

LUName *luname* $\begin{bmatrix}\textit{userid}\\ \textbf{VTAM}\end{bmatrix}$

Displays the RDEVBLOK and SNARBLOK information for a VM/VTAM logical unit. [CP]

MAPA **hexloc**

Locates the module that contains the address specified. [DUMP]

MAPN *entrypointname*

Searches the load map for an entry point. [DUMP]

Mregs

Displays the registers, clocks, PSW, CSW, CAW and timers for the main (IPL) processor. [CP]

MRIoblok *raddr*

Displays the RCHBLOK, RCUBLOK, and RDEVBLOK for the specified device attached to the IPL processor in an MP configuration. [CP]

OSPoint

Displays the formatted contents of three pointers used in OS simulation. [CMS]

Print
PRT

```
[ subcommand ]
[ ON         ]
[ OFF        ]
[ CLOSE      ]
[ ?          ]
```

Prints the displayed data. [COMMON]

QUIT

Ends the IPCSSCAN session and returns to CMS. [COMMON]

Regs

Displays the registers, clocks, PSWs, timers, CSW, and CAW. [DUMP]

RIOblok *raddr*

Displays the RCHBLOK, RCUBLOK, and RDEVBLOK for the specified *raddr*. [CP]

Scroll U
ScrollU

```
[ HEX    ]
[ FORMat ]
```

Repeats the most recent DISPLAY or TRACE subcommand with an adjusted address. SCROLLU displays the preceding screen of data. SCROLL displays the next full screen of data. [COMMON]

```
SELect   { ALL   [ ON  ]
         {       [ OFF ]
         {
         { [ typenum TRAPId   trapid  [     { TRAPId   trapid  } ]     ]
         { [         TRAPSet  trapset [     { TRAPSet  trapset } ]     ]
         { [         Vmblok   address [     { Vmblok   address } ]     ]
         { [         DEVaddr  nnnn    [ AND { DEVaddr  nnnn    } ]...  ]...
         { [         COde     code    [     { COde     code    } ]     ]
         { [         MACHtype mtype   [     { MACHtype mtype   } ]     ]
         { [         USERid   userid  [     { USERid   userid  } ]     ]
         { [         ECODE    ecode   [     { ECODE    ecode   } ]     ]
         { [                                                           ]
         { [         OFF                                               ]
         {
         { DEVice { ALL                    } [ ON    ]
         {        { {nnnn[-nnnn]...}       } [ ERROR ]
         {                                   [ OFF   ]
```

Specifies which trace entry types to process. [CP, CPTRAP]

```
SYMPtom
```

Formats and displays the summary of the symptom record. [DUMP]

```
TACtive    [ taskid ]
           [ ALL    ]
```

Displays the task's active program list. [GCS, AVS, RSCSV2]

```
TIME       { POSition  timestamp }
           { DATE      datespec  }
```

Moves to a particular area in a CPTRAP file using the time stamp as an index value, or changes the default date for subsequent time stamp specifications. [CPTRAP]

```
TLoadl      [ taskid ]
            [ ALL    ]
```

Displays the task load list. [GCS, AVS, RSCSV2]

```
TOP
```

Moves to the top of a CPTRAP file. [CPTRAP]

```
Trace     [ [[FOR] count] [FROM fromloc] ]      [ HEX    ]
          [                              ]      [ FORMat ]
          [ [ Scroll  [U] ]              ]
          [ [ ScrollU     ]              ]
```

Displays trace table entries in short or fully-formatted versions. [CP, TSAF, AVS, SFS]

```
Trace     [ [[FOR] count]   ]     [ HEX    ]
          [ [ Scroll [U] ]  ]     [ FORMat ]
          [ [ ScrollU    ]  ]
```

Displays CPTRAP file entries in short or fully-formatted versions. [CPTRAP]

```
TSab        [ taskid ]
            [ ALL    ]
```

Displays the subpool map and chain header of a task. [GCS, AVS, RSCSV2]

UP $\left[\frac{1}{n}\right]$

Moves a specified number of entries toward the top of the CPTRAP file and displays the new current entry. [CPTRAP]

USERMAP

Adds a user load map to the dump being viewed. [CMS]

VIOblok *cuu* $\left[\begin{array}{l}\textit{userid}\\ \underline{\textbf{OPERATOR}}\end{array}\right]$

Displays the VCHBLOK, VCUBLOK, and VDEVBLOK for the specified device address and user ID. [CP]

Vmblok [*userid*]

Displays information relating to VMBLOKs. [CP]

VMLoadl

Displays information about all programs currently loaded in this virtual machine. [GCS,AVS, RSCSV2]

IPL

CP Class G

Simulates an initial program load function for a virtual machine.

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

ITASK

CMS

Performs most of the installation procedure by invoking other execs and commands.

```
ITASK  ┌ ALLOCATE                                          ┐
       │                                                   │
       │ ASSEMBLE    ┌ALLCP  ┐                             │
       │             │DMKBOX │                             │
       │             │DMKFCB │                             │
       │             ┤DMKRIO ├                             │
       │             │DMKSNT │                             │
       │             │DMKSYS │                             │
       │             └DMSNGP ┘                             │
       │ BASEIDS                                           │
       │                                                   │
       │ BUILD       ┌ CMS                          ┐      │
       │             │                              │      │
       │             │┌CP  ┐                        │      │
       │             ││    │                        │      │
       │             ┤│CPAP│ [NOASSEM]              ├      │
       │             ││    │                        │      │
       │             │└CPMP┘                        │      │
       │             │                              │      │
       │             │ GCS    ┌systemname┐          │      │
       │             └        └GCS       ┘          ┘      │
       │ FILESERV    {VMSYS }                              │
       │             {VMSYSU}                              │
       ┤                                                   ├
       │ LOAD        ┌AVS                          ┐       │
       │             │CMS                          │       │
       │             │CMSFTSRC                     │       │
       │             │CMSSRC                       │       │
       │             │CP                           │       │
       │             │CPSRC                        │       │
       │             │GCS                          │       │
       │             │GCSSRC                       │       │
       │             │HELP                         │       │
       │             │HPO                          │       │
       │             │IPCS                         │       │
       │             │IPCSSRC                      │       │
       │             ┤LANG  ┌ALL     ┐             ├       │
       │             │      │ALLOBJ  │             │       │
       │             │      │AVS     │             │       │
       │             │      │CMS     │             │       │
       │             │      ┤CMSSRC  ├             │       │
       │             │      │CP      │             │       │
       │             │      │GCS     │             │       │
       │             │      │HELP    │             │       │
       │             │      └TSAF    ┘             │       │
       │             │TSAF                         │       │
       └             └TSAFSRC                      ┘       ┘
```

ITRACE

CMS

Starts file pool server internal trace processing of APPC/VM communication related activities. (File pool server operator only.)

```
ITRACE   {ON  [buffersize]}
         {OFF             }
```

ITRACE

GCS

Enables or disables recording of internal trace events within a virtual machine or virtual machine group.

```
ITrace   { [ [GTrace]        ] }
         { [ [SUP   ]        ] }
         { [                 ] }
         { [           [OFF] ] }      [GRoup]
         { [ [ALL]           ] }
         {                     }
         { [END]               }
```

L

CMS Border Command

Scrolls the window to the left.

```
L
```

LABELDEF

CMS

Specifies standard HDR1 and EOF1 tape label description information for CMS, CMS/DOS, and OS simulation.

```
LAbeldef   [ { *  }  CLEAR                                                      ]
           [ { fn }                                                             ]
           [         ( [FID { ?   }]  [VOLID { volid   }]  [VOLSEQ volseq]  )   ]
           [         (      { fid }          { ?       }                    )   ]
           [         (                       { SCRATCH }                    )   ]
           [         ( [FSEQ fseq]  [GENN genn]  [GENV genv]                )   ]
           [         ( [CRDTE yyddd]  [EXDTE yyddd]   [SEC { 0 }]           )   ]
           [         (                                     { 1 }            )   ]
           [         (                                     { 3 }            )   ]
           [         (  [(options...[)]]                                    )   ]

      Options:    [PERM]    [CHANGE  ]
                            [NOCHANGE]
```

LANGGEN

CMS

Combines all the text files created by LANGMERG for a language and saves them in a Saved Segment named NLS*xy*, where x is the *levelid* and y is the *langid*. LANGGEN also saves CP's message repository for CP to use.

```
LANGGEN       langid  [levelid]  [( CTL  filename [)]]
```

LANGMERG

CMS

Combines all the language-related files for an application into one text file. (The LANGGEN command can then load this single text file into a Saved Segment as a language segment.)

```
LANGMERG     langid   applid   [( CTL filename [)]]
```

LINK

CP Class G

Permits one user to access minidisks belonging to another user.

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

[1] If password suppression is in effect, the DASD password (access mode password) cannot be entered on the LINK command line. The password must be entered after the prompting message: ENTER PASSWORD.

LISTDIR

CMS

Lists directories in a specified directory structure.

```
LISTDIR        [dirid]      [(options...[)]]

               Options:  ┌ACCessed┐        ┌STACK ┌FIFO┐ ┐
                         └ALL     ┘        │      └LIFO┘ │
                                           │LIFO         │
                         ┌SUBdirectory ┐   │FIFO         │
                         └NOSubdirectory┘  └XEDIT        ┘
```

LISTDS

CMS

Lists information about data sets and space allocation on OS, DOS, and VSAM minidisks.

```
LISTDS     [?      ]    {fm}  [(options...[ )]]
           [dsname ]    {* }

        Options:  [EXTENT]  [FORMAT ]  [PDS ]
                  [FREE  ]
```

LISTFILE

CMS

Lists information about CMS files stored on a minidisk or in an SFS directory.

```
Listfile      [fn [ft [fm]]]          [ ( options... [ )] ]
              [*  [*  [* ]]]

        Options:  [ALLFile ]  [SHAre ]  [Header  ]
                  [AUThfile]  [SEArch]  [NOHeader]

                  [ Exec    [Trace] [ARGS] ]  [ FName  ]  [Blocks ]
                  [ Trace   [ARGS ]        ]  [ FType  ]
                  [ APpend  [ARGS ]        ]  [ FMode  ]  [ % x ]
                  [ STACK [FIFO |LIFO]     ]  [ FOrmat ]
                  [ FIFO                   ]  [ ALloc  ]
                  [ LIFO                   ]  [ Date   ]
                  [ XEDIT                  ]  [ Label  ]
```

LISTIO

CMS

Displays information concerning CMS/DOS system and programmer logical units.

```
LISTIO     ┌ SYS    ┐   [(options... [)]]
           │ PROG   │
           │ SYSxxx │
           │ A      │
           │ UA     │
           └ ALL    ┘

           Options:  ┌EXEC  ┐   [STAT]
                     └APPEND┘
```

LKED

CMS

Link-edit a CMS TEXT file or OS object module into a CMS LOADLIB.

```
LKED     fname     [(options...[)]]
         Options:
         [NCAL] [LET] [ALIGN2] [NE] [OL] [RENT]
         [REUS] [REFR] [OVLY] [XCAL]
         [NAME membername] [LIBE libraryname]
         ┌XREF┐  ┌TERM  ┐  ┌PRINT  ┐
         │MAP │  └NOTERM┘  │DISK   │
         └LIST┘            └NOPRINT┘
         ┌SIZE ┌value1    value2 ┐┐
         │     │value1           ││
         │     ┤value1,          ├│
         │     │,value2          ││
         └     └,                ┘┘
         ┌AMODE ┌24 ┐┐      ┌RMODE ┌24 ┐┐
         │      ┤31 ├│      └      └ANY┘┘
         └      └ANY┘┘
```

LOAD

CMS

Brings TEXT files into storage for execution.

```
LOAD     fn ...        [(options...[)]]

         Options:

            ┌CLEAR  ┐ ┌RESET ┌entry┐┐   ┌MAP  ┐
            └NOCLEAR┘ └      └ *   ┘┘   └NOMAP┘

            ┌TYPE  ┐ ┌INV  ┐ ┌REP  ┐ ┌AUTO  ┐
            └NOTYPE┘ └NOINV┘ └NOREP┘ └NOAUTO┘

            ┌LIBE  ┐  [START] ┌DUP  ┐ ┌NORLDsave┐
            └NOLIBE┘          └NODUP┘ └RLDsave  ┘
            ┌NOPRES┐ ┌HIST  ┐
            └PRES  ┘ └NOHIST┘

            ┌RMODE  ┌24 ┐   ┐       ┌AMODE ┌24 ┐┐
            │       └ANY┘   │       │     < 31  >│
            │ORIGIN ┌hexloc┐│       └      └ANY┘┘
            └       └TRANS ┘┘
```

LOADBUF

CP Class D

Loads UCS (Universal Character Set) buffer or FCB (forms control buffer) on real printer.

```
LOADBUF   ┌ raddr   UCS   name   [Fold ]   [Ver  ]  ┐
          └ raddr   FCB   name   [Index   [nn   ]]  ┘
```

LOADCMD

GCS

Defines a program to be executed as a command.

```
LOADCmd      name member
```

LOADLIB

CMS

Maintains CMS LOADLIB libraries.

```
LOADLIB      ┌LIST      fileid1          ┐
             │COMPRESS         fileid1   │ [fileid2   [fileid3]] [(options...[)]]
             └COPY      fileid1          ┘

       Options:    ┌TERM ┐    ┌REPLACE┐
                   │PRINT│    └MODIFY ┘
                   └DISK ┘

         SYSIN Control Statements  (COPY function only )  ┌SELECT ┐
                                                           └EXCLUDE┘
```

LOADMOD

CMS

Brings a single MODULE file into storage.

```
LOADMod      fn   [MODULE [fm]]   [(options...[) ]]
                          [* ]

        Options:   ┌NOPRES┐
                   └PRES  ┘
```

LOADVFCB

CP Class G

Specifies the forms control buffer image for a virtual spooled 3203, 3211, 3262, 4245, 4248, or 3289E printer.

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

LOCATE

CP Class C and E

Provides the addresses of CP control blocks related to a specified user, virtual device, or real device.

```
LOCate        {userid [vaddr]}
              {raddr         }
```

LOCK

CP Class A

Locks specified pages in processor storage.

```
LOCK          {userid }firstpage   lastpage   [MAP]
              {SYSTEM }
```

LOGOFF

CP Class Any

Terminates a terminal session.

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

LOGON

CP Class Any

Initiates all virtual machine operation.

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

M

CMS Border Command

Changes the location of the window.

M

MACLIB

CMS

Creates or modifies CMS macro libraries.

```
MAClib   ⎧ ⎧GEN⎫                                                          ⎫
         ⎪ ⎨ADD⎬   libname   fn1  [fn2...]                                ⎪
         ⎪ ⎩REP⎭                                                          ⎪
         ⎨ DEL     libname   membername1 [membername2... ]                ⎬
         ⎪ COMP    libname                                                ⎪
         ⎩ MAP     libname   [membername1 [ membername2... ]] [(options...[)]] ⎭

         Options:   ⎡DISK              ⎤
                    ⎢PRINT             ⎥
                    ⎢TERM              ⎥
                    ⎢STACK    ⎡FIFO⎤   ⎥
                    ⎢         ⎣LIFO⎦   ⎥
                    ⎢FIFO              ⎥
                    ⎢LIFO              ⎥
                    ⎣XEDIT             ⎦
```

MACLIST

CMS

Displays a list of information about all members in the specified maclib, with the ability to edit and issue commands from the list.

```
MACLIST
MList        libname      [(options [)]]

          Options: [Append] [Compact  ] [PROFile fn]
                            [NOCompact]
```

Special commands that can be used in the MACLIST environment, followed by their descriptions, are:

```
EXECUTE   [Cursor lines]    [command]
```

Issues CP/CMS commands (or EXECs) that make use of files displayed by MACLIST.

```
DISCARD   [ libname libtype libmode  (MEMBER membername ]
```

Removes a member from the named library.

MAKEBUF

CMS

Creates a new program stack buffer.

```
MAKEBUF
```

MAP

IPCS

Converts various types of load maps into the proper format for IPCS.

```
MAP          type    [Prompt]
```

MAXIMIZE WINDOW

CMS

Expands a window to the physical screen size.

```
MAXimize WINdow      [wname]
                     [  =  ]
```

MESSAGE

CP Class A and B

Sends text messages to other users, system operator, or self.

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

MESSAGE

CP Class Any

Sends text messages to other users, system operator or self.

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

MIGRATE

CP Class A

Activates usual page/swap table migration routines or forces the pages of the specified user to a secondary device when the user is currently active.

```
MIGrate        [userid]
```

MINIMIZE WINDOW

CMS

Reduces the size of the window to one line.

```
MINimize WINdow   [ wname ]
                  [ =     ]
```

MODIFY USER

CMS

Modifies a user's filespace allocation in the Shared File System. (For use by the file pool administrator only.)

```
MODIfy USEr          {+ nnnnnnnnnn}  FOR {userid  } [filepoolid:][(options... [)]]
                     {- nnnnnnnnnn}      {nickname}

                     Options:   [NOType             ]
                                [TYPe               ]
                                [                   ]
                                [STACK  [FIFO]      ]
                                [       [LIFO]      ]
                                [                   ]
                                [LIFO               ]
                                [FIFO               ]
```

MODMAP

CMS

Displays a MODULE file load map.

```
MODmap        fn
```

MONITOR

CP Class A and E

Starts or stops the recording of interruptions and other events that occur in the real machine.

```
MONitor  ┌ AUTOdisk   { ON  }                                                          ┐
         │            { OFF }                                                          │
         │ CLose                                                                       │
         │ Display    [ SPOOL ]                                                        │
         │            [ TAPE  ]                                                        │
         │            [ ALL   ]                                                        │
         │ ENable     { APPLdata }                                                     │
         │            { PERForm  }                                                     │
         │            { RESPonse }                                                     │
         │            { SCHedule }                                                     │
         │            { USER     }                                                     │
         │            { INSTsim  }                                                     │
         │            { DAStap   }                                                     │
         │            { SEEKs    }                                                     │
         │            { SYSprof  }                                                     │
         │ INTerval   nnnnn  [ SEC ]  mm                                               │
         │                   [ MIN ]                                                   │
         │ LIMit   n  [ NOSTOP ]                                                       │
        {             [ STOP   ]                                                        }
         │            [ SAMPLE ]                                                       │
         │ SEeks      { INclude   raddr raddr... }                                     │
         │            { EXclude   raddr raddr... }                                     │
         │            { DELete                   }                                     │
         │            { DISplay                  }                                     │
         │ STArt      [ SPOOL       [TO userid]  [BUFFS n]                          ]  │
         │            [ CPTRACE                                                     ]  │
         │            [                                                             ]  │
         │            [ TAPE        raddr      [ MODE { 800  } ]  [BUFFS n]         ]  │
         │            [                        [      { 1600 } ]                    ]  │
         │            [                        [      { 6250 } ]                    ]  │
         │            [                        [      { 38K  } ]                    ]  │
         │ STOP       [ SPOOL   ]                                                      │
         │            [ CPTRACE ]                                                      │
         │            [ TAPE    ]                                                      │
         │ TIME       { FROM    h1:m1  TO  h2:m2 }                                     │
         │            { FOR     hh:mm            }                                     │
         │            { ALL                      }                                     │
         └            { NONE                     }                                     ┘
```

MOREHELP

CMS

Displays additional or related information about the latest valid HELP command you issued.

```
MOREhelp        [([ optionA] [optionB] [) ]]

          OptionA:  [DETail  ]
                    [BRIef   ]
                    [RELated ]

          OptionB:  [ALL] [DESCript] [FORMat] [PARMs]

                    [OPTions] [NOTEs] [ERRors]
```

MOVEFILE

CMS

Moves data from one device to another device of the same or different type.

```
MOVEfile  [ inddname  [ outddname ] ]  [(PDS [)]]
          [ INMOVE    [ OUTMOVE   ] ]
```

MSG

RSCS

Sends a message line to a local or remote operator or user.

```
Msg        nodeid   { userid }    [msgtext]
                    { SYSTEM }
```

MSGNOH

CP Class A and B

Lets a virtual machine send messages without the standard header associated with the MESSAGE command.

```
MSGNOH   { userid   }    msgtext
         { ALL      }
         {   *      }
         { OPerator }
```

N

CMS Border Command

Minimizes the window.

```
N
```

NAMEFIND

CMS

Displays/stacks information from a NAMES file (default 'userid NAMES').

NAMEFind *:tag value* [*:tag* [*value*]...] [**(options...** [**)**]]

Options:

STACK [*n* |*|1] [**FIFO**|**LIFO**]
FIFO [*n* |*|1]
LIFO [*n* |*|1]
TYPE [*n* |*|1]

[**FILe** *fn*] [**LINenum**] [**STARt** *recnum*]

[**SIze** [*n* |*|8]] [**XEDIT**]

NAMES

CMS

Displays a menu to create, display, or modify entries in a 'userid NAMES' file. (The menu is available only on display terminals.)

NAMES [*nickname*]

NCPDUMP

CMS

Processes CP spool reader files created by 3705x dumping operations.

NCPDUMP [**DUMP** *xx*][**(**[**ERASE**][**NOFORM**][**NCPBUFF**] [**)**]]

NETDATA

CMS

Queries, receives or sends files to users at a network node or on your system. Normally called from an exec.

```
NETDATA  { QUERY                    [(optionA...[)]]                           }
         { RECEIVE fn ft fm         [(optionA optionB optionD...[)]]           }
         { SEND fn ft fm TO userid AT node [(optionA optionC optionD..[)]]     }


OptionA:  [ TYPe                ]   [ MSGSubs ]
          [ NOType              ]   [ MSGAll  ]
          [ STACK [ FIFO ]      ]
          [       [ LIFO ]      ]
          [ LIFO                ]
          [ FIFO                ]

OptionB:  [ Fullprompt ]   [ NEwdate ]   [ Replace   ]
          [ Minprompt  ]   [ Olddate ]   [ NOReplace ]
          [ NOPrompt   ]

          [ NOTE ]   [ NOTEBook fn ]   [ PURGE ]

OptionC:  [ Ack   ]   [ NOTE ]
          [ NOAck ]

OptionD:  [ Log   ]   [ NOSpool ]   [ Xedit ]
          [ NOLog ]
```

NETWORK

CP Class A

Controls communications to 370x controllers or resources or 3270 remote equipment.

```
NETWORK   ATTach   resid  [To]  userid  [As]  cuu

          DETach   resid  [From]  userid

          DISAble  [ ALL                 ]
                   [ resid  [resid...]   ]

          { DISPlay raddr hexloc1   [ { - } [ hexloc2 ]  ] }
          {                         [ { : } [ END     ]  ] }
          {                         [                    ] }
          {                         [ { . } [ bytecount] ] }
          {                         [       [ END      ] ] }

          DUMP raddr  [ IMMED ]
                      [ OFF   ]
                      [ AUTO  ]

          ENable  [ ALL                ]
                  [ resid  [resid... ] ]

          LOAD  raddr  ncpname

          POLLdlay  nnnn  [ ALL   ]
                          [ raddr ]

          Query     [ ACTive             ]
                    [ OFFline            ]
                    [ FREe               ]
                    [ ALL                ]
                    [ resid  [resid...]  ]

          SHUTDOWN  [ raddr ]
                    [ ALL   ]

          VARY  { ONline  }  resid  [resid...]
                { OFFline }
```

NETWORK

NETWORK

CP Class B

Controls the 370x control program and its resources. Also provides a means of altering binary synchronous line poll delay interval.

```
NETWORK     ATTach   resid  [To] userid [As]   vaddr

            DETach   resid  [From] userid

            DISAble  [ ALL                ]
                     [ resid  [resid...]  ]

      {  DISPlay raddr hexloc1      [ {-} [hexloc2  ] ]  }
      {                             [ {:} [END      ] ]  }
      {                             [                 ]  }
      {                             [ {.} [bytecount] ]  }
      {                             [     [END      ] ]  }

            DUMP  raddr  [ IMMED ]
                         [ OFF   ]
                         [ AUTO  ]

            ENable   [ ALL                ]
                     [ resid  [resid... ] ]

            LOAD  raddr  ncpname

            POLLdlay  nnnn  [ ALL   ]
                            [ raddr ]

            Query     [ ACTive              ]
                      [ OFFline             ]
                      [ FREe                ]
                      [ ALL                 ]
                      [ resid  [resid...]   ]

            VARY   { ONline  }  resid  [resid...]
                   { OFFline }
```

NETWORK

RSCS

Starts or ends communications with ACF/VTAM (for RSCS operator only).

```
             ┌         ┌ APPLid name     ┐  ┐
             │  STArt  │ Pass password   │  │
NETwork      │         │ RETry nn        │  │
             │         └ RPLs nnn        ┘  │
             └  HALT   [ QUICK ]            ┘
```

NOTE

CMS

Prepares a ‘note’ for one or more computer users, to be sent by way of the SENDFILE command.

```
NOTE        [name...  [CC:  name...]]     [(options... [)]]

           Options:

           ┌ ACk   ┐   [ADd]   [Cancel]  ┌ NOTebook fn ┐
           └ NOAck ┘                     │ NOTebook *  │
                                         └ NONotebook  ┘

           ┌ LOG   ┐   ┌ LONg  ┐   [Replace]  [PROFile fn]
           └ NOLog ┘   └ Short ┘
```

NOTREADY

CP Class G

Simulates loss of ready status on virtual device.

```
NOTReady         vaddr
```

NUCXDROP

CMS

Deletes specified nucleus extensions.

```
NUCXDROP     {name1 [name2...]}
             {*               }
```

NUCXLOAD

CMS

Loads a nucleus extension.

```
NUCXLOAD        {name   [fn]          }  [( [SYstem] [SErvice]                    ]
                {name   member  ddname}  [  [ENdcmd] [IMmcmd] [Push] [)]]         ]
```

NUCXMAP

CMS

Identifies existing nucleus extensions, including those residing in saved segments.

```
NUCXMAP     [name]   [(options...[)]]
            [ *  ]
            [ALL ]

      Options:  [NOSEGMent              ]  [ATTRibutes]
                [SEGMent {segname}      ]
                [        {   *   }      ]  [SEGInfo   ]

                [STACK [FIFO]]             [ALL]
                [      [LIFO]]
                [FIFO        ]
                [LIFO        ]
```

O

CMS Border Command

Restores the window.

O

OPTION

CMS

Changes the specified DOS/VS COBOL compiler (FCOBOL) options that are in effect for the current terminal session.

```
OPtion     [options...]

           Options:

           [DUMP  ][DECK  ][LIST  ][LISTX  ][SYM  ]
           [NODUMP][NODECK][NOLIST][NOLISTX][NOSYM]

           [XREF  ] [ERRS  ] [48C] [TERM  ]
           [NOXREF] [NOERRS] [60C] [NOTERM]
```

ORDER

ORDER

CP Class D

Places closed spool files in a specified order by device type. (A combination of CLASS and spoolid specifications may be entered.)

```
ORDer                                     ┌ CLass c1  CLass c2 ...          ┐ 1
        ┌ userid ┐  ┌ Reader|RDR  ┐        │ spoolid1  spoolid2 ...          │
        │ SYSTEM │  │ Printer|PRT │        │ FORM form1  FORM form2 ...      │
        └   *    ┘  └ PUnch|PCH   ┘        └ DEST dest1  DEST dest2 ...      ┘
            -
```

Sequencing may be done with the ORDER command using a combination of "CLASS c" FORM, and spoolid specifications.

ORDER

CP Class G

Places closed spool files in a specified order by device type.

```
ORDer     ┌ Reader  ┐ ┌ CLass  c1  CLass  c2 ...        ┐ 1
          │ Printer │ │ FORM form1  FORM  form2 ...     │
          │ PUnch   │ │ DEST dest1  DEST  dest2 ...     │
          └         ┘ └ spoolid1   spoolid2 ...         ┘
```

Sequencing may be done with the ORDER command using a combination of "CLASS c" FORM, and spoolid specifications.

ORDER

RSCS

Reorders files enqueued on a specific link.

```
ORDer     [linkid]   spoolid    [spoolid ...  ]
```

OSRUN **CMS**

Loads, relocates and executes a load module from a CMS LOADLIB or an OS module library.

```
OSRUN     member    [PARM =parameters ]
```

OSRUN **GCS**

Starts a GCS application program.

```
OSRUN     member   [PARM = parameters]
```

OVERRIDE **CMS**

Implements changes to the class structure.

```
OVERRIDE    fn    ft   fm    [( EDIT ]
                             [( FREE ]
```

DESTINATION control statement:

```
DESTination    cuu  devtype  volser  altcuu
```

Override file control statement:

```
command        [Type=  c]   Class=  {classes}
                                    {   *   }
```

P

CMS Border Command

Pops the window.

```
P
```

PARSECMD

CMS

Calls the parsing facility from within an exec.

```
PARSECMD    uniqueid      [ (options... [)] ]

         Options:  [TYPE    ]  [APPLID applid]   [STRING cmdstring]
                   [NOTYPE  ]
```

PEEK

CMS

Displays a file that is in your virtual reader without reading it onto disk or directory.

```
PEEK      [spoolid]   [(options...[)]]

          Options:
            [FRom recno]  [FOr numrec]  [PROFile fn]
```

A Special Command which can be used in the PEEK environment, followed by its description, is:

```
DISCARD   ┌ fn ft ┌ fm    ┐ ┐
          │       └ dirid ┘ │
          │                 │
          └ dirid           ┘
```

Erases a file displayed on the PEEK screen.

PER

CP Class A,B,C,D,E,F, and G

Monitors certain events as they occur during program execution in the user's virtual machine, such as: the fetching and execution of an instruction, the execution of a successful branch instruction, the execution of an instruction that alters a specific general-purpose register, and the execution of an instruction in the virtual machine that alters storage.

```
PER     EVENT TYPES:
       [Instruct [[DATA ] hex-data ]                              [         ]]
       [BRanch   [[INTO ] into-addr-range  ]                      [ options ]]
       [                                                          [         ]]
       [STore   [[[INTO ]  storage-addr-range ]          ]        [         ]]
       [        [[[INTO ]  addr  [DATA ] hex-data]       ]        [         ]]
       [                                                          [         ]]
       [Mask     [INTO ]  addr  [DATA ] mask-field                [         ]]
       [                                                          [         ]]
       [G [reg1]    [ {-} [reg2 ]     ][[DATA hexword]]           [         ]]
       [            [ {:}             ]                           [         ]]
       [            [ {.} [regcount]  ]                           [         ]]

        OPTIONS:
          Range  instruction-addr-range
          FRom   instruction-addr-range
          PAss  [0]
                [n]
          CMd  { text }
         [ Printer       [RUN ]             ]
         [[TErminal]    [NORun    ]         ]
         [[BOth    ]    [RUN      ]         ]
         [              [STEp [1] ]         ]
         [              [     [n] ]         ]
          GUESTR
          GUESTV
          DATOFF
          DAT

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

POP WINDOW

CMS

Moves a window up in the order of displayed windows.

```
POP WINdow      { wname }  [ n ]
                { WM    }  [ * ]
```

PORT

RSCS

Specifies that a port is to be connected to the public switched telephone network and that the port may be used for an auto-dial or auto-answer link. Reserves a virtual address for a BSC telecommunications line to be dynamically allocated to a link that is started without explicit port specification.

```
                  { DIAL   }
PORT       cuu    { NODial }
                  { OFF    }
```

POSITION WINDOW

CMS

Changes the location of a window on the physical screen.

```
POSition WINdow    { wname }  psline   pscol
                   {   =   }
```

PRB

IPCS

Updates the STATUS, FUNCTN, SEV or DUP/APAR/PTF fields in a symptom summary record or displays a specific problem report.

```
PRB   ⎧          ⎧APAR       aparnumber           ⎫⎫
      ⎪          ⎪CLOSE                           ⎪⎪
      ⎪          ⎪DSPLY                           ⎪⎪
      ⎪          ⎪DUPOF      ⎧nnnnn     ⎫         ⎪⎪
      ⎪          ⎪           ⎩aparnumber⎭         ⎪⎪
      ⎨nnnnn     ⎨IBM                             ⎬⎬
      ⎪          ⎪NEEDINFO                        ⎪⎪
      ⎪          ⎪PTFIS      [filename] filetype  ⎪⎪
      ⎪          ⎪PTFON                           ⎪⎪
      ⎪          ⎪SEV        [1 2 3 4]            ⎪⎪
      ⎪          ⎩USER                            ⎭⎪
      ⎩HELP                                        ⎭
```

PRELOAD

CMS

Collects multiple text files and reformats them into a single text file.

```
PRELOAD   loadlist   [ ctlfile ]
```

PRINT

CMS

Spoools a specified CMS file to the virtual printer.

```
PRint        fn ft [fm]    [ (options... [)]]
                   [* ]
```

Options:

```
[OVersize] [CC [HEADer] ] [UPCASE][TRC  ]
           [NOCC        ]         [NOTRC]

[LINECOUN {nnn}] [MEMBER {*         }][HEX]
[         {55 }] [       {membername}]
```

PROB

IPCS

Enters or appends a problem report in IPCS.

```
PROB
```

PROGMAP

CMS

Displays or places on the program stack information on programs currently loaded in storage or in a saved segment.

```
PROGMAP   [progname]    [(options...[)]]
          [   *    ]

          Options: [PROGRAM              ] [STACK [FIFO]]
                   [NUCX                 ] [      [LIFO]]
                   [ALL                  ] [FIFO        ]
                   [NOSEGment            ] [LIFO        ]
                   [SEGment {segname}    ]
                   [        {   *   }    ]
```

PROGRAMMABLE OPERATOR

CMS

Anyone, authorized by the active routing table, can execute the programmable operator commands. To execute a programmable operator command you must send a message to the programmable operator facility virtual machine. The text of the message is the command to be entered. Use the CMS EXEC, PROPST EXEC, to call the programmable operator facility.

The format of the invocation exec is:

```
PROPST      [rtable-name]   [DISConn ]
            [PROP       ]
```

The local format of the message sent to the programmable operator facility is:

```
Message      userid  propcmd  [parameters]
MSG
```

The distributed (network) format of the message sent to the programmable operator facility is:

```
SMsg        netid Msg nodeid  userid  propcmd  [parameters]
```

The CMS TELL EXEC may be used by the logical operator instead of either the local or the distributed format.

The format of the TELL EXEC is:

```
TELL          name    message
```

PROGRAMMABLE OPERATOR

Programmable operator commands:

```
CMD      vmcmd
```

Executes selected CP or CMS commands in the programmable operator's virtual machine.

```
FEEDBACK     text...
FB
```

Places comments about the operation of the system and the programmable operator in the feedback file.

```
GET       ⎧FEEDBACK     ⎫
          ⎨FB           ⎬
          ⎩LOG [yymmdd] ⎭
```

Retrieves one of the programmable operator files: the feedback file (FB) or the log file (LOG).

```
LOADTBL      [filename]     [( RPL [)]]
```

Loads a new routing table to control the operation of the programmable operator facility.

```
LGLOPR       ⎧ASN⎫
             ⎨RLS⎬
             ⎩RPL⎭
```

Changes the assignment of logical operator of the programmable operator facility.

```
LOG     text...
```

Writes a message to the current day's log file.

```
QUERY     ⎧ HOSTCHK              ⎫
          ⎪ LGLOPR               ⎪
          ⎨ PROPCHK  [nodeid]    ⎬
          ⎪ RTABLE               ⎪
          ⎩ LOGGING              ⎭
```

Indicates node-checking status, the user ID and node ID of the currently assigned logical operator, the logging status, the node-checking status, and the name of the programmable operator's active routing table, respectively.

```
SET       ⎧ DEBUG    {ON }                ⎫
          ⎪          {OFF}                ⎪
          ⎪ HOSTCHK  {ON }                ⎪
          ⎪          {OFF}                ⎪
          ⎨ PROPCHK  {ON } [nodeid]       ⎬
          ⎪          {OFF}                ⎪
          ⎪ LOGGING  {ON }                ⎪
          ⎪          {OFF}                ⎪
          ⎩          {ALL}                ⎭
```

SET DEBUG enters into and exits from programmable operator DEBUG mode.
SET HOSTCHK starts or halts checking of the host system by the distributed system.
SET LOGGING causes the programmable operator facility to stop writing any messages to the log file.
SET PROPCHK restarts or halts checking of the programmable operators on the distributed systems.

```
STOP
```

Stops operation of the programmable operator.

Note: The SET DEBUG command may be entered only at the programmable operator virtual console. The SET LOGGING, SET HOSTCHK, and SET PROPCHK commands may be entered at the programmable operator virtual console, and also from the logical operator's console.

PSERV

CMS

Copies a procedure from the VSE procedure library onto a CMS minidisk or an SFS directory, displays it at the terminal, or spools it to the virtual punch or printer.

PSERV	*procedure*	[*ft* **PROC**]	[(**options...** [ )]]
	Options:	[**DISK**] [**PRINT**] [**PUNCH**] [**TERM**]	

PUNCH

CMS

Spools a specified CMS file to the virtual punch.

PUnch	*fn ft*	[*fm* *]	[(**options...** [ )]]	
	Options:	[**Header** **NOHeader**]	[**MEMber** { * *membername* }]	

PURGE

CP Class D

Deletes a closed spool file before reading, printing, or punching occurs.

```
PURge    [FORCE] ┌ userid ┐ ┌ Reader|RDR                                      ┐
                 │ SYSTEM │ │ Printer|PRT  ┌ ALL                         ┐    │
                 │   *    │ │ PUnch|PCH    │ CLass c1  CLass c2 ... ¹    │    │
                 └        ┘ {              │ spoolid1  spoolid2 ...      │    }
                            │              │ FORM form1  FORM form2      │    │
                            │ ALL          └ DEST dest1  DEST dest2      ┘    │
                            └                                               ┘
```

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

PURGE

CP Class G

Deletes a closed file before reading, printing, or punching occurs.

```
PURge        ┌                                                 ┐
             │ Reader    ┌ CLass c1  CLass c2 ...          ┐ ¹ │
             │ Printer   │ FORM form1  FORM form2 ...      │   │
             { PUnch     │ DEST dest1  DEST dest2 ...      │   }
             │ ALL       │ spoolid1    spoolid2 ...        │   │
             │           └ ALL                             ┘   │
             └                                                 ┘
```

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

PURGE

RSCS

Removes and discards all or specified inactive files from a link.

General User Format:

```
PURge       [*] spoolid
```

Operator Format:

```
PURge     [ linkid ]   { spoolid  [spoolid ... ] }
                       { ALL                     }
```

PUT SCREEN

CMS

Makes a copy of the physical screen and writes the image to a CMS file.

```
PUT SCREEN        fn   ft  [ fm ]
                           [ *  ]
                           [ A1 ]
```

PUT VSCREEN

CMS

Writes the data from the data area of a virtual screen to a CMS file.

```
PUT VSCreen      vname  fn  ft  [ fm  [ fromlin  [ numlin  ] ] ]
                                [ *   [    1     [   *     ] ] ]
                                [ A1  [          [         ] ] ]
```

QUERY

CMS

Requests information about CMS files, minidisks or SFS directories.

```
Query  ┌ ABBREV                                                      ┐
       │ ACCESSED [fm  ]                                             │
       │          [*   ]                                             │
       │          [R/O ]                                             │
       │          [R/W ]                                             │
       │ ALIAS     {fn ft} [ dirid]                                  │
       │           {*  * }                                           │
       │ APL                                                         │
       │ AUThority [fn ft] [ dirid]                                  │
       │           [*  * ]                                           │
       │ AUTOREAD                                                    │
       │ BLIP                                                        │
       │ BORDER    wname  [ ALL ]                                    │
       │ CHARMODE                                                    │
       │ CMSLEVEL                                                    │
       │ CMSPF     [nn]                                              │
       │           [* ]                                              │
       { CMSTYPE                                                     } [(options ...[ )]]
       │ COMDIR                                                      │
       │ CSLLIB                                                      │
       │ CURSOR    [vname ]                                          │
       │ DISK      [fm       ]                                       │
       │           [*        ]                                       │
       │           [R/W      ]                                       │
       │           [MAX      ]                                       │
       │           [FIRSTR/W ]                                       │
       │ DISPLAY                                                     │
       │ DLBL                                                        │
       │ DOS                                                         │
       │ DOSLIB                                                      │
       │ DOSLNCNT                                                    │
       │ DOSPART                                                     │
       │ ENROLL {USEr         } FOR {userid  } [filepoolid:]         │
       │        {ADMinistrator}     {nickname}                       │
       └                            {ALL     }                       ┘
```

(format continued on the next page)

QUERY

(continued from previous page)

```
┌                                                              ┐
  EXECTRAC
  FILEDEF
  FILEPOOL  ┌CURrent                                     ┐
            │PRImary                                     │
            │CONFlict ┌userid  ┐[filepoolid:]            │
            │         └nickname┘                         │
           {                                              }
            │CONNect  FOR ┌userid  ┐[filepoolid:]        │
           {│             {nickname}                     │   } [(options ...[)]]
            └             └ALL     ┘                     ┘
  FILEWAIT
  FULLREAD
  FULLSCREEn
  HIDE       [wname]
             [  *  ]
                _
└                                                              ┘
```

```
Query    ┌IMESCAPE                                          ┐
         │IMPCP                                             │
         │IMPEX                                             │
         │INPUT                                             │
         │INSTSEG                                           │
         │KEY                                               │
         │KEYPROTect                                        │
         │LABELDEF                                          │
         │LANGLIST                                          │
         │LANGUAGE  [ALL]                                   │
         │LDRTBLS                                           │
         │LIBRARY                                           │
         │LIMITS      *  [filepoolid:]                      │
         │LINEND                                            │
         │LOADAREA                                          │
         │LOADLIB                                           │
         │LOCATION  wname                                   │
         │LOCK      [fn ft] [dirid]                         │
         │          [*  * ]                                 │
         │LOGFILE   vname                                   │
         │MACLIB                                            │
         │NAMEDEF                                           │
         │NONDISP                                           │
         │OPTION                                            │
         │OUTPUT                                            │
         │PROTECT                                           │
         {RDYMSG                                            } [(options . . .[)]]
         │REDTYPE                                           │
         │RELPAGE                                           │
         │REMOTE                                            │
         │RESERVED  wname                                   │
         │ROUTE     [msgclass]                              │
         │          [   *    ]                              │
         │              -                                   │
         │SEARCH                                            │
         │SEGMENT   [segname   [CONtents ] ]                │
         │          [           Assign     ]                │
         │          [           SPACE      ]                │
         │          [           PHysical   ]                │
         │          [           LOgical  ] ]                │
         │                                                  │
         │          [   *      [SPACE    ] ]                │
         │              -       PHysical                    │
         │          [          [LOgical  ] ]                │
         │SERVER                                            │
         │SHOW      [wname]                                 │
         │          [  *  ]                                 │
         │             -                                    │
         │STORECLR                                          │
         │SYNONYM   {SYSTEM}                                │
         │          {USER  }                                │
         │          {ALL   }                                │
         └SYSNAMES                                          ┘
```

QUERY

```
Query   ┌ TEXT                                                       ┐
        │ TRANslate [ SYStem  [ TRANslate [ APPLID  applid ] ] ]     │
        │           [ USER    [ SYNonym   [           *    ] ] ]     │
        │           [ ALL     [ BOTH                       ] ]       │
        │                                                            │
        │ TXTLIB                                                     │
        │ UPSI                                                       │
        │ VSCREEN    [ vname  [ ALL ] ]                              │
        <            [   *            ]                              >[(options . . .[ )]]
        │                                                            │
        │ WINDOW     [ wname  [ ALL ] ]                              │
        │            [   =            ]                              │
        │            [   *            ]                              │
        │ WMPF       [ nn ]                                          │
        │            [ *  ]                                          │
        └                                                            ┘

          Options:  [ STACK [ FIFO ] ]
                    [       [ LIFO ] ]
                    [ FIFO           ]
                    [ LIFO           ]
                    [ XEDIT 1        ]
```

[1] XEDIT option is only for QUERY ALIAS and QUERY AUTHORITY.

Query

CP Class A

Displays system software maintenance information, log messages, the number of logged-on users; lists logged-on users. Provides the paging activity index or specified user priority or status of the Virtual Machine Assist feature.

```
Query     AFFInity   [userid]
          CPAssist 1
          JOurnal 2
          LOGmsg
          Names
          PAGing
          PRIORity  userid
          PROCessr
          QDROP
          SASsist 1
          SPMODE
          SRM       APAGes
                    DSPSlice
                    IB
                    MAXDrum
                    MAXWss
                    MHFULL
                    PB
                    PCI
                    PGMStat
                    PGMTlim
          Users      [userid]
```

[1] The collective use of both QUERY CPASSIST and QUERY SASSIST determines the current status of the expanded Virtual Machine Assist portion of the Extended Control-Program Support: VM/370

[2] The JOURNAL operand is valid only if STQUERY=YES is specified in the SYSJRL macro instruction in DMKSYS.

QUERY

CP Class B

Displays system status, paging, scheduling, machine configuration information, system software maintenance information, log messages, the number of logged-on users; lists logged-on users.

```
Query  ┌    ┌ DAsd      ┌Sysvirt┐                         ┐┐
       │    │           └Virtual┘  ┌ACTive ┐  [PATHS]     ││
       │    │                      │ATTach │              ││
       │    │ GRaf                 │FREe   │              ││
       │    │ LINES                │OFFline│              ││
       │    │ UR                   └ALL    ┘              ││
       │    │ TApes                                       ││
       │    └ ALL                                         ┘│
       │ DAsd volid                                        │
       │ DUMP                                              │
       │ L nnn                                             │
       │ LOGmsg                                            │
       │ MITime                                            │
       │ Names                                             │
       │ PROCessr                                          │
       │ raddr1     [- raddr2 ]                            │
       │ lprt                                              │
       │ STATUS  raddr                                     │
       │ STORage                                           │
       │ SYStem  raddr                                     │
       │ TDsk                                              │
       │ userid                                            │
       └ Users    [userid ]                                ┘
```

QUERY

CP Class C

Displays log message, number of logged-on users,the status of CPTRAP, CPLEVEL, specific user ID, and the online processors in the system; lists logged-on users.

```
Query  ┌ CPTrap  ┌ ALL                                            ┐ ┐
       │         │ USER                                           │ │
       │         │ ID trapid                                      │ │
       │         │ SET trapset                                    │ │
       │         │ RECeiver                                       │ │
       │         │       ┌ DATA                                 ┐ │ │
       │         │ TYPE  │ GT                                   │ │ │
       │         │       │ IO                                   │ │ │
       │         │       │ TTable ┌ ALL                       ┐ │ │ │
       │         │       │        │ INTable                   │ │ │ │
       │         │       │        │ INFile                    │ │ │ │
       │         │       └        └ typenum [ typenum...]     ┘ ┘ │ │
       │         └                                                ┘ │
       │ LOGmsg                                                     │
       │ Names                                                      │
       │ PROCessr                                                   │
       └ Users   [userid]                                           ┘
```

QUERY

CP Class D

Provides data on spooling operations.

```
       ┌                                                                          ┐
Query  │ Files    [CLass c] [FORM form] [DEST dest] [userid¹] [Hold    ]          │
       │                                            [  *    ] [NOHold  ]          │
       │                                                      [SYShold ]          │
       │                                                      [USERhold]          │
       │ Hold                                                                     │
       │                                                                          │
       │ Printer ┌ [CLass c] [FORM form] [DEST dest] [userid¹] [Hold    ]┐ [null] │
       │ PUnch   │                                   [  *    ] [NOHold  ]│ [ALL ] │
       │ Reader  ┤                                             [SYShold ]├ [TBL ] │
      <          │                                             [USERhold]│ [PSF ]  >
       │         └ spoolid                                               ┘        │
       │ LOGmsg                                                                   │
       │ Names                                                                    │
       │          ┌ ACTive  ┐                                                       │
       │          │ ATTach  │                                                       │
       │ UR       │ FREe    │  [PATHS]                                              │
       │          │ OFFline │                                                       │
       │          └ ALL     ┘                                                       │
       │                                                                          │
       │ Users    [userid]                                                        │
       └                                                                          ┘
```

[1] Using a one- to four-digit all-numeric *userid* causes unpredictable results for the QUERY command, which also has a one- to four-digit all numeric *spoolid* parameter.

QUERY

CP Class E

Provides the paging activity index or specified user priority or status of the Virtual Machine Assist feature.

```
Query    ┌ AFFInity   [userid ]          ┐
         │ CPAssist 1                    │
         │ JOurnal 2                     │
         │ LOGmsg                        │
         │ Names                         │
         │ PAGing                        │
         │ PRIORity  userid              │
         │ PROCessr                      │
         │ QDROP                         │
         │ SASsist 1                     │
         { SRM    ┌ APAGes  ┐            }
         │        │ DSPSlice│            │
         │        │ IB      │            │
         │        │ MAXDrum │            │
         │        │ MAXWss  │            │
         │        { MHFULL  }            │
         │        │ PB      │            │
         │        │ PCI     │            │
         │        │ PGMStat │            │
         │        └ PGMTlim ┘            │
         └ Users   [userid ]             ┘
```

[1] The collective use of both QUERY CPASSIST and QUERY SASSIST determines the current status of the expanded Virtual Machine Assist portion of the Extended Control-Program Support:VM/370.

[2] The JOURNAL operand is valid only if STQUERY=YES is specified in the SYSJRL macro instruction in DMKSYS.

QUERY

CP Class F

Displays log messages and number of logged-on users, and lists logged-on users.

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

QUERY

CP Class G

Provides system status and machine configuration information.

```
Query  ┌ CPLEVEL                                                              ┐
       │                                                                      │
       │ CPUid                                                                │
       │                                                  ┌ HOld      ┐       │
       │ Files [CLassc ] [FORMform] [ DEST  dest ]  [ * ] │ NOHold    │       │
       │                                                  │ USERhold  │       │
       │                                                  └ SYShold   ┘       │
       │ Links vaddr                                                          │
       │                                                                      │
       │ PF[nn]                                                               │
       │                                                                      │
       │ Printer ┌                                   ┐ ┌ HOld      ┐┐ ┌ ALL ┐ │
       │ PUnch   │ [CLass c] [FORMform] [DESTdest]   │ │ NOHold    ││ │ TBL │ │
       │ Reader  │          spoolid                  │ │ USERhold  ││ └ PSF ┘ │
       │         └                                   ┘ └ SYShold   ┘┘         │
       │ PROCessr                                                             │
       │ SCREen                                                               │
       │ SECuser                                                              │
      <  Set                                                                   >
       │ SPMODE                                                               │
       │ S370E                                                                │
       │ TERMinal                                                             │
       │ Time                                                                 │
       │ USERID                                                               │
       │ ┌                    ┌ ALL                 ┐ ┐                       │
       │ │                    │ CHANNels            │ │                       │
       │ │                    │ CONsole             │ │                       │
       │ │                    │ DAsd                │ │                       │
       │ │                    │ GRaf                │ │                       │
       │ < [Virtual ]         │ LINES               │ >                       │
       │ │                    │ STORage             │ │                       │
       │ │                    │ TApes               │ │                       │
       │ │                    │ UR                  │ │                       │
       │ └                    └ vaddr  [-vaddr ]    ┘ ┘                       │
       │ VMSAVE                                                               │
       └                                                                      ┘
```

QUERY

GCS

Requests information about your GCS virtual machine.

```
Query   ⎧       ⎡ mode ⎤       ⎫
        ⎪       ⎢  *   ⎥       ⎪
        ⎪       ⎢  _   ⎥       ⎪
        ⎪  DISK ⎢ R/W  ⎥       ⎪
        ⎪       ⎣ MAX  ⎦       ⎪
        ⎪                      ⎪
        ⎪  FILEDEF             ⎪
        ⎪  LOADLIB             ⎪
        ⎨  SEARCH              ⎬
        ⎪  SYSNAMES            ⎪
        ⎪  DLBL        [mult]  ⎪
        ⎪  ETRACE              ⎪
        ⎪  ITRACE              ⎪
        ⎪  GROUP               ⎪
        ⎪  LOCK                ⎪
        ⎪  REPLY               ⎪
        ⎪  LOADCMD             ⎪
        ⎩  LOADALL             ⎭
```

The DISK, DLBL, LOADLIB, FILEDEF, SEARCH, and SYSNAMES operands work the same as for the CMS QUERY command with the exception that no options are allowed.

QUERY

TSAF

Requests information about the TSAF configuration when the TSAF virtual machine is running. Only the TSAF virtual console or the secondary user of the TSAF virtual machine can issue this command.

```
Query   ⎧ COLLECT                ⎫
        ⎪ ETRACE                 ⎪
        ⎪ GATEWAY                ⎪
        ⎪ LINKs      ⎡vdev  ⎤    ⎪
        ⎨            ⎢*     ⎥    ⎬
        ⎪            ⎣ALL   ⎦    ⎪
        ⎪ RESOURCE               ⎪
        ⎪ ROUTEs     ⎡nodeid⎤    ⎪
        ⎪            ⎢*     ⎥    ⎪
        ⎩            ⎣ALL   ⎦    ⎭
```

QUERY **RSCS**

Requests system information for a link, a file, or for the system in general.

```
Query     { nodeid                                        }
          {                                               }
          {          [ Active ]                           }
          {          [ Def    ]                           }
          { linkid   [ Files  ]                           }
          {          [ Queue  ]                           }
          {          [ Sum    ]                           }
          {                                               }
          {          { spoolid  }   [ Status ]            }
          { File     { *spoolid }   [ Rscs   ]            }
          {          { *        }   [ Vm     ]            }
          {                                               }
          {          [ Active    ]                        }
          {          [ Dest      ]                        }
          {          [ EXits     ]                        }
          {          [ LEVel     ]                        }
          {          [ Links     ]                        }
          { SYstem   [ LOcal     ]                        }
          {          [ NETwork   ]                        }
          {          [ Ports     ]                        }
          {          [ Queue     ]                        }
          {          [ REroutes  ]                        }
          {          [ Routes    ]                        }
```

QUERY DISABLE **CMS**

Allows an operator to determine if a storage group or filespace has been previously disabled, and the user ID of the disabler. (File pool server operator only.)

```
QUERY DISABLE    { GROUP group-num      }
                 { FILESPACE userid     }
```

QUERY FILEPOOL CONFLICT

CMS

Displays information about lock conflicts in the specified file pool.

```
Query FILEPOOL CONFlict   { userid   }   [filepoolid: ]  [(options... [ )]]
                          { nickname }

                Options:  [ STACK [FIFO]   ]
                          [       [LIFO]   ]
                          [                ]
                          [ LIFO           ]
                          [ FIFO           ]
```

QUERY FILEPOOL STATUS

CMS

Displays information about a specified file pool and file pool server processing against it. (File pool server operator only.)

```
Query FILEPOOL STAtus   [ filepoolid: ]  [(options... [ )]]

              Options:    [ STACK [FIFO] ]
                          [       [LIFO] ]
                          [ LIFO         ]
                          [ FIFO         ]
                          [ XEDIT        ]
                          [              ]
                          [ CATalog ]
```

QUERY LIMITS

CMS

Displays limits information (assigned storage group, 4K file block allocation and usage, file block warning threshold value) about selected users in a particular file pool.

```
Query LIMITS      ┌ FOR  {userid  } ┐  [filepoolid:]   [[(options... [)]]
                  │      {nickname} │
                 <  *                >
                  │                 │
                  └ ALL             ┘

                  Options:   ┌ STACK [FIFO] ┐
                             │       [LIFO] │
                             │ LIFO         │
                             │ FIFO         │
                             │              │
                             └              ┘
```

QVM

CP Class A

Requests the transition from VM/370 to a particular virtual machine, running in native mode.

```
QVM          userid     [NORETURN ]
```

R

CMS Border Command

Scrolls the window to the right.

```
R
```

RDR

CMS

Generates a return code and either displays or stacks a message that identifies the characteristics of the next file in your virtual reader.

```
RDR        [spool-class]        [(options...[)]]
           [     =     ]

      Options:  [ NOTYPE           ]  [MSGSUBS]
                [ STACK    [FIFO]  ]  [MSGALL ]
                [          [LIFO]  ]
                [ FIFO             ]
                [ LIFO             ]
```

RDRLIST **CMS**

Displays information about files in your virtual reader with the ability to issue commands from a list.

```
RDRList    [ (options... [)]]
  RList

           Options:   [PROFile fn]   [Append]
```

Special Commands which can be used in the RDRLIST environment, followed by their descriptions, are:

```
EXECUTE   ┌Cursor┐  [command]
          └lines ┘
```

Issues CP/CMS commands (or EXECs) which make use of the reader spool files displayed by RDRLIST.

```
DISCARD   ┌fn ft  ┌fm    ┐┐
          │       └dirid ┘│
          └dirid          ┘
```

Purges a file displayed in RDRLIST.

READCARD

CMS

Reads data from the spooled card input device.

```
READcard   ⎧ fn ft ⎡fm⎤       ⎫
           ⎪       ⎣A ⎦       ⎪
           ⎨                  ⎬   [ (options...[)] ]
           ⎪ * ⎡* ⎡fm⎤⎤       ⎪
           ⎩   ⎣  ⎣A ⎦⎦       ⎭

           Options:   ⎡Fullprompt⎤   ⎡Replace  ⎤
                      ⎢Minprompt ⎥   ⎣NOReplace⎦
                      ⎣NOPrompt  ⎦
```

READY

CP Class G

Makes a device-end interruption pending for the specified device.

```
READY      vaddr
```

READY

RSCS

Notifies RSCS that a forms mount has been satisfied, or that a setup page is wanted. This command is for RJE, 3270P, SNA3270P, and MRJE type links.

```
Ready      [ linkid ]
```

RECEIVE

CMS

Reads to your SFS directory or minidisk a file or note that is in your virtual reader.

```
RECEIVE   [spoolid [fn [ft [fm ]]]]   [ (options... [ )] ]

          Options:

            [NOTebook fn]  [Log  ]  [Purge]  [Fullprompt]  [Replace  ]
            [NOTebook * ]  [NOLog]           [Minprompt ]  [NOReplace]
                                             [NOPrompt  ]

            [Olddate]  [STack]
            [NEwdate]
```

RECONN

RSCS

Reconnects the RSCS operator console after being disconnected and resets the user ID, if any, that was used on the DISCONN command (for RSCS operator only).

```
REConn
```

REFRESH

CMS

Updates virtual screens and their associated windows, and refreshes the screen.

```
REFresh
```

RELEASE

CMS

Frees a previously accessed SFS directory or minidisk.

```
RELease   ┌vdev ┐    [(DET[)]]
          ┤dirid├
          └fm   ┘
```

RELEASE

GCS

Releases a disk.

```
RELease   ┌cu u ┐  [(DET[)]]
          └mode ┘
```

RELOCATE

CMS

Moves a file or subtree from one directory to another, within the same user ID or file pool.

```
RELOcate   [fn ft]  dirid1  TO  dirid2 [(options...[)]]
           [*  * ]

           Options:  ┌                    ┐
                     │ TYPe               │
                     │ NOType             │
                     │ STACK  ┌FIFO┐      │
                     │        └LIFO┘      │
                     │ FIFO               │
                     │ LIFO               │
                     └                    ┘
```

RENAME

CMS

Changes the name of a CMS file or directory.

```
Rename    { fileid1   fileid2 }   [(options...[)]]
          { dirid1    dirid2  }

          Options:  [ Type              ]  [ UPdirt   ]
                    [ NOType            ]  [ NOUPdirt ]
                    [ STACK  [ FIFO ]   ]
                    [        [ LIFO ]   ]
                    [ FIFO              ]
                    [ LIFO              ]
```

REORDER

RSCS

Causes all inactive spool files which are owned by the RSCS virtual machine to be re-enqueued for transmission on the appropriate links, based on the files' TAG information and the RSCS link and route tables. No other commands will be accepted until the REORDER is completed.

```
REORDer
```

REPEAT

CP Class D

Holds or increases the copies of an output spool file.

```
REPeat    {raddr}  [ [nnn]          ]
          {lprt }  [ [ 1 ]          ]
                   [                ]
                   [ {nnn}  HOld    ]
```

REPLY

GCS

Replies to messages sent to the GCS operator.

```
Reply        id   [text]
```

REQUEST

CP Class G

Makes an attention interruption pending.

```
REQuest
```

REROUTE

RSCS

Modifies the original routing (the destination system and user ID of files and messages for specific systems and user IDs (for RSCS operator only).

```
RERoute      ⎧⎧FIles⎫        ⎧nodeid⎫  ⎧userid ⎫⎫
             ⎪⎨Msgs ⎬  [FOR] ⎨*     ⎬  ⎨SYSTEM ⎬⎪
             ⎪⎩ALL  ⎭        ⎩ANY   ⎭  ⎩ANY    ⎭⎪
             ⎨                                  ⎬
             ⎪                ⎧userid⎫          ⎪
             ⎩NOTrcvg  [FOR]  ⎩ANY   ⎭          ⎭

                   ⎧         ⎧nodeid⎫  ⎧userid ⎫⎫
                   ⎨[TO]     ⎨*     ⎬  ⎨SYSTEM ⎬⎬
                   ⎪         ⎩=     ⎭  ⎩=      ⎭⎪
                   ⎩OFF                         ⎭
```

RESERVE

CMS

Allocates all available blocks of a 512-, 1K-, 2K-, or 4K-byte block-formatted minidisk to a unique CMS file.

RESERVE *fn ft fm*

RESET

CP Class G

Clears all pending interruptions; resets error conditions on the device specified.

RESET *vaddr*

RESTORE WINDOW

CMS

Returns a maximized or minimized window to its size and location prior to a maximize or minimize command.

REStore WINdow $\begin{bmatrix} \textit{wname} \\ \underline{=} \end{bmatrix}$

REVOKE ADMIN

CMS

Deletes file pool administration authority from a user ID. (File pool server operator only.)

```
REVOKE ADMIN    userid
```

REVOKE AUTHORITY

CMS

Unauthorizes specified users from one or more of your files or directories.

```
REVoke AUThority  [fn  ft]  dirid  FROM  {userid  }  [(options...[)]]
                  [*   * ]               {nickname}
                                         {PUBlic  }
                                         {ALL     }

                  Options:  [KEEpread]  [TYPe              ]
                                        [NOType            ]
                                        [STACK  [FIFO]     ]
                                        [       [LIFO]     ]
                                        [LIFO              ]
                                        [FIFO              ]
```

REWIND

CP Class G

Rewinds a real tape drive.

```
REWind        vaddr
```

REXX

CMS

The Restructured Extended Executor (REXX) language is a command programming language that lets you combine useful sequences of commands to create new commands. The System Product Interpreter processes programs written in REXX. This language is not only suitable for writing execs or editor macros, but is also a useful tool for algorithm development.

Instructions:

The formats of the REXX instructions, followed by their descriptions, are:

```
ADDRESS  [ environment [expression ] ]  ;
         [ [VALUE ] expression       ]
```

Effects a temporary or permanent change to the destination of command(s).

```
ARG  [template ]  ;
```

Retrieves the argument strings provided to a program or internal routine and assigns them to variables. It is a short form of the instruction PARSE UPPER ARG [template];.

```
CALL   name [expression ][,[expression ]] . . .  ;
```

Calls an internal routine, an external routine, or a built-in function. The called routine may optionally return a result upon its completion.

```
DO ┌ name=expri  [TO exprt ] [BY exprb] [FOR exprf ]┐ ┌WHILE exprw┐
   │ ;                                              │ └UNTIL expru┘
   │ FOREVER                                        │
   └ exprr                                          ┘

    ┌ instruction ┐
    │      :      │
    └             ┘

END [ symbol ] ;
```

Or, to present the instruction more generally:

```
DO [repetitor ] [ conditional ] ;

    ┌ instruction ┐
    │      :      │
    └             ┘

END [symbol] ;
```

Groups instructions together and optionally executes them repetitively.

```
DROP   name [name] [name]...   ;
```

"Unassigns" variables; that is, restores them to their original uninitialized state.

```
EXIT [expression ] ;
```

Unconditionally leaves a program, and optionally returns a data string to the caller. The program is immediately terminated.

```
IF     expression [;] THEN [;] instruction
                   [ELSE [;] instruction ]
```

Conditionally executes an instruction or group of instructions.

INTERPRET *expression* ;

Executes instructions that have been built dynamically by evaluating an expression (rather than that exist permanently in the program).

ITERATE [*name*] ;

Alters the flow within a repetitive DO loop (that is, any DO construct other than that with a simple DO).

LEAVE [*name*] ;

Causes an immediate exit from one or more repetitive DO loops (that is, any DO construct other than that with a simple DO).

NOP ;

NOP is a dummy instruction that has no effect. It can be useful as the target of a THEN or ELSE clause.

```
NUMERIC  ⎧DIGITS [expression]               ⎫ ;
         ⎪FORM   ⎡SCIENTIFIC               ⎤⎪
         ⎨       ⎢ENGINEERING              ⎥⎬
         ⎪       ⎣[VALUE] expression       ⎦⎪
         ⎩FUZZ   [expression]               ⎭
```

Changes the way in which arithmetic operations are carried out.
NUMERIC DIGITS controls the precision to which arithmetic operations will be carried out.
NUMERIC FORM sets the form of exponential notation to be used.
NUMERIC FUZZ controls how many digits, at full precision, will be ignored during a comparison operation.

```
OPTIONS  [expression] ;
```

Passes special requests or parameters to the language processor. *Expression* is evaluated, and if the result is one of the following words recognizable to the language processors, it is obeyed. Words not recognized are ignored.

ETMODE	DBCS strings can be used in the program.
NOETMODE	DBCS strings cannot be used in the program (this is the default).
EXMODE	DBCS data operations capability is enabled.
NOEXMODE	DBCS data operations capability is disabled.

```
PARSE  [UPPER] ⎧ARG                          ⎫  [template]  ;
               ⎪EXTERNAL                     ⎪
               ⎪NUMERIC                      ⎪
               ⎨PULL                         ⎬
               ⎪SOURCE                       ⎪
               ⎪VALUE [expression] WITH      ⎪
               ⎪VAR name                     ⎪
               ⎩VERSION                      ⎭
```

Assigns data (from various sources) to one or more variables according to the rules of parsing.

PROCEDURE [**EXPOSE** *name* [*name*][*name*]. . .] ;

Used within an internal routine (subroutine or function), PROCEDURE protects all the existing variables by making them unknown to following instructions.

PULL [*template*];

Reads a string from the head of the queue. It is just a short form of the instruction. PARSE UPPER PULL [template];.

PUSH [*expression*] ;

The string resulting from expression will be stacked LIFO onto the queue, limited to 255 characters per entry. If no expression is specified, a null string is stacked.

QUEUE [*expression*] ;

The string resulting from expression will be appended to the most recently created buffer of the program stack (system-provided data queue) limited to 255 characters per entry. That is, it will be stacked FIFO. If no expression is specified, a null string is stacked.

RETURN [*expression*] ;

Returns control (and possibly a result) from a REXX program or internal routine to the point of its invocation.

```
SAY  [expression] ;
```

The result of evaluating the expression is written to the output stream (usually displayed to the user). The result of the expression may be of any length.

```
SELECT
     WHEN  expression [;]  THEN  [;]  instruction
   [ WHEN  expression [;]  THEN  [;]  instruction ]
   [   .       .                .          .      ]
   [   .       .                .          .      ]
   [   .       .                .          .      ]

   [ OTHERWISE [;]      [ instruction ] ]
   [                    [      .      ] ]
   [                    [      .      ] ]
END ;
```

Conditionally executes one of several alternative instructions.

```
SIGNAL   { labelname                          }  ;
         { [VALUE] expression                 }
         {                                    }
         {          { ERROR    }              }
         {  [ON ]   { FAILURE  }              }
         {  [OFF]   { HALT     }              }
         {          { NOVALUE  }              }
         {          { SYNTAX   }              }
```

Causes an **abnormal** change in the flow of control, or (if ON or OFF is specified) controls the trapping of exceptions.

```
TRACE  [ [? [?...]]   [ ALL           ] ] ;
       [ [  [!...]]   [ COMMANDS      ] ]
       [              [ ERROR         ] ]
       [ [! [?...]]   [ FAILURE       ] ]
       [ [  [!...]]   [ INTERMEDIATES ] ]
       [              [ LABELS        ] ]
       [              [ NORMAL        ] ]
       [              [ OFF           ] ]
       [              [ RESULTS       ] ]
       [ [number]     [ SCAN          ] ]
```

Or, alternatively:

```
TRACE  [ string                    ] ;
       [ [VALUE] expression        ]
       [ symbol                    ]
```

Controls the tracing action taken (that is, how much will be displayed to the user) during execution of a REXX program; primarily used for debugging.

```
UPPER   variable [variable] [variable]...  ;
```

Translates the contents of one or more variables to uppercase. The variables are translated in sequence from left to right.

Built-in Functions:

REXX has many built-in functions and also various functions that are supplied externally.

ABBREV(*information,info* [*,length*])

Returns 1 if *info* is a true abbreviation of *information*, with minimum *length*, or 0 if either condition is not met.

ABS(*number*)

Returns the absolute value of *number*.

ADDRESS( )

Returns the name of the current environment for commands.

ARG([*n*[*,option*]]**)**

Returns the number of arguments, the *n*th argument, or tests if the nth argument exists or not.

BITAND(*string1* [, [*string2*][*,pad*]] **)**

Returns a string composed of the two input strings logically ANDed together, bit by bit.

BITOR(*string1* [, [*string2*] [*,pad*]] **)**

Returns a string composed of the two input strings logically ORed together, bit by bit.

BITXOR(*string1* [, [*string2*] [*,pad*]] **)**

Returns a string composed of the two input strings logically exclusive ORed together, bit by bit.

CENTER(*string, length* [*, pad*])
CENTRE(*string, length* [*, pad*])

Returns a string of length *length* with *string* centered in it and *pad* characters added as necessary to make up length.

COMPARE(*string1,string2*[*,pad*])

Returns 0 if the strings are identical. If they are not, returns a nonzero number which is the position of the first character that does not match.

COPIES (*string,n* **)**

Returns *n* concatenated copies of *string*.

C2D(*string* [*n* ,])

Character to Decimal. Returns the decimal value of the binary representation of *string*.

C2X(*string***)**

Character to Hexadecimal. Returns the hexadecimal representation of *string*.

DATATYPE (*string* [,*type*])

If only *string* is specified, returns *NUM* if *string* is a valid REXX number (any format); otherwise *CHAR* is returned. If *type* is specified, returns 1 if *string* matches *type*; otherwise 0 is returned.

DATE ([*option*])

Returns the local date in the format: dd mon yyyy or in the format according to *option*.

DELSTR(*string,n* [,*length*])

Deletes the substring of *string* that begins at the *n*th character, and is of length *length*. Returns the changed string.

DELWORD(*string,n* [,*length*])

Deletes the substring of *string* that starts at the *n*th word, and is of length *length* blank-delimited words. Returns the changed string.

D2C (*wholenumber* [,*n*])

Decimal to Character. Returns a character string which is the binary representation of *wholenumber*. The length of the returned string may be specified by *n*

D2X(*wholenumber* [,*n*])

Decimal to Hexadecimal. Returns a string which is the hexadecimal representation of *wholenumber* The length of the returned string may be specified by *n*.

ERRORTEXT(*n*)

Returns the error message associated with error number *n*.

EXTERNALS()

Returns the number of lines in the terminal input buffer (system external queue).

FIND (*string* , *phrase*)

Returns the word number of the first word of *phrase* in *string*. If *phrase* is not found, then 0 is returned.

FORM()

Returns the current setting of NUMERIC FORM.

FORMAT(*number*[,[*before*][,[*after*][,[*expp*][,[*expt*]]]])

Rounds and formats *number* to specified integer (before) and (after) decimal places. Exponential places and trigger point may be controlled with *expp* and *expt*, respectively. Returns the formatted number.

FUZZ( )

Returns the current setting of NUMERIC FUZZ.

INDEX (*haystack,needle* [*,start*])

Returns the character position of one string, *needle*, in another, *haystack*, beginning at *start*.

INSERT (*new,target* [,[*n*][,[*length*][*,pad*]]])

Inserts the string *new*, padded with *pad* to length *length*, into the string *target* after the *n*th character. Returns the changed target.

JUSTIFY (*string,length* [*,pad*])

Formats blank-delimited words in *string*, by adding *pad* characters between words to justify to both margins. Returns the formatted string.

LASTPOS (*needle,haystack* [*,start*])

Returns the position of the last occurrence of one string, *needle*, in another, *haystack*, beginning at *start*.

LEFT (*string,length* [*,pad*])

Returns a string of length *length* with *string* left-justified in it. The returned string is padded with *pad* characters on the right, as needed.

LENGTH(*string*)

Returns the length of *string*.

LINESIZE ( )

Returns the current terminal line width (the point at which the interpreter will break lines displayed using the SAY instruction).

MAX (*number* [*,number*]. . .)

Returns the largest number out of the list specified.

MIN (*number* [*,number*]. . .)

Returns the smallest number out of the list specified.

OVERLAY (*new,target* [, [*n*][, [*length*] [,*pad*]]])

Overlays the string *new*, padded with *pad* or truncated to length *length*, onto the string *target* starting at the *n*th character. Returns the overlaid target.

POS (*needle,haystack* [,*start*])

Returns the position of one string, *needle*, in another, *haystack*, beginning at start.

QUEUED ( )

Returns the number of lines in the program stack (system-provided data queue).

RANDOM([*min*][,[*max*][,*seed*]])

Returns a pseudo-random nonnegative whole number in the range of 0-999 or *min* to *max* inclusive. The generator *seed* may be specified.

REVERSE(*string*)

Returns *string*, inverted.

RIGHT (*string,length* [,*pad*])

Returns a string of length *length* with *string* right-justified in it. Returned string is padded with *pad* characters on the left, as needed.

SIGN (*number*)

Returns the sign of *number* (-1, 0, or 1), after rounding to the current setting of NUMERIC DIGITS.

SOURCELINE ([*n*])

Returns the line number of the final line in the source file or the *n*th line.

SPACE (*string* [, [*n*][,*pad*]])

Formats the blank-delimited words in *string* with *n pad* characters between each word. If *n* is 0, all blanks are removed. Returns the formatted string.

STRIP (*string* [, [*option*][,*char*]])

Removes Leading, Trailing, or Both blanks or *char*s from *string* when the first character of *option* is L, T, or B respectively (the default is B). Returns the changed string.

SUBSTR (*string,n* [, [*length*] [,*pad*]])

Returns the substring of *string* that begins at the *n*th character. The length of the returned string may be specified with *length* and padded with *pad*, if necessary.

SUBWORD (*string,n* [*,length*])

Returns the substring of *string* that starts at the *n*th word, and is of length *length* blank-delimited words.

SYMBOL (*name*)

If *name* is not a valid REXX symbol, BAD is returned. If it is the name of a variable, VAR is returned. Otherwise LIT is returned.

TIME ([*option*])

Returns the local time in the 24-hour clock format: hh:mm:ss (hours, minutes, and seconds). All calls in one expression are synchronized.

TRACE([*option*])

Returns current trace setting, or sets new trace *option*.

TRANSLATE (*string* [, [*tableo*][, [*tablei*] [*,pad*]]])

Translates characters in *string* to be other characters, or may be used to reorder characters in a string. If neither translate table is given, *string* is simply translated to uppercase. The output table is padded with *pad*, or truncated as necessary. Returns the translated string.

TRUNC *number* [*,n*]

Returns the integer part of the number, and *n* decimal places. The default *n* is zero.

USERID ( )

Returns the system-defined user identifier.

VALUE (*name*)

Returns the value of the symbol *name*.

VERIFY(*string* , *reference* [, [*Nomatch* | *Match*][, *start*]])

If *Nomatch* (the default) is given, then the index of the first character is *string* that is not in *ref* is returned. If *Match* is given, then the index of the first character in *string* that is in *ref* is returned. *Start* may be used to specify the starting point of the search.

WORD (*string,n*)

Returns the *n*th blank-delimited word in *string*.

WORDINDEX(*string,n*)

Returns the position of the *n*th blank-delimited word in *string*.

WORDLENGTH(*string,n*)

Returns the length of the *n*th blank-delimited word in *string*.

WORDPOS (*phrase, string*[*, start*])

Returns the word number of the first word of *phrase* in *string*. Returns '0' if *phrase* is not found. *Start* may be used to specify the starting point of the search.

WORDS *(string)*

Returns the number of blank-delimited words in *string*.

XRANGE ([*start*][*, end*])

Returns a string of all one-byte codes between and including the values *start* and *end*.

X2C (*hexstring*)

Hexadecimal to Character. Converts hexadecimal *hexstring* to character(s) and returns those character(s).

X2D(*hexstring* [*,n*])

Hexadecimal to Decimal. Converts *hextring* to a decimal number (unsigned unless the length *n* is specified) and returns that number.

RXSYSFN PACKAGE OF CP/CMS FUNCTIONS

These all provide useful CP or CMS functions. The package is automatically loaded when needed. The formats are followed by their descriptions.

CMSFLAG(*flag* **)**

Returns the setting of one of the specified CMS flags (the entire name of the flag must be given):

ABBREV
IMPCP
AUTOREAD
IMPEX
CMSTYPE
PROTECT
DOS
RELPAGE
EXECTRAC
SUBSET

CSL (*rtnname retcode* [*parms*]**)**

Calls routine *rtnname*, with parameters *parms*, that resides in a callable services library (CSL). *Retcode* receives the CSL routine's return code.

DIAG(*n* [*?*] [*,data*] [*,data*]. . . **)**

Communicates with CP via a dummy DIAGNOSE instruction and returns data as a character string. The following hexadecimal diagnose codes (specified with *n*) are supported by DIAG:

DIAG (00)
DIAG (08)
DIAG (0C)
DIAG (14)
DIAG (24)
DIAG (5C)
DIAG (60)
DIAG (64)
DIAG (8C)
DIAG (C8)
DIAG (CC)

DIAGRC(*n* [*?*] [*,data*] [*,data*]. . .] **)**

Is identical to the DIAG function, except that CP return code and condition code are prefixed to the result. Has the same diagnose codes as DIAG.

STORAGE ([*address* [, [*length*] [,*data*]]])

Returns the current virtual machine size if no arguments are specified; else returns *length* bytes from user's memory starting at address *address*. If *data* is specified, it is stored at *address*.

RO

CMS

Resumes recording of trace information previously suspended by the SO Immediate command.

RO

ROUTE

CMS

Directs data of a particular message class to a virtual screen.

```
ROUTE     msgclass  TO  vname   [( options ... [)]]

          Options:  [ALARM  ]  [NOTify  ]
                    [NOALARM]  [NONotify]
```

ROUTE

RSCS

Temporarily adds, deletes, or alters an RSCS routing table entry (for RSCS operator only).

```
ROUte     nodeid   {TO  linkid}
                   {OFF       }
```

RSERV

CMS

Copies a VSE relocatable module onto a CMS minidisk or SFS directory, displays it at the terminal, or spools a copy to the virtual punch or printer.

```
RSERV     modname   [ ft   ]        [(options...[)]]
                    [ TEXT ]

          Options:  [DISK]  [PRINT]  [PUNCH]  [TERM]
```

RTNDROP

CMS

Undoes the binding of a callable services library routine.

```
RTNDrop      {namelist}              [(options... [)]]
             {   *    }

                options:

                 [User          ]    [TYpe  ]
                 [SYstem        ]    [NOType]
                 [GRoup grpname ]
```

RTNLOAD

CMS

Searches for, loads, and binds a callable services library routine to a fixed location in storage, and makes it available for invocation.

```
RTNLoad   ┌namelist ┌ ( ┌FRom  *                    ┐ ┌ALias   ┐ [other options...[)]]┐┐
          │         │   │FRom library ┌IN fm   ┐   │ └NOAlias ┘                      ││
          │         └   └             └IN dirid┘   ┘                                 ┘│
          ┤                                                                           ├
          │   *      ( FRom library ┌IN fm   ┐ [NOAlias]  [other options...[)]]       │
          └                         └IN dirid┘                                        ┘

          Other options:

          ┌User  ┐  [GRoup grpname]  ┌TYpe   ┐  ┌PUsh  ┐
          └SYstem┘                   └NOType ┘  └NOPush┘
```

RTNMAP

CMS

Displays information about the callable services library routines that are currently loaded and bound to an address.

```
RTNMap       ┌runname┐        [(options... [)]]
             └   *   ┘

          Options:

              ┌STACK [FIFO]┐
              │STACK  LIFO │
              │FIFO        │
              └LIFO        ┘

              ┌User  ┐  [GRoup grpname]  ┌Header  ┐   [ALL]
              └SYstem┘                   └NOHeader┘
```

RTNSTATE

CMS

Obtains the status of one or more specific callable services library routines.

```
RTNState      { runname }          [ (options... [)]]
              {    *    }
               Options:
                [ User   ]  [GRoup grpname]
                [ SYstem ]
```

RT

CMS

Resumes terminal displaying.

```
RT
```

RUN

CMS

Initiates a series of functions to be performed on a source, MODULE, TEXT, or EXEC file.

```
RUN        fn [ ft  [fm]] [(args...[)]]
```

RUNTSAF

TSAF

Starts the TSAF virtual machine. Only the TSAF virtual console or the secondary user of the TSAF virtual machine can issue this command.

```
RUNTSAF   [nnn] [ETRACE]
          [40 ]
```

S

CMS Border Command

Changes the size of the window.

```
S
```

SAMGEN

CMS

Builds the CMSBAM physical saved segment.

```
SAMGEN
```

SAVEFD

CMS

Places file directory information for a shared, extended data format (EDF) R/O minidisk into a discontiguous shared segment (DCSS). The DCSS is then available to users who access the disk R/O.

```
SAVEFD    ┌INIT vdev label segname  ┐
          ┤SAVE vdev label segname  ├
          └NOSAVE vdev label        ┘
```

SAVENCP

CMS

Reads 370x control program load into virtual storage and saves an image on a CP-owned disk.

Note: CP command privilege class A, B, or C is required to use SAVENCP.

```
SAVENCP   fn   [(options ...)]

          Options:

          [ENTRY symbol]      [NAME ncpname]
          [CXFINIT     ]      [fn          ]

          [LIBE libraryname]  [CAMOD {0}]
          [fn              ]  [      {1}]
```

SAVESYS

CP Class E

Creates a copy of virtual machine storage, registers, and PSW.

```
SAVESYS       systemname
```

SCREEN

CP Class G

Alters or changes color and extended highlighting for the virtual machine display area, as well as the color in the input area and the status area.

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

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

SCROLL

CMS

Moves a window to a new location on the virtual screen to which it is connected.

```
SCROLL  ┌ BAckward   [ wname  [ n ] ]  ┐
        │            [ =      [ * ] ]  │
        │            [        [ 1 ] ]  │
        │                              │
        │ Bottom     [ wname ]         │
        │            [ =     ]         │
        │                              │
        │ Down       [ wname  [ n ] ]  │
        │            [ =      [ * ] ]  │
        │            [        [ 1 ] ]  │
        │                              │
        │ Forward    [ wname  [ n ] ]  │
        │            [ =      [ * ] ]  │
        │            [        [ 1 ] ]  │
        │                              │
        { Left       [ wname  [ n ] ]  }
        │            [ =      [ 1 ] ]  │
        │                              │
        │ Next       [ wname  [ n ] ]  │
        │            [ =      [ * ] ]  │
        │            [        [ 1 ] ]  │
        │                              │
        │ Right      [ wname  [ n ] ]  │
        │            [ =      [ 1 ] ]  │
        │                              │
        │ Top        [ wname ]         │
        │            [ =     ]         │
        │                              │
        │ Up         [ wname  [ n ] ]  │
        │            [ =      [ * ] ]  │
        └            [        [ 1 ] ]  ┘
```

SEGGEN

CMS

Builds and saves a physical saved segment that is composed of one or more logical saved segments.

```
SEGGEN   fn  [ft    [fm  [fn2     [ft2    [fm2]]]]]    [( options  [) ]
             [PSEG  [*   [SYSTEM  [SEGID  [*2 ]]]]]

         Options:  [MAP  ]  [GEN  ]
                   [NOMAP]  [NOGEN]
```

SEGMENT ASSIGN

CMS

Indicates the logical segment to be associated with the physical segment.

```
SEGMENT  ASSIGN    lsegname    psegname
```

SEGMENT LOAD

CMS

Loads a saved segment.

```
SEGMENT  LOAD    segname      [(options...[)]]

                 Options:  [SYSTEM]  [SHare  ]
                           [USER  ]  [NOSHare]
```

SEGMENT PURGE

CMS

Purges a saved segment.

SEGMENT PURGE *segname*

SEGMENT RELEASE

CMS

Releases the storage held by a segment space.

SEGMENT RELEASE *name*

SEGMENT RESERVE

CMS

Creates a segment space for subsequent loading.

SEGMENT RESERVE *name* [(options...[)]]

Options: [SYSTEM | USER]

SEND

CP Class G

Passes commands and message replies to the designated disconnected virtual machines for execution.

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

SENDFILE

CMS

Sends files or notes to one or more computer users, locally or remotely attached, by entering the command or by using a menu (display terminal only).

```
SENDFile    [fn ft [fm ][[TO] name...] [(options...[)]]]
SFile
            Options:
            [Ack  ] [Filelist  ] [Log  ] [NEw] [NOTE]
            [NOAck] [NOFilelist] [NOLog] [Old]

            [Type  ]
            [NOType]
```

SENTRIES

CMS

Determines the number of lines currently in the program stack.

```
SENTRIES
```

SET **CMS**

Establishes, sets or resets CMS virtual machine characteristics.

Options available with SET are listed here; a complete description of each option follows this section.

ABBREV
APL
AUTOREAD
BLIP
BORDER
CHARMODE
CMSPF
CMSTYPE
COMDIR
DOS
DOSLNCNT
DOSPART
EXECTRAC
FILEPOOL
FILEWAIT
FULLREAD
FULLSCREEN
IMESCAPE
IMPCP
IMPEX
INPUT
INSTSEG
KEYPROTECT
LANGUAGE
LDRTBLS
LINEND
LOADAREA
LOCATION
LOGFILE
NONDISP
NONSHARE
OUTPUT
PROTECT
RDYMSG
REDTYPE
RELPAGE
REMOTE
RESERVED
SERVER
STORECLR
SYSNAME
TEXT
THRESHOLD
TRANSLATE
UPSI
VSCREEN
WINDOW
WMPF

SET ABBREV

CMS

Controls whether the system ignores user abbreviations of system commands and execs or accepts only the full system command name or the full user synonym (if one is available) for system commands.

```
SET          ABBREV  {ON }
                     {OFF}
```

SET APL

CMS

Activates character code conversion to APL characters for windows.

```
SET        APL    {ON }
                  {OFF}
```

SET AUTOREAD

CMS

Specifies whether a console read is to be issued immediately after command execution, or not until the ENTER key (or its equivalent) is pressed.

```
SET          AUTOREAD  {ON }
                       {OFF}
```

SET BLIP

CMS

Turns ON or OFF the BLIP character string displayed at the terminal to indicate every two seconds of virtual interval timer time.

```
SET        BLIP {string [(count )]}
                {ON                }
                {OFF               }
```

SET BORDER

CMS

Defines borders around windows.

```
SET      BORDER   wname  {ON }  [( [optionA]  [optionB] [)] ]
                         {OFF}

         OptionA: [ TOP char ]    [BOTTOM char ]
                  [ LEFT char]  [RIGHT char]  [ALL char]

         OptionB: [ High  ]  [color] [exthi ] [psset ]
                  [ NOHigh]
```

SET CHARMODE

CMS

Specifies whether character attributes should be used when displaying virtual screen data on the physical screen.

```
SET      CHARMODE     { ON }
                      { OFF}
```

SET CMSPF

CMS

Defines a command that should be executed when a specified PF key is pressed in CMS full-screen mode.

```
SET     CMSPF   nn   [ {pseudonym }  [keyword ]   string ]
                     [ {NOWRITE   }  [DELAYED ]          ]
```

SET CMSTYPE

CMS

Specifies suppression of a CMS terminal display within an EXEC.

```
SET           CMSTYPE  {HT}
                       {RT}
```

SET COMDIR

CMS

Sets up and controls the CMS communications directory.

```
SET   COMDIR  ┌ FILE     {SYSTEM} {NONE  } ┐
              │          {USER  } {fileid} │
              │                            │
              │ ON       {SYSTEM}          │
              │          {USER  }          │
              │          {BOTH  }          │
              {                            }
              │ OFF      {SYSTEM}          │
              │          {USER  }          │
              │          {BOTH  }          │
              │ RELOAD   {SYSTEM}          │
              │          {USER  }          │
              │          {BOTH  }          │
              └                            ┘
```

SET DOS

CMS

Indicates whether your CMS virtual machine is in CMS/DOS environment, specifies the mode letter at which the VSE system residence is accessed, and specifies that you are going to use the AMSERV command or you are going to execute programs to access VSAM data sets.

```
SET        DOS {ON  [fm    [(VSAM[)]]]}
               {OFF                   }
```

SET DOSLNCNT

CMS

Specifies the number of SYSLST lines per page.

```
SET        DOSLNCNT nn
```

SET DOSPART

CMS

Specifies control regarding the size of the virtual partition in which you want a program to execute.

```
SET        DOSPART  {nnnnnK}
                    {OFF   }
```

SET EXECTRAC

CMS

Specifies whether you want tracing turned ON or OFF for your System Product Interpreter or EXEC2 program.

```
SET        EXECTRAC {ON }
                    {OFF}
```

SET FILEPOOL

CMS

Sets the default file pool for the user.

```
SET FILEPool        [ filepoolid: ]
                    [ NONe        ]
                    [ PRImary     ]
```

SET FILEWAIT

CMS

Determines whether or not the user wants a request to wait for control of a user file.

```
SET FILEWait        { ON  }
                    { OFF }
```

SET FULLREAD

CMS

Lets 3270 null characters be recognized in the middle of the physical screen.

```
SET    FULLREAD  { ON  }
                 { OFF }
```

SET FULLSCREEN

CMS

Runs CMS in full-screen mode.

```
SET    FULLSCREEn  ┌ ON      ┐   [(options . . . [ )] ]
                   │ OFF     │
                   │ SUSPEND │
                   └ RESUME  ┘

       Options:  ┌ CLear   ┐
                 └ NOCLear ┘
```

A **special command** that can only be used in the CMS virtual screen in full-screen CMS, followed by its description, is:

```
#WM    wmcommand
```

Executes any of the following commands immediately from the CMS virtual screen:

- CLEAR WINDOW
- CP
- DROP WINDOW
- HIDE WINDOW
- MAXIMIZE WINDOW
- MINIMIZE WINDOW
- POP WINDOW
- POSITION WINDOW
- PUT SCREEN
- QUERY BORDER
- QUERY HIDE
- QUERY LOCATION
- QUERY RESERVED
- QUERY SHOW
- QUERY WINDOW
- QUERY WMPF
- RESTORE WINDOW
- SCROLL
- SET BORDER
- SET LOCATION
- SET RESERVED
- SET WINDOW
- SET WMPF
- SHOW WINDOW
- SIZE WINDOW

SET IMESCAPE

CMS

Indicates whether an escape character is required to execute immediate commands.

```
SET          IMESCAPE  ┌ ON   ┐
                       │ OFF  │
                       └ char ┘
```

SET IMPCP

CMS

Specifies whether command names unrecognized by CMS are considered CP commands and are passed on to CP.

$$\textbf{SET} \qquad \textbf{IMPCP} \begin{Bmatrix}\textbf{ON}\\ \textbf{OFF}\end{Bmatrix}$$

SET IMPEX

CMS

Controls whether exec files are treated as commands.

$$\textbf{SET} \qquad \textbf{IMPEX} \begin{Bmatrix}\textbf{ON}\\ \textbf{OFF}\end{Bmatrix}$$

SET INPUT

CMS

Controls the translation of a specified character *a* to hexadecimal code xx for characters entered from the terminal and the reset of the hexadecimal code xx to the specified hexadecimal code yy in your translate table.

$$\textbf{SET} \qquad \textbf{INPUT} \begin{bmatrix}a\ xx\\ xx\ yy\end{bmatrix}$$

SET INSTSEG

CMS

Specifies whether the system should search the Installation Saved Segment to locate an exec or editor macro.

```
SET     INSTSEG {ON    [fm | LAST ]}
                {OFF               }
```

SET KEYPROTECT

CMS

Resets the user keys.

```
SET        KEYPROTect {ON }
                      {OFF}
```

SET LANGUAGE

CMS

Changes the current language of your CMS session and any application running on CMS that uses National Language Support.

```
SET          LANGUAGE   [langid]   [(options ... [)]]

             Options:

              [ADD    applid  ]  [USER  ]  [TYPE   ]
              [DELETE  applid ]  [SYSTEM]  [NOTYPE ]
                                 [ALL   ]
```

SET LDRTBLS

CMS

Defines the number (nn) of pages of storage to be used for loader tables.

```
SET          LDRTBLS [nn]
```

SET LINEND

CMS

Activates and defines the logical line end for full-screen CMS.

```
SET     LINEND  {ON }  [char]
                {OFF}
```

SET LOCATION

CMS

Specifies whether the location indicator should be displayed in the window when the data in the virtual screen exceeds the size of the window.

```
SET     LOCATION    wname   {ON }
                            {OFF}
```

SET LOADAREA

CMS

Defines the ORIGIN default for the load process. (Only affects where TEXT files are to be loaded; does not influence the RMODE that may be propagated to the GENMOD process.)

```
SET   LOADAREA  { 20000   }
                { RESPECT }
```

SET LOGFILE

CMS

Indicates whether a log file should be updated with the data being written to the virtual screen.

```
SET      LOGFILE vname { ON  }  [fn  [LOGFILE  [fm  ]]]
                       { OFF }  [    [         [ *  ]]]
                                [    [         [ A1 ]]]
```

SET NONDISP

CMS

Defines a character used in place of nondisplayable characters.

```
SET         NONDISP  [char]
```

SET NONSHARE

CMS

Specifies a nonshared copy of a typical shared, named system.

```
SET        NONSHARE  ⎧CMSDOS ⎫
                     ⎨CMSVSAM⎬
                     ⎪CMSAMS ⎪
                     ⎩CMSBAM ⎭
```

SET OUTPUT

CMS

Controls the translation and reset of the specified hexadecimal representation xx to the specified character *a* for all xx characters displayed at the terminal.

```
SET      OUTPUT [xx  a]
```

SET PROTECT

CMS

Specifies whether the CMS nucleus is protected against writing in its storage area.

```
SET          PROTECT  ⎧ON ⎫
                      ⎩OFF⎭
```

SET RDYMSG **CMS**

Indicates whether the standard CMS ready message or a shortened form of the CMS ready message is used.

SET **RDYMSG** $\left\{\begin{matrix}\textbf{LMSG}\\ \textbf{SMSG}\end{matrix}\right\}$

SET REDTYPE **CMS**

Controls whether CMS error messages are typed in red for certain terminals equipped with the appropriate terminal feature and a two-color ribbon.

SET **REDTYPE** $\left\{\begin{matrix}\textbf{ON}\\ \textbf{OFF}\end{matrix}\right\}$

SET RELPAGE **CMS**

Releases or holds the page frames of storage and sets them to binary zeros, after the following commands complete execution: ASSEMBLE, COPYFILE, COMPARE, EDIT, MACLIB, SORT, TXTLIB, UPDATE, HELP, and the program product language processors supported by VM/SP.

SET **RELPAGE** $\left\{\begin{matrix}\textbf{ON}\\ \textbf{OFF}\end{matrix}\right\}$

SET REMOTE

CMS

Controls the display of data transmissions.

```
SET        REMOTE  {ON }
                   {OFF}
```

SET RESERVED

CMS

Specifies the number of lines in a window used to display virtual screen reserved lines.

```
SET        RESERVED   wname   {rtop} {rbot}
                              { *  } { *  }
```

SET SERVER

CMS

Enables private resource processing.

```
SET          SERVER {ON }
                    {OFF}
```

SET STORECLR

CMS

Sets point of automatic GETMAIN storage cleanup and determines the action for user invocation of STRINIT.

```
SET    STORECLR    ┌ENDCMD┐
                   └ENDSVC┘
```

SET SYSNAME

CMS

Allows for the replacement of a saved system name entry in the SYSNAMES table with the name of an alternative, or backup system.

```
SET          SYSNAME ┌CMSDOS  ┐
                     │CMSVSAM │  entryname
                     │CMSAMS  │
                     └CMSBAM  ┘
```

SET TEXT

CMS

Activates character code conversion of TEXT characters for windows.

```
SET          TEXT  ┌ON ┐
                   └OFF┘
```

SET THRESHOLD

CMS

Changes the warning threshold for the usage of space allocated to the user in the file pool.

```
SET      THReshold nn [filepoolid:]
```

SET TRANSLATE

CMS

Suppresses translations and translation synonyms of command names for a language.

```
SET      TRANslate  {ON }  [SYStem   [TRANslate  [ APPLID  applid ]]]
                    {OFF}  [USER     [SYNonym    [           *    ]]]
                           [ALL      [BOTH                         ]]
```

SET UPSI

CMS

Controls the setting of the UPSI (User Program Switch Indicator) byte to the specified bit string of 0's and 1's or to binary zeros.

```
SET          UPSI  {nnnnnnnn}
                   {OFF     }
```

SET VSCREEN

CMS

Indicates what action should take place when the virtual screen is updated with data.

$$\textbf{SET}\qquad \textbf{VSCREEN}\ \textit{vname} \left\{\begin{array}{lll} \begin{bmatrix}\textbf{TYPe}\\ \textbf{NOType}\end{bmatrix} & \begin{bmatrix}\textbf{PRotect}\\ \textbf{NOPRotect}\end{bmatrix} & \begin{bmatrix}\textbf{High}\\ \textbf{NOHigh}\end{bmatrix} \\ [\textit{color}] & [\textit{exthi}] & [\textit{psset}] \end{array}\right\}$$

SET WINDOW

CMS

Specifies whether the window is to be variable or fixed size.

$$\textbf{SET}\qquad \textbf{WINDOW}\ \textit{wname} \left\{ \begin{bmatrix}\textbf{VARiable}\\ \textbf{FIXed}\end{bmatrix} \begin{bmatrix}\textbf{POP}\\ \textbf{NOPop}\end{bmatrix} \begin{bmatrix}\textbf{TOP}\\ \textbf{NOTop}\end{bmatrix} \right\}$$

SET WMPF

CMS

Defines a WMPF key to execute a windowing command.

$$\textbf{SET}\qquad \textbf{WMPF}\ \textit{nn} \left[\begin{Bmatrix}\textit{pseudonym}\\ \textbf{NOWRITE}\end{Bmatrix} \begin{bmatrix}\textit{keyword}\\ \underline{\textbf{DELAYED}}\end{bmatrix}\ \textit{string} \right]$$

SET

CP Class A

Sets special CP preferred options.

```
SET   AFFInity  [userid ]  [ON  ]
                           [OFF ]
                           [nn  ]

      CPAssist             {ON }  [PROC [nn ]]
                           {OFF}

      FAVORed    userid    [nnn ]
                           [OFF ]

      JOurnal 1 {LOgon}    [ON  ]
                {LInk }    [OFF ]

      PRIORity   userid    nn

      QDROP   userid  {ON }   [USERS ] [NOQ3 ]
                      {OFF}

      REServe    userid    {nnnn}
                           {OFF }

      SASsist              {ON }  [PROC [nn ]]
                           {OFF}

      S370E  {ON } [[PROC ] [addr ]]
             {OFF}
```

---

[1] The JOURNAL operand is valid only if STQUERY=YES is specified in the SYSJRL macro instruction DMKSYS.

SET

CP Class B

Establishes disposition for log messages and dumps.

```
SET    ┌ DUMP     {AUTO }  [CP ]                                   ┐
       │          {raddr}  [ALL]                                   │
       │                                                           │
       │ LOGmsg   [nn         [text] ]                             │
       {          [NULL              ]                             }
       │                                                           │
       │ MITime   {class  {mm:ss}  [class {mm:ss}...] }            │
       │          {       {OFF  }  [      {OFF  }   ] }            │
       │          {                                   }            │
       └          {OFF                                }            ┘
```

SET

CP Class E

Sets SRM function and the number used in the working set size estimate control algorithm.

```
SET    ┌PAGing  nn                                ┐
       │SRM  ┌APAGES  nnnn                   ┐   │
       │     │DSPSlice  nnn                  │   │
       │     │IB  n                          │   │
       │     │MAXDrum     { nnnn }           │   │
       │     │            { OFF  }           │   │
       {     {MAXWss      { nnnn }           }   }
       │     │            { OFF  }           │   │
       │     │MHFULL      { nnn }            │   │
       │     │            { OFF }            │   │
       │     │PB  nn                         │   │
       │     │PCI         { DRUM }           │   │
       │     │            { DISK }           │   │
       │     │PGMTlim                        │   │
       └     └                               ┘   ┘
```

SET

CP Class F

Sets recording mode for a device, and enables or disables soft machine check interrupts.

```
SET    RECord { OFF                                                                    }
              { ON  raddr  LIMIT nn BYTE  nn BIT n [{ AND } BYTE nn BIT n]             }
              {                                    [{ OR  }               ]            }

       MODE   { RETRY } { Quiet  }    [cpuid ]
              { MAIN  } { Record }
```

SET

CP Class G

Controls various functions within your virtual machine.

```
SET      ┌ ACNT      {ON  }                                 ┐
         │           {OFF }                                 │
         │ AFFInity  [ON  ]                                 │
         │           [OFF ]                                 │
         │ ASsist    {[ON] ] [SVC   ] [TMR  ]}              │
         │           {[OFF]] [NOSVC ] [NOTMR]}              │
         │ AUTOPoll  {ON  }                                 │
         │           {OFF }                                 │
         │ CONCeal   {ON  }                                 │
         │           {OFF }                                 │
         │ CPCONIO   {OFF }                                 │
         │           {IUCV}                                 │
         │ CPUid     bbbbbb                                 │
         │ ECmode    {ON  }                                 │
         │           {OFF }                                 │
         │ EMSG      {ON  }                                 │
        {            {OFF }                                  }
         │           {CODE}                                 │
         │           {TEXT}                                 │
         │           {IUCV}                                 │
         │ IMSG      {ON  }                                 │
         │           {OFF }                                 │
         │           {IUCV}                                 │
         │           {ON  }                                 │
         │ ISAM      {OFF }                                 │
         │ LINEDit   {ON  }                                 │
         │           {OFF }                                 │
         │ MIH       {ON  }                                 │
         │           {OFF }                                 │
         │ MSG       {ON  }                                 │
         │           {OFF }                                 │
         │           {IUCV}                                 │
         │ NOTRans   {ON  }                                 │
         │           {OFF }                                 │
         │ PAGEX     {ON  }                                 │
         │           {OFF }                                 │
         └   (format continued on the next page)            ┘
```

(format continued from previous page)

```
 PFnn     [ COPY  (resid)  ]
 PFnn     [ COPY  (cuu)    ]
 PFnn     [ COPY  (luname) ]
 PFnn     [ COPY  (Laddr)  ]

 PFnn     [ IMMed   ] [pfdata1 #pfdata2 #...pfdatan ]
          [ DELayed ]
            -------

 PFnn   RETrieve   [ backward ]
                     --------
                   [ FORward  ]

 PFnn   [TAB  n1  n2  ... ]

 RUN              { ON  }
                  { OFF }

 SMsg             { ON   }
                  { OFF  }
                  { IUCV }

 STBypass         { [nnnnn K  [ NOVERIFY ]] }
                  { [nn M                 ] }
                  {                         }
                  {   VR                    }
                  {   OFF                   }

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

 SVCAccl          { ON  }
                  { OFF }

 TIMER            { ON   }
                  { OFF  }
                  { REAL }

 VMCONIO          { OFF  }
                  { IUCV }

 VMSAVE           { ON   }
                  { OFF  }
                  { name }

 WNG              { ON   }
                  { OFF  }
                  { IUCV }

 370E             { ON  }
                  { OFF }
```

SET **GCS**

Replaces a saved system name entry for VSAM in the SYSNAMES table.

```
SET      SYSNAME  {GCSVSAM}  entry name
                  {GCSBAM }
```

SET **RSCS**

Requests or disables console message routing (for authorized alternative operator only).

```
SET        {linkid}  {MSG  }
           {*     }  {NOMsg}
```

SET ETRACE **TSAF**

Enables or disables external tracing. Only TSAF virtual console or the secondary user of the TSAF virtual machine can issue this command.

```
SET      ETRACE {ON }
                {OFF}
```

SETKEY

CMS

Assigns storage protect keys to storage assigned to named systems.

```
SETKEY    key  systemname  [startadr]
```

SETPRT

CMS

Loads a virtual 3800 printer. Command is valid only for the 3800.

```
SETPRT        ┌                           ┐
              │ Chars   [(]cccc... [)]    │
              │ COpies  [(]nnn[)]         │
              │ COPYnr  [(]nnn[)]         │
              │ Fcb     [(]ffff [)]       │
              │ FLash   [(]id nnn[)]      │
              │ Init                      │
              │ Modify  [(]nmmm [n][)]    │
              └                           ┘
```

SHOW WINDOW

CMS

Places a window on top of all other displayed windows and connects a window to a virtual screen.

```
SHOW        WINdow  wname    [ON  vname    [line  col]]
```

SHUTDOWN

CP Class A

Checkpoints and terminates the current VM/370 operation.

```
SHUTDOWN   ┌REIPL [raddr]┐
           └POWEROFF     ┘
```

SHUTDOWN

CMS

Stops RSCS operations in an orderly fashion. Issues DRAIN to all active links, unless faster termination is requested by the QUICK command. Deactivates the RSCS/VTAM interface if active. (For RSCS operator only).

```
SHUTDOWN   [QUICK ]
```

SIZE WINDOW

CMS

Changes the number of lines and columns for a specified window.

```
SIZE WINdow     {wname}  lines    [cols]
                {  =  }
```

SLEEP

CP Class Any

Places the virtual machine in a dormant state with the terminal keyboard entry blocked. Allows message display.

```
SLeep      [ nn   [ SEC ] ]
                  [ MIN ]
                  [ HRs ]
```

SMSG

CP Class G

Sends a special message to a virtual machine that is running with SET SMSG ON.

```
SMsg             userid  msgtext
```

SMSG

RSCS

Delivers the command text to the RSCS virtual machine to be executed. All RSCS commands entered by a virtual machine user (including authorized alternative operators) must be included as text in an SMSG command. (The exception is when a local installation has provided an exec for each command that automatically puts the *SMSG rscsid* characters in front of the RSCS command expression).

```
                         { command-text                        }
SMsg         rscsid      { CMD    nodeidc    [command-text]    }
                         { Msg    nodeidm    userid   message-text }
```

SNTMAP

CMS

Processes DMKSNT macro definitions and produces a saved segment DASD map and a virtual memory map.

```
SNTMAP  ┌ fn      ┌ ft        ┌ fm ┐ ┐ ┐
        │         └ ASSEMBLE  └ *  ┘ ┘ │
        │                              │
        │ ┌ HELP ┐                     │
        └ └ ?    ┘                     ┘
```

SO

CMS

Suspends the recording of trace information during the execution command or program.

```
SO
```

SORT

CMS

Arranges a specified file in ascending order according to sort fields in the data records.

```
SORT     fileid1    fileid2
```

SPACE

CMS

Forces single spacing on the printer.

```
SPAce      { raddr }
           { lprt  }
```

SPLOAD

CMS

Loads the VM/SP product tapes to the appropriate minidisks during initial installation.

```
SPLOAD   group   element   [ fn  [ ft ] ]
                           [ *   [ *  ] ]
                             -     -
```

SPMODE

CP Class A

Establishes or resets the single processor mode.

```
SPMode     { ON  }
           { OFF }
```

SPOOL

CP Class G

Changes spooling control options.

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

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

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

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

[3] These options can only be used to modify a virtual spooling printer. These options apply only to a device type 3800 as a virtual spooling device.

SPTAPE

CP Class D

Dumps output spool files to tape or loads output spool files from tape.

```
SPTape  ┌ STOP raddr                                                                ┐
        │ CANCEL raddr                                                              │
        │ SCAN raddr SADump option2                                                 │
        │ LOAD raddr SADump option2 option5                                         │
        │ LOAD raddr {Printer} {spoolid1 [spoolid2]                       } option2 │
        │            {PUnch  } {         [END     ]                       } option3 │
        │            {Reader } {                                          } option5 │
        │                      {CLass c1 [c2 [c3 [c4]]] [FORM form][DEST dest]}     │
        │                      {FORM form [DEST dest]                     }         │
        │                      {DEST dest                                 }         │
        │                      {ALL                                       }         │
        │                                                                           │
       {  DUMP raddr {Printer} {spoolid1 [spoolid2]                       } option1 }
        │            {PUnch  } {         [END     ]                       } option2 │
        │            {Reader } {                                          } option3 │
        │                      {CLass c1 [c2 [c3 [c4]]] [FORM form][DEST dest]} option4
        │                      {FORM form [DEST dest]                     }         │
        │                      {DEST dest                                 }         │
        │                      {ALL                                       }         │
        │                                                                           │
        │                         ----options----                                   │
        │  option 1           option 2     option 3     option 4     option5        │
        │ [MODE { 800}]       [LEAVE ]     [SYSHOLD ]   [PURGE]      [FOR userid]   │
        │ [     {1600}]       [REWind]     [USERHOLD]                               │
        │ [     {6250}]       [RUN   ]     [NOHOLD  ]                               │
        │ [     { 38K}]                                                             │
        └                                                                           ┘
```

SSERV

CMS

Copies a VSE source statement book onto a CMS minidisk or SFS directory, displays it at the terminal, or spools it to the virtual punch or printer.

```
SSERV      sublib bookname [ft  ]     [ (options... [)]]
                           [COPY]

           Options:   [DISK]  [PRINT] [PUNCH]  [TERM]
```

START

CP Class D

Restarts a drained device or changes its output spooling class.

```
STArt   [ALL                                                                                   ]
        [                                                                                      ]
        [Printer                                                                               ]
        [PUnch                                                                                 ]
        [Reader                                                                                ]
        [{raddr}[CLass c...][FORM form][DEST dest][NOSep][AUto   ][NO3800 ]                    ]
        [{lprt }            [FORM *   ][DEST OFF ]       [SETup  ][BEG3800]                    ]
        [                              [DEST *   ]       [MAnual ][ANY3800]                    ]
        [                                                                                      ]
        [                   [FLash  name    ]      [DEFfcb  ]                                  ]
        [                   [CHars  name    ]      [FILefcb ]                                  ]
        [                   [FCB    plpi    ]      [CFIlefcb]                                  ]
        [                   [IMage  imagelib]                                                  ]
        [                   [PUrge          ]                                                  ]
```

START

CMS

Begins execution of programs previously loaded (OS and CMS) or fetched (CMS/DOS).

```
START     [ entry [args...] ]
          [ *               ]
          [ (option [)]     ]
```

```
          Option:   [NO]
```

START

RSCS

Activates a specified communication link.

```
STArt   [linkid]  [CLass c]
                  [DP dpriority]
                  [FOrm name]
                  [LINE vaddr]
                  [LOGMode logmodename]
                  [LUName luname]

                  [{ MAnual }]
                  [{ AUto   }]
                  [{ SETup  }]

                  [        { Priority }]
                  [ Queue  { Fifo     }]
                  [        { Size     }]

                  [{ RETry   }]
                  [{ NORETry }]

                  [ TRace { ALL }]
                  [       { LOG }]

                  [       { ASCII     }]
                  [       { LISTPROC  }]
                  [       { MRJE      }]
                  [ TYPE  { NJE       }]
                  [       { RJE       }]
                  [       { SNANJE    }]
                  [       { SNARJE    }]
                  [       { SNA3270P  }]
                  [       { 3270P     }]

                  [{ OParm operation parameters ...   }]
                  [{ Parm  [operation parameters ...] }]
```

Note: Any combination of keywords with associated options may be entered in any order, except that the Parm keyword must be the rightmost keyword.

STAT

IPCS

Lists current status (as found in the SYMPTOM SUMMARY file) of a problem, a specific subset of problems, or all problems.

```
STAT    ┌ nnnnn                                          ┐
        │                                                │
        │ ALL  ┌ {OPENUSER } ┐1┌ ABend            ┐2     │
        │      │ {OPNUSR   } │ │                  │      │
        │      │             │ │ {DOC}            │      │
        │      │ {OPENIBM }  │ │ {DD }            │      │
        │      │ {OPNIBM  }  │ │                  │      │
        │      │             │ │ INcorr           │      │
        │      │ {OPEN}      │ │                  │      │
        │      │ {OPN }      │ │ INF              │      │
        │      │             │ │                  │      │
        │      │  APARED     │ │ {LOOP}           │      │
        │      │             │ │ {LP  }           │      │
        {      │ {NEEDINFO}  │ │                  │      }
        │      │ {NDINFO  }  │ │ MSg              │      │
        │      │             │ │                  │      │
        │      │ {PTFRCVD}   │ │ {PERFORM}        │      │
        │      │ {PTFRCV }   │ │ {PR     }        │      │
        │      │             │ │                  │      │
        │      │  PTFON      │ │ {WAIT}           │      │
        │      │             │ │ {WS  }           │      │
        │      │             │ │                  │      │
        │      │             │ │ {PTFERROR}       │      │
        │      └  CLOSED     ┘ └ {PE      }       ┘      │
        │                                                │
        └ HELP                                           ┘
```

[1] One of these status keywords may be specified with the ALL operand.

[2] One of these failure keywords may be specified with the ALL operand.

STATE/STATEW

CMS

STATE verifies the existence of a CMS file on a minidisk or in an SFS file pool. STATEW verifies the existence of a file on a read/write file mode.

```
STATE       { fn }  { ft }  [ fm ]
STATEW      {  * }  {  * }  [  * ]
ESTATE                      [  - ]
ESTATEW
```

STCP

CP Class C

Alters contents of real storage. The real PSW or registers cannot be altered. Shared pages in a system running in AP mode cannot be altered.

```
STCP      ┌ {ML hexloc}                                  ┐
          │ {NL hexloc}                                  │
          │ { M hexloc}      hexword1     [hexword2 ...] │
          │ { N hexloc}                                  │
          │ { L hexloc}                                  │
          { {   hexloc}                                  }
          │                                              │
          │ {MS hexloc}                                  │
          │ {NS hexloc}      hexdata                     │
          └ { S hexloc}                                  ┘
```

STOP **CMS**

Stops multiple user mode processing for a file pool. (File pool server operator only.)

```
STOP  ┌ BACKUP    ┐
      │ NOBACKUP  │
      └ IMMEDIATE ┘
```

STOP **RSCS**

Quickly deactivates a specified link without completing transmission of a file.

```
STOP        [ linkid ]
```

STOP TSAF **TSAF**

Stops the TSAF virtual machine. Only the TSAF virtual console or the secondary user of the TSAF virtual machine can enter this command.

```
STOP       TSAF
```

STORE

CP Class G

Alters virtual machine storage, PSW, and registers.

```
STore     ┌                                                        ┐
          │  hexloc                                                │
          │  L hexloc       hexword1      [hexword2  ... ]         │
          │                                                        │
          │  S hexloc       hexdata...                             │
          │                                                        │
          │ {G reg }                                               │
         <  {X reg }        hexword1      [hexword2  ... ]          >
          │                                                        │
          │ {Y reg }        hexword1      [hexword2  ... ]         │
          │                                                        │
          │  Psw          [hexword1 ]     hexword2                 │
          │                                                        │
          │  STATUS                                                │
          └                                                        ┘
```

SVCTRACE

CMS

Records information about supervisor calls.

```
SVCTrace    {ON }
            {OFF}
```

SYNONYM

CMS

Uses a table containing synonyms you have created for CMS and user-written commands.

```
SYNonym     [fn [SYNONYM [fm]]]  [ (options...[)]]
                         [A1]
                         [* ]

                Options:   [STD  ]   [CLEAR]
                           [NOSTD]
```

SYSTEM

CP Class G

Simulates virtual machine console functions and clears virtual storage and storage keys to binary zeros.

```
SYStem      {CLEAR  }
            {RESET  }
            {RESTART}
```

TAG

CP Class G

Appends or queries the TAG text to a VM/SP spool file utilized by subsystems (such as RSCS).

```
TAg
                 { Printer }
                 { PUnch   }
         DEv     { CONsole }   [tagtext]
                 { vaddr   }

         FIle    spoolid       [tagtext]

                 {         { Printer } }
                 {         { PUnch   } }
                 { DEv     { CONsole } }
         QUery   {         { vaddr   } }
                 {                     }
                 { FIle    spoolid     }
```

TAPE

CMS

Performs tape-to-directory or minidisk and directory- or minidisk-to-tape operations for CMS files, positions tapes, and displays or writes VOL1 labels.

```
TAPE   ┌ DUMP  {fn}{ft}[fm]           [(optionA optionB optionD optionF[)]] ┐
       │       {* }{* }[* ]                                                 │
       │ LOAD  [{fn}{ft}[fm]]         [(optionB optionC optionD[)]]         │
       │       [{* }{* }[A ]]                                               │
       │ SCAN  [{fn} {ft}]            [(optionB optionC optionD[)]]         │
       │       [{* } {* }]                                                  │
      {  SKIP  [{fn}{ft}]             [(optionB optionC optionD[)]]          }
       │       [{* }{* }]                                                   │
       │ DVOL1                        [(optionD optionE[)]]                 │
       │ WVOL1 volid {owner}          [(optionD optionE[)]]                 │
       │ MODESET                      [(optionD [)]]                        │
       │ tapcmd [n]                   [(optionD [)]]                        │
       └        [1]                                                         ┘

OptionA:  [WTM  ]  [BLKsize 800 ]          OptionB:  [NOPRint]
          [NOWTM]  [BLKsize 4096]                    [PRint  ]
                   [BLKsize 4K  ]                    [Term   ]
                   [BLKsize 32K ]                    [DISK   ]
                   [BLKsize 64K ]

OptionC:  [EOT  ]    OptionD:  [[TAPn]] [7TRACK ]  [DEN den] [TRTCH a]
          [EOF n]              [[TAP1]] [9TRACK ]
          [EOF 1]              [        [18TRACK]
                               [[vdev]]
                               [[181 ]]

OptionE:  [REWIND]          OptionF:  [TRANsfer BUFF ]
          [LEAVE ]                    [TRANsfer IMMED]
```

TAPEMAC

CMS

Creates CMS MACLIB libraries directly from an IEHMOVE-created partitioned data set on tape.

```
TAPEMAC   fn [SL   [labeldefid]                   ] [(options... [)]]
             [NSL   filename  [ID = identifier]   ]

          Options:     [TAPn]   [ITEMCT yyyyy]
                       [TAP1]   [ITEMCT 50000]
```

TAPPDS

CMS

Loads OS partitioned data set (PDS) files or card image files from tape to minidisk or SFS directory.

```
TAPPDS   [fn [ft [fm ]]] [SL  [labeldefid                 ]  [(options...[)]]
         [*  [*  [A1 ]]] [NSL filename  [ID = identifier] ]
                 [*  ]

         Options:  [PDS   ] [COL1  ] [TAPn] [END  ] [MAXTEN  ]
                   [NOPDS ] [NOCOL1] [TAP1] [NOEND] [NOMAXTEN]
                   [UPDATE]
```

TE

CMS

Stops all tracing of your System Product Interpreter or EXEC 2 program or macro.

```
TE
```

TELL

CMS

Sends a message to one or more computer users who are logged on to your computer or to one attached to yours by way of RSCS.

```
TELL          name    message
```

TERMINAL

CP (Class G)

Controls virtual console functions.

```
TERMinal  ┌ CHardel     ┌ ON  ┐        ┐ 1
          │ LINEDel     │ OFf │        │
          │ LINENd      └ char┘        │
          │ EScape                     │
          │ TABchar                    │
          │                            │
          │ APL         ┌ ON  ┐        │
          │ TEXT        └ OFf ┘        │
          │ ATtn                       │
          │ HIlight                    │
          │ SCRNsave                   │
          │                            │
          │ MODE        ┌ CP  ┐        │
          │             └ VM  ┘        │
          │                            │
          │ LINESize    ┌ nnn ┐        │
          │             └ OFF ┘        │
          │                            │
          │ CONmode     ┌ 3215 ┐       │
          │             └ 3270 ┘       │
          │                            │
          │ BREakin     ┌ IMmed    ┐   │
          │             └ GUESTctl ┘   │
          │                            │
          │ BRKkey      ┌ PA1  ┐       │
          │             │ PF1  │       │
          │             │  :   │       │
          │             │ PF24 │       │
          │             └ NONE ┘       │
          │                            │
          │ TYpe        ┌ 3101 ┐       │
          │             └ TTY  ┘       │
          │                            │
          │ PROMpt      ┌ VM  ┐        │
          │             └ TTY ┘        │
          │                            │
          │ SCROll      ┌ nnn  ┐       │
          │             └ CONT ┘       │
          │                            │
          │ CNTL        ┌ USR ┐        │
          │             └ SYS ┘        │
          │                            │
          │ ASCiitbl    ┌ VM2 ┐        │
          └             └ VM1 ┘        ┘
```

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

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

TRACE

CP Class G

Traces and records program execution.

```
TRace   ┌ ┌SVC      ┐ 1  ┌PRINter                     ┐┐
        │ │I/O      │    │                            ││
        │ │PROgram  │    │ ┌TERMinal┐ ┌NORun┐        ││
        │ │EXTernal │    │ └BOTH    ┘ └RUN  ┘        ││
        │ │PRIV     │    │                            ││
        │ │SIO      │    │                            ││
        { │CCW      }    │  OFf                       │}
        │ │SNS      │    └                            ┘│
        │ │BRanch   │                                  │
        │ │INSTruct │                                  │
        │ │ALL      │                                  │
        │ └CSW      ┘                                  │
        │  END                                         │
        └                                              ┘
```

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

`TRACE SVC PROGRAM SIO PRINTER.`

TRACE

RSCS

Monitors line activity on a specified link.

```
                   ┌OFF  ┐
TRace   [linkid ]  │ALL  │  [ TO  userid   [nodeid ]]
                   │LOG  │
                   └NOLog┘
```

TRANSFER

CP Class D

Transfers closed reader spool files.

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

TRANSFER

CP Class G

Transfers closed reader spool files.

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

TRANSFER

RSCS

Changes the destination address for specified files.

General User Format:

```
TRANsfer    [*] spoolid    TO    nodeid [ userid ]
```

Operator Format:

```
TRANsfer    [linkid] spoolid [spoolid... ] TO nodeid [userid ]
```

TRAPFILE

IPCS

Lists, dumps or prints the set of CPTRAP files for a specific problem number.

```
TRAPFILE  [LIST   [nnnnn]                   ]
          [DUMP   [nnnnn    [tape]]         ]
          [PRINT  [nnnnn    [options...]]   ]
          [HELP                             ]

          options:

               [SUMMARY] [ENTRY  ] [PROMPT  ]
               [NOSUM  ] [NOENTRY] [NOPROMPT]

               [HEX   ]
               [FORMAT]
```

TS

CMS

Starts tracing your System Product Interpreter or EXEC 2 program or macro.

```
TS
```

TXTLIB

CMS

Generates and modifies text libraries.

```
TXTlib    { GEN   libname fn1 [fn2... ] [(optionA [) ]]  }
          { ADD   libname fn1 [fn2... ] [(optionA [) ]]  }
          { DEL   libname membername1 [membername2... ]  }
          { MAP   libname [(optionB [)]]                 }

          OptionA: [FILename ]    OptionB: [ TERM  ]
                                           [ DISK  ]
                                           [ PRINT ]
```

TYPE

CMS

Displays all or part of a CMS file at a terminal.

```
Type      fn ft [fm]   [rec1 [recn]]   [(options...])]]
                [ * ]  [ *   [ *  ]]
                       [ 1         ]
        options:
        [HEX] [COL {xxxxx} - [yyyyy]] [MEMber {  *   }]
                   {    1}   [lrecl]          { name }
```

UNLOCK

CP Class A

Releases pages of storage.

```
UNLOCK   ┌┌ userid ┐            firstpage lastpage┐
         │└ SYSTEM ┘                              │
         {                                        }
         │┌ VIRT=REAL ┐                           │
         └└ V=R       ┘                           ┘
```

UPDATE

CMS

Makes changes in a program source file as defined by control cards in a control file.

```
Update     fn1  [ft1       [fm1 [fn2 [ft2  [fm2]]]]] [( options...[ ) ] ]
                 ASSEMBLE   A1

           Options:

           [REP  ] [SEQ8  ] [INC  ] [CTL  ]  [OUTMODE fm]
           [NOREP] [NOSEQ8] [NOINC] [NOCTL]

           [STK  ] [TERM  ] [DISK ] [STOR  ]
           [NOSTK] [NOTERM] [PRINT] [NOSTOR]
```

Control Statements:

./S [*seqstrt* [*seqincr* [*label*]]]

Resequences the updated source output file.

./I *seqno* [$ [*seqstrt* [*seqincr*]]]

Inserts all records following it, up to the next control statement, into the output file.

```
Update     fn1  [ft1     ]  [fm1 [fn2 [ft2  [fm2]]]]][( options...[ ) ]]
                [ASSEMBLE]  [A1
```

Options:

```
[REP  ]  [SEQ8  ]  [INC  ]  [CTL  ]   [OUTMODE fm]
[NOREP]  [NOSEQ8]  [NOINC]  [NOCTL]

[STK  ]  [TERM  ]  [DISK ]  [STOR  ]
[NOSTK]  [NOTERM]  [PRINT]  [NOSTOR]
```

Deletes one or more records from the source file.

```
./R  seqno1  [seqno2] [$  [seqstrt [seqincr]]]
```

Replaces one or more input records with updated records from the update file.

```
./*  [comment]
```

Allows the insertion of comments.

UTILITY

CMS

Provides installation functions such as printing system definition files, creating stand-alone service utility tape and service programs on disk, etc.

```
UTILITY    { DTYPE     vdev                        }
           { IPLDECK   { ALL }                     }
           {           { DDR } [ctlfile]           }
           {           { DIR }                     }
           {           { FMT }                     }
           { NUCTAPE   [ CP     ]                  }
           {           [ CPHPO  ]                  }
           { PRSAMPLE                              }
           { UTILTAPE  { ALL }                     }
           {           { DDR }                     }
           {           { DIR }                     }
           {           { DSF }                     }
           {           { FMT }                     }
```

VALIDATE

CMS

Verifies the syntax of a file identifier and verifies whether or not a specified file mode is accessed.

```
VALIDATE   { fn }  { ft }  [ fm ]
           { *  }  { *  }  [ *  ]
```

VARY

CP Class B

Varies the availability of a device.

```
VARY        {ONLine }  {raddr-raddr          }
            {OFfline}  {raddr ... [lprt]     }
                       {lprt ... [raddr]     }
                       {PROCessr             }

                OFfline  PROCessr  nn [VPHY]   [FORCE]
                                      [VLOG]
```

(VPHY is underlined as the default.)

VMDUMP

CP Class G

Dumps storage for virtual machine. It also enables the sending of dumps to other users; used in conjunction with VM/IPCS.

```
VMDump      [hexloc1]   [ {-}  [hexloc2]   ]
            [   0   ]   [ {:}  [END    ]   ]
            [       ]   [                  ]
            [       ]   [ {.}  [bytecount] ]
            [       ]   [      [END      ] ]

            [TO *      ]
            [TO userid ]
            [SYSTEM    ]

            [FORMAT  vmtype ]

            [DSS]

            [*dumpid ]
```

(0, END and TO * are underlined as defaults.)

VMFAPPLY

CMS

Creates and/or updates auxiliary control files for the PTFs on the service tape.

```
VMFAPPLY   ppfname [compname [updateid]] [(options...[)]]

           options:

           [EXCLUDE  ] [CHECK  ] [LOG  ]
           [NOEXCLUDE] [NOCHECK] [NOLOG]

           [SETUP  ] [PUT]
           [NOSETUP] [COR]
```

VMFASM

CMS

Creates an updated source file using IBM updates, PTFs, and user updates, then assembles the source file.

```
VMFASM   fn{ ctlfile            } [(options...)]
           {ppfname[compname]}

         Options:

         [PPF]    [SETUP  ]
         [CTL]    [NOSETUP]

         Assembler Options:

         [DISK ]  [TERM  ] [LIST  ]
         [PRINT]  [NOTERM] [NOLIST]

         [DECK  ] [RENT  ]  [EXP] [XREF]  [MAX]
         [NODECK] [NORENT]                [MIN]
                                          [STD]
```

VMFBLD

CMS

Copies and renames PTF numbered text decks, applies patches, builds objects (nuclei, SFS load modules, RUNTSAF MODULE).

```
VMFBLD   {ppfname [compname [bldlist]]                  }[(options[)]]
         {prodid [compname][product-exec-parameters]}

         options:

         [CHECK  ] [LOG  ] [IPL  ] [SETUP  ]
         [NOCHECK] [NOLOG] [NOIPL] [NOSETUP]
```

VMFDOS

CMS

Creates CMS files for VSE modules for specific installation purposes. Uses either a VSE distribution library or SYSIN tape.

```
VMFDOS   [LOAD]  [ [181]  ]  [PRIVate]  [fn]  [(options ...) ]
         [SCAN]  [ [182]  ]  [SYSTem ]
                 [        ]  [SYSIN  ]
                 [ [TAP1] ]
                 [ [TAP2] ]

         PRIV or SYST options:

              CSL  {*          }
                   {xxx*       }
                   {module name}

              RL   {*          }
                   {yyy*       }
                   {module name}

              SL   {*          }
                   {zzz*       }
                   {module name}

         SYSIN options:

              [ALL   ]
              [SELECT]
```

VMFHASM

CMS

Updates an ASSEMBLE source file according to entries in a control file, then assembles the updated file with the H-Assembler to produce an object file.

```
VMFHASM   fn{ ctlfile                   } [(options...)]
            {ppfname [compname]}

          Options:

          [PPF]     [SETUP   ]
          [CTL]     [NOSETUP ]

          Assembler Options:

          [DISK ]   [TERM   ] [LIST  ]
          [PRINT]   [NOTERM ] [NOLIST]

          [DECK  ] [RENT  ]   [EXP] [XREF]  [MAX]
          [NODECK] [NORENT]                 [MIN]
                                            [STD]
```

VMFLKED

CMS

Calls the CMS LKED command to link-edit modules into a LOADLIB.

```
VMFLKED     fn [ft          [fm ]]   [(options...[)]]
               [LKEDCTRL    [ * ]]
```

VMFLOAD

CMS

Generates a new CP, CMS or RSCS module.

```
VMFLOAD   loadlist ctlfile [langid] [(PTF [)]]
```

VMFMAC

CMS

Updates macro libraries using IBM and user updates.

```
VMFMAC      libname   [ctlfile]
```

VMFMERGE

CMS

Applies PTFs to object code and maintains a record in the Merge log.

```
VMFMERGE     prodid  { PTF {ptfnum}       } [EXCLUDE  exclist]
                     {     {*     }       }
                     {                    }
                     { PTFLIST  applist   }
```

VMFNLS

CMS

Applies updates to national language source files and compiles the updated versions.

```
VMFNLS       fn   ft   ctlfile    [(options ... [)]]
```

VMFOVER

CMS

Creates a temporary product parameter file containing parameters for a single component and applies component parameter overrides.

```
VMFOVER      [ppfname  compname]
```

VMFPLC2

CMS

Loads source code from product tape, dumps CMS-formatted files from disk to tape, loads previously dumped files from tape to disk, performs various control operations on a specified tape drive, and loads the service installation VMSERV EXEC from the PUT. (VMFPLC2 does not process multi-volume files. Files must also be CMS-formatted.)

```
VMFPLC2  ┌ DUMP     {fn}   {ft}   [fm]   [(optionA   optionB]
         │          { * }  { * }  [ * ]  [ optionD)         ]
         │ LOAD    [{fn}   {ft}   [fm]   [(optionB   optionC]
         │         [{ * }  { * }  [ A ]  [ optionD   optionE]
         │                               [ optionF)         ]
         │ SCAN    [{fn}   {ft}]         [(optionB   optionC ]
        <          [{ * }  { * }]        [ optionD   optionF)]
         │ SKIP    [{fn}   {ft}]         [(optionB   optionC ]
         │         [{ * }  { * }]        [ optionD)          ]
         │ MODESET                       [(optionD)]
         │ tapcmd   [n]                  [(optionD)]
         └          [1]

      optionA:  [WTM  ]  [BLKsize 4000]
                [NOWTM]  [BLKsize 32K ]
                         [BLKsize 64K ]

      optionB:  [NOPRint]
                [PRint  ]
                [Term   ]
                [DISK   ]
                [APPend ]

      optionC:  [EOT    ]
                [EOF n  ]
                [EOF 1  ]

      optionD:  [[TAPn]]  [ 7TRACK]  [DEN den] [TRTCH a]
                [[TAP1]]  [ 9TRACK]
                [[vdev]]  [18TRACK]
                [[181 ]]

      optionE:  [SELect] [STOP]

      optionF:  [DATE]
```

VMFREC

CMS

Receives program update service or corrective service from tape.

```
VMFREC  { INFO                                                                 }
        { ppfname [compname] [(options [)]]                                    }
        { prodid [compname] [product-exec-parameters] [(option [)]]            }
        { LIST fn ft                                                           }

        options:

        [LOG  ]  [SETUP  ]  [PUT]
        [NOLOG]  [NOSETUP]  [COR]
```

VMFREMOV

CMS

Removes PTFs applied using VMFMERGE.

```
VMFREMOV    prodid   [ PTF {ptfnum}                ]
                     [     {  *   }                ]
                     [                             ]
                     [ PTFLIST remlist             ]
                     [ CONVERT [lastfilemode]      ]
```

VMFSETUP

CMS

Defines a minidisk and SFS directory access order.

```
VMFSETUP ppfname[compname [PPFTEMP]] [(options...[)]]

        options:

        ┌[ACCESS] ┌ASM┐┐ [RESTORE]
        │         │REC││
        │         │APP││
        │         │BLD││
        └         └ALL┘┘
```

VMFTXT

CMS

Creates text libraries using IBM and user updates.

```
VMFTXT      libname   [ctlfile]
```

VMFZAP

CMS

Applies ZAPs to object code and maintains a record of them in the ZAP Log.

```
ZAP      ┌MODULE ┐
         │LOADLIB│ [libname1 ... libname3] [(option ...[)]]
         └TXTLIB ┘

      Options:
         ┌TERM          ┐  ┌PRINT  ┐
         └INPUT filename┘  └NOPRINT┘
```

VRSIZE

CMS

Builds a DMKSLC TEXT file used to generate a virtual = real (V = R) area when building the CP nucleus.

VRSIZE

VSAMGEN

CMS

Builds the CMSVSAM and CMSAMS physical saved segments.

VSAMGEN

VSEVSAM

CMS

Builds a VSE/VSAM maclib containing the supported VSE/VSAM macros as well as the following VSE macros: CDLOAD, CLOSE, CLOSER, GET, OPEN, OPENR, AND PUT.

VSEVSAM

WAITREAD VSCREEN

CMS

Used from an exec to update the virtual screen with data, refresh the physical screen, and wait for the next attention interrupt.

WAITREAD VSCreen *vname*

WAITT VSCREEN

CMS

Updates the virtual screen with data.

WAITT VSCreen $\left[\begin{array}{c} \mathit{vname} \\ \underline{*} \end{array}\right]$

WARNING

CP Class A and B

Sends high-priority messages.

Warning
Wng $\left\{\begin{array}{l} \mathit{userid} \\ \mathbf{OPerator} \\ \mathbf{ALL} \end{array}\right\}$ *msgtext*

WRITE VSCREEN

CMS

Enters information in a virtual screen.

```
WRITE VSCreen  vname line col length  [ ( [REServed ]  [optionA] [optionB ]
                                          [ optionC ][optionD ][) ]]

     OptionA:  [ BLANKs ]
               [ NULls  ]

     OptionB:  [ PROtect   ]   [ High      ]
               [ NOPROtect ]   [ NOHigh    ]
                               [ Invisible ]

     OptionC:  [color] [exthi ][psset ]

     OptionD:  { FIELD }
               { DATA  }
               { COLOR }   text
               { EXTHI }
               { PSS   }
```

Note: If option D is used, a right parenthesis should not be used to mark the end of the options.

X

CMS Border Command

Maximizes the window.

```
X
```

XEDIT **CMS**

Uses the VM/SP System Product Editor to create or modify a file in an SFS directory or on a minidisk. XEDIT subcommands, the System Product Interpreter, and the EXEC 2 macro facility can all be used from within XEDIT.

Note: In all formats of the XEDIT subcommands and macros, use of the word "subcommand" means an XEDIT subcommand only.

```
Xedit      [fn [ft [fm]]]     [(options...[)]]
           Options:
           [WINdow wname]  [Width nn] [NOSCreen]
           [PROFile macroname] [NOPROFil] [NOCLear]
           [NOMsg][MEMber membername]
           ┌LOCk    ┐
           └NOLOCk  ┘

           Options Valid Only in Update Mode:
             ┌Update  ┐  ┌Seq8  ┐  ┌Ctl fn1┐
             └NOUpdate┘  └NOSeq8┘  └NOCtl  ┘

             [Merge]  [UNtil filetype]   [Incr nn]
             [SIDcode string]
```

XEDIT subcommands and macros:

```
&               [subcommand]
```

Redisplays the subcommand and allows reexecution by pressing the ENTER key.

?

Displays the last executed XEDIT subcommand except for the = (equal sign) or the ? (question mark) subcommands.

= [*subcommand*]

Reexecutes the last subcommand or macro that was entered. Also executes a specified subcommand and *then* reexecutes the last one entered.

Add [*n*| 1]

Inserts blank lines immediately following the current line.

ALL [*rtarget*]

Displays a specified collection of lines for editing, while excluding others from the display. This is a macro.

```
ALter    char 1  char 2 [ target   [ n    [ p ] ] ]
                        [          [ *    [   ] ] ]
                        [ 1        [ 1    [ 1 ] ] ]
                        [          [ G    [   ] ] ]
```

Changes a single character to another character unavailable on a terminal keyboard by referencing its hexadecimal value. This is a macro.

BAckward [*n*|* 1]

Scrolls backward the number of screen displays specified.

Bottom

Makes the last line of the file or of the range (see SET RANGE) the new current line.

CANCEL

Terminates the editing session for all of the files. This is a macro.

CAppend [*text*]

Appends specified text to the end of the current line. This is a macro.

CDelete [*column-target* | 1]

Deletes one or more characters from the current line, starting at the column pointer.

CFirst

Moves the column pointer to the beginning of the zone (**see** SET ZONE).

XEDIT

```
Change     / string 1  [ / string 2 / [ target  [ p  [ q ] ] ] ]
                                        1         *    1
                                                  1
```

Changes a specified group of characters on one or more lines at one time.

CInsert *text*

Inserts text into the current line immediately ahead of the column pointer.

CLAst

Moves the column pointer to the end of the zone (**see** SET ZONE).

CLocate *column-target*

Scans the file for a specified column-target starting at the column following (or preceding) the column pointer in the current line. Also finds successively **all** occurrences of a character string.

CMS [*commandline*]

Forces the editor to transmit a command to CMS for execution or causes the editor to enter CMS subset mode.

CMSG [*text*]

Displays a message in the command line; intended for issuance from a macro.

COMMAND [*commandline*]

Causes the editor to execute a specified XEDIT command without first checking for a synonym or macro with the same name.

COMPress [*target* |1]

Prepares one or more lines for automatic repositioning of data (**see** SET TABS).

COpy *target* 1 *target* 2

Copies one or more lines at a specified location in the file.

COUnt /*string* [/*target* |1]

Displays the number of times a specified character string appears in one or more lines, beginning at the current line.

COVerlay *text*

Selectively replaces one or more characters in the current line with the same number of characters keyed in.

CP [*commandline*]

Transmits commands to the VM/SP control program environment during an editing session.

CReplace *text*

Replaces one or more characters in the current line.

CURsor	**CMdline**	[*colno* | 1] [**Priority** *n*]
	Column	[**Priority** *n*]
	File	*lineno* [*colno*] [**Priority** *n*]
	Home	[**Priority** *n*]
	Screen	*lineno* [*colno*] [**Priority** *n*]

Moves the cursor to a specified position and assigns a priority to the specified position.

DELete [*target* | 1]

Deletes one or more lines from a file beginning with the current line.

Down [*n* | * |1]

Moves the line pointer down a specified number of lines toward the end of the file.

DUPlicat [*n* [*target*]]
1 1

Duplicates one or more lines beginning with the current line.

EMSG [*text*]
[*mmmnnn*[*n*] s]

Displays a message at the terminal; or used in macros and modules that interface with XEDIT and whose messages follow VM/SP message rules. The severity determines whether or not the alarm sounds.

EXPand [*target* | 1]

Repositions data in one or more lines that contain tab characters (X'05').

EXTract */operand*[*/operand*[*/operand* . . .]]

Operand may be any one of these keywords:

ACTion
ALT
APL
ARBchar
AUTosave
BASEft
BRKkey
CASE
CMDline
COLOR [field | *]
COLPtr
COLumn
CTLchar [char]
CURLIne
CURSor
DISPlay
EDIRName
EFMode
EFName
EFType
EOF
ESCape
ETARBCH
ETMODE
FILler
FLscreen
FMode
FName
FType
FULLread
HEX
IMage
IMPcmscp
INPmode
LASTLorc
LASTmsg
LENgth
LIBName
LIBType
LIne
LINENd
LOCk
LRecl
LScreen
MACRO
MASK
MEMber
MSGLine
MSGMode
NBFile
NONDisp
NULls
NUMber
PA [n|*]
PACK
PENDing (see below)
PF [n|*]
Point *
PREfix (see below)
RANge
RECFm
REMOte
RESERved [*]
RING
SCALe
SCOPE
SCReen
SELect
Seq8
SERial
SHADow
SIDcode
SIZe
SPAN
SPILL
STAY
STReam
SYNonym [name |*]
TABLine
TABS
TARGet
TERMinal
TEXT
TOF
TOFEOF
TOL
TRANSLat
TRunc
UNIQueid
UNTil
UPDate
VARblank
Verify
VERShift
WINdow
Width
WRap
Zone
=

PENDing [BLOCK][OLDNAME name |* [target1 [target2]]

PREfix [Synonym name | *]

Used within a macro to get information about internal XEDIT variables or about file data. The operand may be any one of the keywords listed below.

FILE $\left[\begin{matrix} fn \\ = \end{matrix} \left[\begin{matrix} ft \\ = \end{matrix} \left[\begin{matrix} fm \\ = \end{matrix}\right]\right]\right]$

Writes the edited file onto disk and optionally overrides the file identifier originally supplied.

Find *text*

Searches forward, starting with the current line, for the first line that corresponds to the text specified in the operand.

FINDUp *text*
FUp

Searches backward, starting with the current line, for the first line that corresponds to the text specified in the operand.

FOrward [*n* |* |$\underline{1}$]

Scrolls (toward the end of the file) the operand-specified number of screen displays.

GET $\left[\begin{matrix}fn\\=\end{matrix}\left[\begin{matrix}ft\\=\end{matrix}\left[\begin{matrix}fm\\=\end{matrix}\right]\left[\begin{matrix}firstrec\\\underline{1}\end{matrix}\quad\left[\begin{matrix}numrec\\\underline{*}\end{matrix}\right]\right]\right]\right]$

Inserts all or part of a specified CMS file following the current line of the edit file.

Help [|MENU |HELP |TASK| *name*]

Displays a list of all XEDIT subcommands and macros and their descriptions, formats, and parameters, or calls the CMS HELP command.

HEXType [*target* | 1]

Displays a specified number of lines in both hexadecimal and EBCDIC. This is a macro.

XEDIT

Input [*line*]

Inserts a single line into a file. Also used to leave edit mode for entry into input mode.

Join [**ALigned**] **Column**
CURSOR

[**ALigned**] *colno* ...
/string/

Combines two or more lines into one replacement line. The first format lets you join two lines at the column pointer or at the cursor. The second format lets you join two or more lines at a specified column number(s) or inserts a specified character string(s) before appending the next line.

LEft [*n* |1]

Allows viewing of columns not currently visible on the screen that lie to the left of the first column on the screen.

LOAD [*fn* [*ft* [*fm*]]] [(*options...* [)]]

Options:

[**Width** *nn*] [**NOSCreen**] [**PROFile** *macroname*]
[**NOPROFil**] [**NOCLear**] [**NOMsg**]
[**MEMber** *membername*] [**WINdow** *wname*]
[**LOCk**
NOLOCk]

Options Valid Only in Update Mode:

[**Update** | **NOUpdate**] [**Seq8** | **NOSeq8**] [**Ctl** *fn* 1 | **NOCtl**]

[**Merge**] [**UNtil** *filetype*] [**Incr** *nn*]
[**SIDcode** *string*]

Reads a copy of the file being edited into virtual storage. This subcommand **can only be issued from the XEDIT profile**. Lets the macro prompt for edit options or assign default values to edit variables. The LOAD subcommand has the same format and editing options as in the XEDIT command; however, the options specified in the XEDIT command override those specified in the LOAD subcommand.

[**Locate**] *target* [*subcommand*]

Scans file for a specified target, which (when found) becomes the current line.

LOWercas [*target* 1]

Changes all uppercase letters to lowercase letters in one or more lines.

LPrefix [*text*]

Simulates writing in the prefix area of the current line.

MACRO [*macroline*]

Causes the specified operand to be executed as a macro.

MErge *target* 1 *target* 2 [*col*]

Combines two sets of lines. The first set of lines is deleted and the second set is modified in place.

MODify *keyword*

keyword operands:

ALT
APL
ARBchar
AUtosave
BRKkey
CASE
CMDline
COLOR field
COLPtr
COLumn
CTLchar [char]
CURLine
DISPlay
ENTer
ESCape
ETARBCH
ETMODE
FILler
FMode
FName
FType
FULLread
HEX
IMage
IMPCMSP
LASTlorc
LINENd
LRecl
MACRO
MASK
MSGline
MSGMode
NONDisp
NULls
NUMber
PAn
PACK
PFn
PREfix [synonym name]
RANge
RECFm
REMOte
SCALe
SCOPE
SCReen
SELect
SERial
SHADow
SIDcode
SPAN
SPILL
STAY
STReam
SYNonym
TABLine
TABS
TERMinal
TEXT
TOFEOF
TRunc
VARblank
Verify
VERShift
WRap
Zone

Displays a subcommand and its current operand values so that new values can be typed over the current ones and the subcommand immediately reentered. This is a macro.

MOve *target* 1 *target* 2

Moves one or more lines, beginning with the current line, to a specified place in the file.

MSG [*text*]

Displays a message in the message area of the screen.

Next [*n* |* |1]

Advances the line pointer a specified number of lines toward the end of the file.

NFind *text*

Searches forward for the first line that does **not** start with the text specified in the operand.

NFINDUp *text*
NFUp

Searches backward for the first line that does **not** start with the text specified in the operand.

Overlay *text*

Replaces, selectively, one or more characters with nonblank characters starting at the first tab column of the current line.

```
PARSE     startcol  Alphaword
                    Number
                    String     . . .
                    Dblstring
                    Target
                    Word
                    Line
```

Helps in writing new macros by scanning the new macro(s) to see if the format-specified-operands match those in the macro. This is a macro.

POWerinp

Enters an input mode where data can be keyed in as though the screen were one long line.

PREServe

settings saved include :

ARBCHAR
AUTOSAVE
CASE
CMDLINE
COLOR
COLPTR
CURLINE
DISPLAY
ESCAPE
ETARBCH
FILLER
FMODE
FNAME
FTYPE
HEX
IMAGE
IMPCMSCP
LASTLORC
LINEND
LRECL
MACRO
MASK
MSGMODE
NULLS
NUMBER
PACK
PREFIX
RECFM
SCALE
SCOPE
SERIAL
SHADOW
SPAN
SPILL
STAY
STREAM
SYNONYM
TABLINE
TABS
TOFEOF
TRUNC
VARBLANK
VERIFY
WRAP
ZONE
=

Saves the settings of various XEDIT variables until a subsequent RESTORE subcommand is issued.

PURge *macroname*

Removes a copy of a macro in virtual storage.

PUT $\left[\begin{matrix} \textit{target} \\ \underline{1} \end{matrix} \quad \left[\begin{matrix} \textit{fn} \\ = \end{matrix} \left[\begin{matrix} \textit{ft} \\ = \end{matrix} \left[\begin{matrix} \textit{fm} \\ = \end{matrix}\right]\right]\right]\right]$

Inserts one or more lines, starting at the current line, into the end of an existing file or into a new file or into a temporary file created by the editor.

PUTD $\left[\begin{matrix}\textit{target}\\ \underline{1}\end{matrix}\quad \left[\begin{matrix}fn\\ =\end{matrix}\left[\begin{matrix}ft\\ =\end{matrix}\left[\begin{matrix}fm\\ =\end{matrix}\right]\right]\right]\right]$

Inserts one or more lines, starting with the current line, into the end of an existing file or into a new file or into a temporary file. This command deletes the specified lines from the original file.

Query

ACTion
ALT
APL
ARBchar
AUtosave
BASEft
BRKkey
CASE
CMDline
COLOR *|*field*
COLPtr
COLumn
CTLchar [*char*]
CURLine
CURSor
DISPlay
EDIRName
EFMode
EFName
EFType
ENTer
EOF
EOL
ESCape
ETARBCH
ETMODE
FILler
FMode
FName
FType
FULLread
HEX
IMage
IMPcmscp
LASTLorc
LASTmsg
LENgth
LIBName
LIBType
LIne
LINENd
LRecl
LScreen
MACRO
MASK
MEMber
MSGLine
MSGMode
NBFile
NONDisp
NULls
NUMber
PA [*n* |*]
PACK
PENDing [BLOCK][OLDNAME] *name* |*
PF [*n* |*]
Point [*]
PREfix [Synonym *|*name*]
RANge
RECFm
REMOte
RESERved
RING
SCALe
SCOPE
SCReen
SELect
Seq8
SERial
SHADow
SIDcode
SIZe
SPAN
SPILL
STAY
STReam
SYNonym [*|*name*]
TABLine
TABS
TARGet
TERMinal
TEXT
TOF
TOFEOF
TOL
TRANSLat
TRunc
UNIQueid
UNTil
UPDate
VARblank
Verify
VERShift
Width
WRap
Zone
=

Displays the current setting of various editing options.

```
QUIT      [n]
```

Terminates the editing session and leaves the previous copy intact.

```
READ      [Cmdline              ]   [Tag  ]
          [All  [Number]        ]   [     ]
          [Nochange  [Number]   ]   [Notag]
```

Places data from the terminal into the console stack (LIFO). This subcommand generally is issued from a macro.

```
RECover      [n |* |1]
```

Replaces a specified number of lines removed by a DELETE or PUTD subcommand or a D (delete) prefix subcommand.

```
REFRESH
```

Displays the screen. Issued from a macro, it presents the screen as of that moment in processing, without waiting for input.

```
RENum        [startno [incr]]
                10
```

Renumbers the line numbers of VSBASIC and FREEFORT files.

REPEat [*target* 1]

Advances the line pointer and executes the last subcommand entered.

Replace [*text*]

Replaces the current line with a specified line or keyed in text, or deletes the current line and enters input mode.

RESet

Removes all prefix subcommands when the screen is in a *pending* or *incomplete* status.

RESTore

Restores the settings of the XEDIT variables to the values in effect when last the PRESERVE subcommand was issued.

RGTLEFT [*n*]

Allows viewing of columns of data not currently visible on the screen.

RIght $[n \mid \underline{1}]$

Allows viewing of data in columns not currently visible on the screen. These columns are to the right of the right-most column on the screen.

SAVE $\left[\begin{matrix} fn \\ = \end{matrix} \left[\begin{matrix} ft \\ = \end{matrix} \left[\begin{matrix} fm \\ = \end{matrix}\right]\right]\right]$

Enters the file that is currently being edited onto disk without returning control to CMS.

SCHANGE [*keynumber*]

Locates every occurrence of a string and changes the string only when specified to do so. This is a macro.

SET *option*

Options include:

ALT
APL
ARBchar
AUtosave
BRKkey
CASE
CMDline
COLOR
COLPtr
CTLchar
CURLine
DISPlay
ENTer
ESCape
ETARBCH
ETMODE
FILler
FMode
FName
FType
FULLread
HEX
IMage
IMPcmscp
LASTLorc
LINENd
LRecl
MACRO
MASK
MSGLine
MSGMode
NONDisp
NULls
NUMber
PAn
PACK
PENDing
PFn
Point
PREfix
RANge
RECFm
REMOte
RESERved
SCALe
SCOPE
SCReen
SELect
SERial
SHADow
SIDcode
SPAN
SPILL
STAY
STReam
SYNonym
TABLine
TABS
TERMinal
TEXT
TOFEOF
TRANSLat
TRunc
VARblank
Verify
WRap
Zone
=

Changes the settings of various editing options while editing is in progress.

SET **ALT** *n* [*p*]

Changes the number of alterations that have been made to the file since the last SAVE or AUTOSAVE.

[**SET**] **APL** **ON**
OFF

Shows whether APL keys are available.

```
[SET]     ARBchar    ON     [char]
                     OFF
```

Defines an arbitrary character used in a target definition. Note that the initial setting is OFF.

```
[SET]     AUtosave   n   [mode]
                         [ A  ]
                     OFF
```

Sets or resets the automatic save function of the editor. Note that the initial setting is OFF.

```
[SET]     BRKkey     ON        key
                     OFF
```

Specifies whether CP should break in when the “BRKKEY” (defined by CP TERMINAL BRKKEY) is pressed.

```
[SET]     CASE  Uppercase   [Respect]
                Mixed       [Ignore ]
```

Controls letters entered and specifies significance in target searches.

```
[SET]     CMDline    On
                     OFf
                     Top
                     Bottom
```

Specifies the position of the command line on the screen.

```
[SET]       COLOR   field   [color][exthi] [High|Nohigh] [PSs]
                      *
```

Associates specific colors with certain areas of the XEDIT screen.

```
[SET]       COLPtr      ON
                        OFF
```

Determines (on typewriter terminals) whether or not the column pointer (underscore) is displayed.

```
[SET]       CTLchar  char  Escape
                           OFF
                           Protect    [color][exthi] [High|Nohigh |Invisible][PSs]
                           Noprotect  [color][exthi] [High|Nohigh |Invisible][PSs]
                     OFF
```

Defines control character.

```
[SET]       CURLine   ON    M[+n |-n] |  [+|-]n
```

Defines the *n*th line of the screen as the current line. Note that, on initial setting, the n is the middle line of the screen.

```
[SET]       DISPlay  n1  [n2|*]
```

Specifies which selection level of lines (as displayed by SET SELECT) are displayed.

```
[SET]      ENTer    BEFORE    string
                    AFTER     NULLKEY
                    ONLY      COPYKEY
                    IGNORE    TABKEY
```

Defines a meaning for the hardware ENTER key or removes the meaning associated with the ENTER key.

```
[SET]      ESCape     ON    [char]
                      OFF
```

Allows entry of subcommand (on typewriter terminals) when in input mode without leaving input mode.

```
[SET]      ETARBCH    ON    [char]
                      OFF
```

Defines an extended arbitrary character used in a target definition within a DBCS string. The initial setting is OFF.

```
[SET]      ETMODE     ON
                      OFF
```

Inform the editor that there are double-byte characters in the file. The initial setting is OFF.

```
[SET]      FILler  [char]
```

Defines a character to be used when expanding a line (see EXPAND subcommand).

[SET] **FMode** *fm*

Changes the file mode of the edited file.

[SET] **FName** *fn*

Changes the file name of the edited file.

[SET] **FType** *ft*

Changes the file type of the edited file.

[SET] **FULLread** **ON**
OFF

Lets 3270 null characters be recognized in the middle of screen lines.

[SET] **HEX** **ON**
OFF

Lets subcommand operands and targets be specified in hexadecimal. Note that the initial setting is OFF.

[SET]	**IMage**	**ON** **OFF** **Canon**	

Determines how tab characters (X'05') and backspace characters (X'16') are handled.

[SET]	**IMPcmscp**	**ON** **OFF**	

Determines whether or not non-XEDIT recognized subcommands are implicitly transmitted to CMS, and later to CP, for execution.

[SET]	**LASTLorc**	*line*	

Specifies the contents of the LASTLORC subcommand (used within a macro).

[SET]	**LINENd**	**ON** **OFF**	[*char*]

Determines whether or not # (pound sign) or other character is used as the line end character.

[SET]	**LRecl**	*n* *	

Defines a new logical record length for writing file to disk.

```
[SET]      MACRO      ON
                      OFF
```

Controls sequence of editor's search for subcommands and macros. Note that the initial setting is OFF.

```
[SET]      MASK     Define
                    Immed   [text]
                    Modify
```

Changes contents of mask. Note that the initial setting is a blank line.

```
[SET]        MSGLine   ON      M[+n |-n] |  [±|-]n   [p |1]   [Overlay]
                       OFF
```

Defines the location of the message line on the screen, and the maximum number of lines that a message can occupy.

```
[SET]      MSGMode    ON     [Short|Long]
                      OFF
```

Controls message display. Note that the initial setting is ON LONG.

```
[SET]      NONDisp  [char]
```

Defines a character to use in place of a nondisplayable character.

```
[SET]    NULls   ON
                 OFF
```

Specifies whether trailing blanks in each line are written to the screen as blanks (X'40') or nulls (X'00'). Note that the initial setting is OFF.

```
[SET]    NUMber     ON
                    OFF
```

Determines whether or not line numbers are displayed in the prefix area. Note that the initial setting is OFF.

```
[SET]     PAn     ┌BEFORE┐  ┌string ┐
                  │AFTER │  │NULLKEY│
                  │ONLY  │  │COPYKEY│
                  └IGNORE┘  └TABKEY ┘
```

Defines a meaning for a specified hardware attention (PA) key or removes the meaning associated with the specified PA key.

```
[SET]    PACK    ON
                 OFF
```

Specifies whether or not packed file is entered on disk.

```
[SET]      PENDing   ON      string
                     BLOCK   string
                     ERROR   string
                     OFF
```

Controls the execution of a prefix macro and the status of the screen while the prefix macro is being executed.

```
[SET]      PFn     ┌BEFORE┐  ┌string  ┐
                   │AFTER │  │NULLKEY │
                   │ONLY  │  │COPYKEY │
                   └IGNORE┘  └TABKEY  ┘
```

Defines or removes a meaning for a specified program function (PF) key. Note that TABKEY is the initial setting of the PF4 dey.

```
[SET]      Point  .symbol  [OFF]
```

Defines or redefines the symbolic name for the current line.

```
[SET]      PREfix    ON       [Left|Right]
                     OFF
                     Nulls    [Left|Right]

           PREfix    Synonym   newname    oldname
```

Controls display of the prefix area. Also defines a synonym for a prefix subcommand.

```
[SET]      RANge   target 1  target 2
```

Defines new limits for line-pointer movement.

```
[SET]      RECFm    F
                    V
                    FP
                    VP
```

Defines the record format for the file.

```
[SET]      REMOte    ON
                     OFF
```

Controls the way XEDIT handles the display, in terms of data transmission.

```
[SET]      RESERved  M [+n | -n]  [color][exthi] [PSs] High   [text]
                      [± | -] n                        Nohigh
                                   Off
```

Reserves a specific line on the screen for displaying blank or specified information with or without any of the following features: color, extended highlighting, highlighting, and programmed symbol set.

```
[SET]      SCALe  ON    [M[+n |-n]  |  [± | -]n]
                  OFF
```

Displays a scale line under the current line (the default) or on a specified line.

```
[SET]      SCOPE   Display
                   All
```

Specifies the set of lines on which the editor operates.

```
[SET]     SCReen    n [Horizontal |Vertical]
                    Size   s1 [s2 [s3 ...[sn]]]
                    Width  w1 [w2 [w3 ...[wn]]]
                    Define sl1 sw1 sh1 sv1 [sl2 sw2 sh2 sv2]...
```

Divides the screen into a specified number of logical screens to allow editing of multiple files or multiple views of the same file.

```
[SET]     SELect    [+ | -]n     [target |1]
```

Designates a "selection level" for specified lines. A selection level is a positive value assigned to a line in a file.

```
[SET]     SERial    ON       [incno     [startno]]
                              10         10
                    ALL      [incrno    [startno]]
                              1000       1000
                    string   [incrno    [startno]]
                              10         10
                    OFF
```

Controls file serialization.

```
[SET]     SHADow    ON
                    OFF
```

Displays a notice (called a shadow line) that indicates how many lines have been excluded from the display.

```
[SET]     SIDcode   [string]
```

Inserts a character string in every line of an update file.

```
[SET]     SPAN     ON    [Blank      [n]]
                         [Noblank    [*]]

                   OFF
```

Specifies whether a target-search character string must be included in one line or span a certain number of lines. Note that the initial setting is OFF Blank 2.

```
[SET]     SPILL   ON
                  OFF
                  WORD
```

Specifies if data is spilled onto new lines or lines are truncated following these subcommands: CHANGE, CINSERT, COVERLAY, CREPLACE, EXPAND, GET, INPUT, MERGE, OVERLAY, REPLACE, SHIFT, (and macros that use these subcommands internally, including CAPPEND, JOIN and PRFSHIFT(>,>>)).

```
[SET]     STAY   ON
                 OFF
```

Specifies whether or not the line pointer moves when target-search object is not found. Note that the initial setting is OFF.

[SET] **STReam** **ON**

Specifies whether to search entire file or only the current line for a character string. Note that the initial setting is ON.

[SET] **SYNonym** **ON**
OFF

SYNonym [**LINEND** *char*] *newname* [*n*] *oldname*

SYNonym [**LINEND** *char*] *newname* [*n* [*format* 1... *format n*]]
oldname [&1...& *n*]

Specifies whether or not to look for synonyms. Also assigns a synonym to any existing subcommand or macro (except prefix subcommands or prefix macros) and defines an abbreviation for the synonym.

[SET] **TABLine** **ON** [M[+*n* |-*n*] | [±|-]*n*]
OFF

Displays a "T" in every tab column according to current tab settings.

[SET] **TABS** *n*1 [*n*2... *n*28]

Defines the logical tab stops for a file.

[SET] **TERMinal** **Typewriter**
Display

Specifies whether a terminal is to be used in line mode or in full- screen mode.

```
[SET]     TEXT     ON
                   OFF
```

Shows whether keys are available.

```
[SET]     TOFEOF     ON
                     OFF
```

Controls the display of Top of File, End of File, Top of Range, and End of Range null lines. Note that the initial setting is ON.

```
[SET]     TRANSLat   char1 char2 [char1 char2]   .....
                     OFF
```

Controls uppercase translation of specified characters. This option is designed for terminals whose keyboards support characters other than English.

```
[SET]     TRunc  n
                 *
```

Defines last column in which data may be entered.

```
[SET]     VARblank   ON
                     OFF
```

Controls whether or not the number of blanks between two words is significant in target search. Note that the initial setting is OFF.

```
[SET]     Verify  ON  [[Hex] startcol  endcol]   ...
                  OFF [[Hex] startcol  endcol]   ...
                       [Hex] startcol  endcol    ...
```

Controls whether or not subcommand(s)-changed lines are to be displayed. Also defines columns to be displayed on screen. Data can also be displayed in hexadecimal.

```
[SET]     WRap   ON
                 OFF
```

Controls use of wraparound. The initial setting is OFF.

```
[SET]     Zone   zone 1  zone 2
                           *
```

Defines starting and ending column of each record for target search scanning.

```
[SET]     = string
```

Inserts specified string into the equal buffer (see = subcommand).

```
SHift     Left    [cols [target]]
          Right   [ 1   [  1   ]]
```

Moves data either to the left or to the right. Note that data loss is possible.

SI

Continually adds lines for indented text to a file. A line is added immediately following the line that contains the cursor. The cursor is positioned at the column where the text on the previous line begins.

SORT *target* [A] *col* 1 *col* 2 [*col* 1 *col* 2] ...
[D]

Arranges a specified number of file lines in ascending or descending EBCDIC sequence according to specified sort columns.

SOS *option*

Options:

Alarm
CLEAR
LINEAdd
LINEDel
NUlls
NUlls ON
NUlls OFF
PFn
POP
PUsh
TABB [*n* |1]
TABCmd
TABCMDB [*n* |1]
TABCMDF [*n* |1]
TABF [*n* |1]

Provides a set of functions used mainly in XEDIT macros or assigned to PF keys.

```
SPlit          [ALigned] [ Column ]
                         [ CURSOR ]

                         [ colno                ]
               [ALigned] [ [Before]   /string / ]   ...
                         [ [After ]             ]
```

Splits a line into two or more lines at the column pointer or at the cursor. The second format splits a line into several lines. This is a macro.

SPLTJOIN

Either splits a line or joins two lines, depending on the position of the cursor on a file line. If the cursor is positioned before or at the last nonblank character, the line is split (at the cursor position). If the cursor is positioned after the last nonblank character on a line (that is, after the end of the data on a line), the next line is appended, starting at the cursor position. This is a macro.

```
STAck      [ target   [ startcol   [ length ]]]
           [   1      [    1       [   *    ]]]
```

Places part or all of a specified number of lines into the console stack, starting with the current line.

```
STATus       [filename]
```

Displays the SET subcommand options and their current settings or creates an XEDIT macro that contains the SET subcommands with their current settings. This is a macro.

TOP

Moves the line pointer to the null line above the first line of the file or of the range (**see** SET RANGE).

TRAnsfer *keyword* ...

keywords: (more than one can be specified)

APL
ARBchar
AUtosave
CASE
CMDline
COLPtr
COLumn
CTLchar [char]
CURLine
CURSor
EOF
ESCape
FILler
FMode
FName
FType
HEX
IMage
IMPcmscp
LASTmsg
LENgth
LIne
LINENd
LRecl
LScreen
MACRO
MASK
MSGMode
NBFile
NONDisp
NULls
NUMber
PACK
PFn
Point
PREfix
RANge
RECFm
RESERved
SCALe
SCReen
Seq8
SERial
SIDcode
SIZe
SPAN
STAY
STReam
SYNonym [name]
TABLine
TABS
TARGet
TERMinal
TEXT
TOF
TOFFEO
Trunc
UPDate
VARblank
Verify
VERShift
Width
WRap
Zone
=

Accesses, within a macro, specified editing variables and places their values in the console stack for subsequent reading by the EXEC 2 &READ control statements.

Type [*target* |1]

Displays a specified number of lines, starting with the current line.

Up [*n* |* |1]

Moves the line pointer a specified number of lines toward the top of the file.

UPPercas [*target* | 1]

Translates all lowercase characters to uppercase ones, starting at the current line.

Xedit [*fn* [*ft* [*fm*]]] [(*options*... [)]]

When entered from the file being edited, enables the editing of multiple files in virtual storage.

Options: are the same as the command options (see XEDIT command).

XEDIT Prefix subcommands

A - add one line
***n*A - add *n* lines**
A*n* - add *n* lines

Adds one or more lines immediately after the line in which the subcommand is entered.

C - copy line
C*n* - copy *n* lines
***n*C - copy *n* lines**
CC - copy block of lines

Copies one or more lines to a location specified with the F or P prefix subcommand.

D - **Delete one line**
D*n* - **Delete *n* lines**
***n*D** - **Delete *n* lines**
DD - **Delete block of lines**

Deletes one or more lines starting at the line in which subcommand is entered.

E

Extend logical line by one more physical line.

F

Data is entered following this point (using the C or M prefix subcommand)

I - **Insert one line**
***n*I** - **Insert *n* lines**
I*n* - **Insert *n* lines**

Inserts one or more lines immediately following line in which subcommand is entered.

M - **move one line**
M*n* - **move *n* lines**
***n*M** - **move *n* lines**
MM - **move block of lines**

Moves one or more lines to a location specified with the F or P prefix subcommand.

P

Data is entered preceding this point (using the C or M prefix subcommands).

S - show all lines
S* - show all lines
S*n* - show the first *n* lines
S+*n* - show the first *n* lines
***n*S** - show the first *n* lines
S-*n* - show the last *n* lines

Redisplays one or more lines excluded by the X prefix subcommand.

SCALE

Displays the scale on this line.

SI

Continually add lines for indented text to a file. A line is added immediately following the line that contains the cursor. The cursor is positioned at the column where the text on the previous line begins.

TABL

Displays a "T" in every tab column in the line

X - exclude one line from display
X*n* - exclude *n* lines from display
***n*X - exclude *n* lines from display**
XX - exclude a block of lines from display

Excludes one or more lines from display (and the scope of editing subcommands) starting with the line in which the subcommand is entered.

.xxxx

Assigns xxxx as symbolic name to this line.

/[*n*] or [*n*] /

Make this line current and move the column pointer under the nth column.

> - shift one line one column to the right
> *n* - shift one line *n* columns to the right
***n* > - shift one line *n* columns to the right**
> > - shift a block of lines one column to the right
> > *n* - shift a block of lines *n* columns to the right
***n* > > - shift a block of lines *n* columns to the right**

Shifts one or a block of lines (starting with the line in which the subcommand is entered) one or more columns to the right.

< - **shift one line one column to the left**
< *n* - **shift one line** *n* **columns to the left**
n **<** - **shift one line** *n* **columns to the left**
< < - **shift a block of lines one column to the left**
< < *n* - **shift a block of lines** *n* **columns to the left**
n **< <** - **shift a block of lines** *n* **columns to the left**

Shifts one or a block of lines (starting with the line in which the subcommand is entered) one or more columns to the left.

" - **duplicate one line**
" *n* **or** *n* " - **duplicate line** *n* **times**
" " - **duplicate block of lines**
" "*n* **or** *n* " " - **duplicate block** *n* **times**

Duplicates one or a block of lines, either one time or a specified number of times.

XMITMSG

CMS

Retrieves a message from a CMS message repository file or your own message repository file.

```
XMITMSG   msgnumber [sublist ]  [(options... [)]]

          Options:
           [FORmat nn] [LINE  nn]   [LETter a]
                       [      * ]

           [APPLID applid] [CALLER name]    [VAR]

           [COMPress  ] [HEADer  ] [DISPlay  ]
           [NOCOMPress] [NOHEADer] [NODISPlay]
                                   [ERRMSG   ]

           [SYSLANG]
```

ZAP

CMS

Modifies or dumps MODULE, LOADLIB, or TXTLIB files.

```
ZAP       ┌MODULE ┐
          │LOADLIB│ [libname1 ... libname3] [(option ...[)]]
          └TXTLIB ┘

        Options:
          ┌TERM          ┐  ┌PRINT  ┐
          └INPUT filename┘  └NOPRINT┘
```

ZAPTEXT

CMS

Modifies or dumps individual text files.

```
ZAPTEXT     fn [ft [fm]]  [(options: ... [)]]

          Options:
          [INPUT filename]  ┌PRINT  ┐
                            └NOPRINT┘
```

Summary of Changes

New Commands for Release 6

CMS Commands

ALIALIST
AUDIT
AUTHLIST
BACKUP
CREATE ALIAS
CREATE DIRECTORY
CREATE LOCK
CREATE NAMEDEF
CSLGEN
CSLLIST
DELETE ADMINISTRATOR
DELETE LOCK
DELETE NAMEDEF
DELETE USER
DIRLIST
DISABLE
ENROLL ADMINISTRATOR
ENROLL PUBLIC
ENROLL USER
ETRACE
FILEPOOL BACKUP
FILEPOOL CLEANUP
FILEPOOL FORMAT AUDIT
FILEPOOL RESTORE
FILESERV BACKUP
FILESERV DEFAUDIT
FILESERV DEFBACKUP
FILESERV GENERATE
FILESERV LIST
FILESERV LOG
FILESERV MINIDISK
FILESERV MOVEUSER
FILESERV REGENERATE
FILESERV REORG
FILESERV START
FORCE
GRANT ADMIN
GRANT AUTHORITY
LISTDIR
MODIFY USER
NETDATA
PROGMAP
QUERY ACCESSED
QUERY ALIAS
QUERY AUTHORITY
QUERY COMDIR
QUERY CSLLIB
QUERY DISABLE
QUERY FILEPOOL CONFLICT
QUERY FILEPOOL STATUS
QUERY KEYPROTECT
QUERY LIMITS
QUERY LOADAREA
QUERY LOCK
QUERY NAMEDEF
QUERY SEGMENT
QUERY SERVER
QUERY STORECLR
RELOCATE
REVOKE ADMIN
REVOKE AUTHORITY
RTNDROP
RTNLOAD
RTNMAP
RTNSTATE
SEGGEN
SEGMENT ASSIGN
SEGMENT LOAD
SEGMENT PURGE
SEGMENT RELEASE
SEGMENT RESERVE
SET COMDIR
SET FILEPOOL
SET FILEWAIT
SET KEYPROTECT
SET LOADAREA
SET SERVER
SET STORECLR
SET THRESHOLD
SETKEY
STOP

CP Commands

GIVE

IPCS Commands

IPCSSCAN
IPCSSCAN Subcommands
BOTTOM
DOWN
FORMAT
GDISPLAY
HELP
HEX
IDENTIFY
LOCATE (UP)
LUNAME
SELECT
TIME
TOP
TRACE
UP
IPCSPRT
IPCSPRT Subcommands

END
FORMAT
HELP
HEX
HX
PROCESS
QUIT
SELECT
TIMESPAN

REXX Built-in Functions

FORM
FUZZ
WORDPOS

AVS Commands

AGW ACTIVATE GATEWAY
AGW CNOS
AGW DEACTIVATE CONV
AGW DEACTIVATE GATEWAY
AGW QUERY
AGW QUIESCE
AGW SET ETRACE
AGW SET ITRACE
AGW START
AGW STOP

RSCS Commands

DISABLE
ENABLE
EXIT
PORT
REORDER

Changed Commands for Release 6

CMS Commands

(E)STATE/(E)STATEW
ACCESS
CMSSERV
COPYFILE
DDR
DEFAULTS
DISK
ERASE
EXECMAP
FILELIST
FINIS
GLOBAL
HELP
HELPCONV
LISTFILE
NAMEFIND

NUCXMAP
QUERY
READCARD
RECEIVE
RELEASE
RENAME
SET
XEDIT

CP Commands

ATTACH (CLASS B)
CPTRAP (CLASS C)
DEFINE (CLASS A)
DETACH (CLASS B)
MONITOR (CLASS A, E)
NETWORK (CLASS A, B)
QUERY (CLASS G)
SET (CLASS A and G)
SPOOL (CLASS G)
SPTAPE (CLASS D)
TERMINAL (CLASS G)
TRACE (CLASS G)

RSCS Commands

DEFINE
ROUTE
START

GCS Commands

ACCESS
ETRACE
FILEDEF
QUERY

IPCS Commands

APAR
IPCSDUMP
MAP
PROB

TSAF Commands

ADD
DELETE
QUERY

REXX Instructions

NUMERIC
TRACE

XEDIT Subcommand

QUERY

Bibliography

Prerequisite Publications

Virtual Machine/System Product:

- *Introduction*, GC19-6200
- *CMS Command Reference*, SC19-6209
- *CMS User's Guide*, SC19-6210
- *CP General User Command Reference*, SC24-5401
- *CMS Primer*, SC24-5236
- *CMS Primer for Line-Oriented Terminals*, SC24-5242
- *CP System Command Reference*, SC24-5402

Corequisite Publications

Virtual Machine/System Product:

- *Planning Guide and Reference*, SC19-6201
- *Operator's Guide*, SC19-6202
- *System Messages and Codes*, SC19-6204
- *Terminal Reference*, GC19-6206
- *System Product Editor User's Guide*, SC24-5220
- *System Product Interpreter User's Guide*, SC24-5238
- *System Product Interpreter Reference*, SC24-5239
- *System Messages Cross-Reference*, SC24-5264

EXEC 2 Reference, SC24-5219

System Product Editor Command and Macro Reference, SC24-5221

Installation Guide, SC24-5237

Distributed Data Processing Guide, SH24-5241

Application Development Guide for FORTRAN and COBOL, SC24-5247

Group Control System Command and Macro Reference, SC24-5250

Interactive Problem Control System Guide and Reference, SC24-5260

Application Development Reference for CMS, SC24-5284

Administration, SC24-5285

Application Development Guide for CMS, SC24-5286

System Facilities for Programming, SC24-5288

Programmer's Guide to the Server-Requester Programming Interface for VM/SP, SC24-5291

Application Migration Guide for CMS, SC24-5366

CMS Shared File System Administration, SC24-5367

Connectivity Programming Guide and Reference, SC24-5377

Connectivity Planning, Administration, and Operation, SC24-5378

Virtual Machine/Remote Communications Spooling Subsystem (RSCS):

Operation and Use, SH24-5058

Index

A

ACCESS command (CMS) 39
ACCESS command (GCS) 39
ACNT command (CP class A) 40
ADD LINK command (TSAF) 40
ADSTOP command (CP class G) 40
AGW ACTIVATE GATEWAY command (AVS) 41
AGW CNOS 41
AGW DEACTIVATE CONV command (AVS) 41
AGW DEACTIVATE GATEWAY command (AVS) 41
AGW QUERY command (AVS) 42
AGW QUIESCE command (AVS) 42
AGW SET ETRACE command (AVS) 43
AGW SET ITRACE command (AVS) 43
AGW START command (AVS) 43
AGW STOP command (AVS) 44
ALARM VSCREEN command (CMS) 44
ALIALIST command
  usage with FILELIST command 118
AMSERV command (CMS) 44
APAR command (IPCS) 45
APPC/VM VTAM Support (AVS) commands
  See AVS (APPC/VM VTAM Support) commands
ASMGEND command (CMS) 45
ASM3705 command (CMS) 45
ASSEMBLE command (CMS) 46
ASSGN command (CMS) 47
ATTACH command (CP class B) 47
ATTN command (CP class G) 48
AUDIT command (CMS) 48
AUTHLIST command
  usage with DIRLIST command 80
  usage with FILELIST command 118
AUTOLOG command (CP class A,B) 48
AVS (APPC/VM VTAM Support) commands
  AGW ACTIVATE GATEWAY (AVS) 41
  AGW CNOS 41
  AGW QUERY 42
  AGW QUIESCE 42
  AGW SET ETRACE 43
  AGW SET ITRACE 43
  AGW START 43
  AGW STOP 44

B

B (CMS Border Command) 49
BACKSPAC command (CP class D) 49
BACKSPAC command (RSCS) 49
BACKUP command (CMS) 50
BEGIN command (CP class G) 50

C

C (CMS Border Command) 50
CATCHECK command (CMS) 51
CHANGE command (CP class D) 51
CHANGE command (CP class G) 51
CHANGE command (RSCS) 52
CLEAR VSCREEN command (CMS) 53

CLEAR WINDOW command (CMS) 54
CLOSE command (CP class G) 54
CMD command (RSCS) 54
CMDCALL command (CMS) 55
CMS 7
CMS Border Commands
  B 49
  C 50
  D 65
  F 114
  H 136
  L 162
  M 170
  N 178
  O 185
  P 188
  R 217
  S 249
  X 318
CMS DASD Dump Restore (DDR) command 10
CMS DDR command return codes 10
CMS file identifiers
CMS immediate commands
  HB 136
  HI 138
  HO 139
  HT 139
  HX 140
  RO 245
  RT 248
  RTNDROP 246
  RTNLOAD 247
  RTNMAP 247
  RTNSTATE 248
  SO 283
  TE 297
  TS 302
CMS reserved file types 20
CMS return codes 7
CMS (Conversational Monitoring System) commands
  ACCESS 39
  AGW DEACTIVATE CONV 41
  AGW DEACTIVATE GATEWAY 41
  ALARM VSCREEN 44
  AMSERV 44
  ASMGEND 45
  ASM3705 45
  ASSEMBLE 46
  ASSGN 47
  B (CMS Border Command) 49
  BACKUP 50
  C (CMS Border Command) 50
  CATCHECK 51
  CLEAR VSCREEN 53
  CLEAR WINDOW 54
  CMDCALL 55
  CMSBATCH 55
  CMSGEND 55
  CMSSERV 56
  COMPARE 56
  CONVERT COMMANDS 57
  CONWAIT 58
  COPYFILE 58
  CP 59
  CREATE ALIAS 62
  CREATE DIRECTORY 62
  CREATE LOCK 63
  CREATE NAMEDEF 63
  CSLGEN 64
  CSLLIST 64
  CURSOR VSCREEN 65
  D (CMS Border Command) 65
  DCSSGEN 66
  DDR 67
  DEBUG 68
  DEFAULTS 68
  DEFINE VSCREEN 73

CMS (Conversational Monitoring System)
  commands *(continued)*
    DEFINE WINDOW 73
    DELETE ADMINISTRATOR 74
    DELETE LOCK 75
    DELETE NAMEDEF 75
    DELETE PUBLIC 75
    DELETE USER 76
    DELETE VSCREEN 76
    DELETE WINDOW 76
    DESBUF 77
    DIRECT 78
    DIRLIST 81
      AUTHLIST command 81
    DISK 83
    DISKMAP 84
    DLBL 85
    DOSGEN 87
    DOSLIB 87
    DOSLKED 87
    DROP WINDOW 88
    DROPBUF 89
    DSERV 89
    EDIT 90
    ENABLE 90
    ENROLL ADMINISTRATOR 91
    ENROLL PUBLIC 92
    ENROLL USER 92
    ERASE 93
    ESERV 94
    ESTATE, ESTATEW 94
    ETRACE 95
    EXEC 96
    EXEC 2 97
    EXECDROP 109
    EXECIO 109
    EXECLOAD 110
    EXECMAP 111
    EXECOS 112
    EXECSTAT 112

CMS (Conversational Monitoring System)
  commands *(continued)*
    EXECUPDT 112
    EXPAND 113
    F (CMS Border Command) 114
    FETCH 114
    FILEDEF 115
    FILELIST 118
    FILEPOOL BACKUP 119
    FILEPOOL CLEANUP 119
    FILEPOOL FORMAT AUDIT 120
    FILEPOOL RESTORE 120
    FILESERV BACKUP 120
    FILESERV DEFAUDIT 121
    FILESERV DEFBACKUP 121
    FILESERV GENERATE 121
    FILESERV LIST 122
    FILESERV LOG 122
    FILESERV MINIDISK 122
    FILESERV MOVEUSER 123
    FILESERV REGENERATE 123
    FILESERV REORG 123
    FILESERV START 124
    FINIS 124
    FORMAT 126
    GENDIRT 129
    GENIMAGE 129
    GENMOD 130
    GENMSG 130
    GENSERVE 131
    GENTSAF 131
    GEN3705 131
    GET VSCREEN 132
    GLOBAL 132
    GLOBALV 134
    GRANT ADMIN 135
    GRANT AUTHORITY 135
    GROUP 135
    H (CMS Border Command) 136
    HB 136

CMS (Conversational Monitoring System)
  commands *(continued)*
    HELP 137
    HELPCONV 138
    HI 138
    HIDE WINDOW 138
    HO 139
    HT 139
    HX 140
    IDENTIFY 140
    IMAGELIB 141
    IMAGEMOD 141
    IMMCMD 141
    INCLUDE 142
    ITASK (CMS) 161
    L (CMS Border Command) 162
    LABELDEF 163
    LANGGEN 163
    LANGMERG 164
    LISTDIR 164
    LISTDS 165
    LISTFILE 165
    LISTIO 166
    LKED 166
    LOAD 167
    LOADLIB 168
    LOADMOD 168
    M (CMS Border Command) 170
    MACLIB 171
    MACLIST 172
    MAKEBUF 172
    MAXIMIZE WINDOW 173
    MINIMIZE WINDOW 174
    MODIFY USER 175
    MODMAP 175
    MOREHELP 177
    MOVEFILE 177
    N (CMS Border Command) 178
    NAMEFIND 179
    NAMES 179

CMS (Conversational Monitoring System)
  commands *(continued)*
    NCPDUMP 179
    NETDATA 180
    NOTE 183
    NUCXDROP 184
    NUCXLOAD 184
    NUCXMAP 184
    O (CMS Border Command) 185
    OPTION 185
    OSRUN 187
    P (CMS Border Command) 188
    PARSECMD 188
    PEEK 189
    POP WINDOW 191
    POSITION WINDOW 191
    PRELOAD 192
    PRINT 193
    PROGMAP 194
    PSERV 198
    PUNCH 198
    PUT SCREEN 200
    PUT VSCREEN 200
    QUERY 201
    QUERY DISABLE 214
    QUERY FILEPOOL CONFLICT 215
    QUERY FILEPOOL STATUS 215
    QUERY LIMITS 216
    R (CMS Border Command) 217
    RDR 217
    RDRLIST 218
    READCARD 219
    RECEIVE 220
    REFRESH 220
    RELEASE 221
    RELOCATE 221
    RENAME 222
    RESERVE 224
    RESTORE WINDOW 224
    REVOKE ADMIN 225

CMS (Conversational Monitoring System)
  commands *(continued)*
    REVOKE AUTHORITY 225
    REXX 226
    RO 245
    ROUTE 245
    RSERV 246
    RT 248
    RTNDROP 246
    RTNLOAD 247
    RTNMAP 247
    RTNSTATE 248
    RUN 248
    S (CMS Border Command) 249
    SAMGEN 249
    SAVEFD 250
    SAVENCP 250
    SCROLL 252
    SEGGEN 253
    SEGMENT ASSIGN 253
    SEGMENT LOAD 253
    SEGMENT PURGE 254
    SEGMENT RELEASE 254
    SEGMENT RESERVE 254
    SENDFILE 255
    SENTRIES 255
    SET 256
    SET ABBREV 257
    SET APL 257
    SET AUTOREAD 257
    SET BLIP 258
    SET BORDER 258
    SET CHARMODE 258
    SET CMSPF 259
    SET CMSTYPE 259
    SET COMDIR 260
    SET DOS 260
    SET DOSLNCNT 261
    SET DOSPART 261
    SET EXECTRAC 261

CMS (Conversational Monitoring System)
  commands *(continued)*
    SET FILEPOOL 262
    SET FILEWAIT 262
    SET FULLREAD 262
    SET FULLSCREEN 263
    SET IMESCAPE 263
    SET IMPCP 264
    SET IMPEX 264
    SET INPUT 264
    SET INSTSEG 265
    SET KEYPROTECT 265
    SET LANGUAGE 265
    SET LDRTBLS 266
    SET LINEND 266
    SET LOADAREA 267
    SET LOCATION 266
    SET LOGFILE 267
    SET NONDISP 267
    SET NONSHARE 268
    SET OUTPUT 268
    SET PROTECT 268
    SET RDYMSG 269
    SET REDTYPE 269
    SET RELPAGE 269
    SET REMOTE 270
    SET RESERVED 270
    SET SERVER 270
    SET STORECLR 271
    SET SYSNAME 271
    SET TEXT 271
    SET THRESHOLD 272
    SET TRANSLATE 272
    SET UPSI 272
    SET VSCREEN 273
    SET WINDOW 273
    SET WMPF 273
    SETKEY 280
    SETPRT 280
    SHOW WINDOW 280

CMS (Conversational Monitoring System)
  commands *(continued)*
    SIZE WINDOW 281
    SNTMAP 283
    SO 283
    SORT 283
    SPLOAD 284
    SSERV 287
    START 288
    STATE/STATEW (ESTATE/ESTATEW) 291
    SVCTRACE 293
    SYNONYM 294
    TAPE 296
    TAPEMAC 297
    TAPPDS 297
    TE 297
    TELL 298
    TS 302
    TXTLIB 303
    TYPE 303
    UPDATE 304
    UTILITY 306
    VALIDATE 306
    VMFAPPLY 308
    VMFASM 308
    VMFBLD 309
    VMFDOS 309
    VMFHASM 310
    VMFLKED 310
    VMFLOAD 311
    VMFMAC 311
    VMFMERGE 311
    VMFNLS 311
    VMFOVER 312
    VMFPLC2 313
    VMFREC 314
    VMFREMOVE 314
    VMFSETUP 315
    VMFTXT 315
    VMFZAP 315
CMS (Conversational Monitoring System)
  commands *(continued)*
    VRSIZE 316
    VSAMGEN 316
    VSEVSAM 316
    WAITREAD VSCREEN 317
    WAITT VSCREEN 317
    WRITE VSCREEN 318
    X (CMS Border Command) 318
    XEDIT 319
    XMITMSG.XMITMSG 361
    ZAP 362
    ZAPTEXT 362
    * 38
CMSBATCH command (CMS) 55
CMSGEND command (CMS) 55
CMSSERV (CMS command) 56
CMS, full-screen
    See full-screen CMS
command format notation 1
COMMANDS command (CP Class Any) 56
COMPARE command (CMS) 56
component code in message identifier 5
Control Program (CP)
    See CP (Control Program) commands
Conversational Monitoring System (CMS)
    See CMS (Conversational Monitoring
      System) commands
CONVERT command (IPCS) 57
CONVERT COMMANDS (CMS) 57
CONVIPCS (IPCS) 57
CONWAIT command (CMS) 58
COPYFILE command (CMS) 58
COUPLE command (CP class G) 59
CP 10
CP command privilege classes 35
CP command (CMS) 59
CP command (CP class any) 59
CP command (RSCS) 59

CP error message numbers 10
CP privileged commands 35
  description 35
  user and function 35
CP return codes 10
CP (Control Program) commands
  ACNT (class A) 40
  ADSTOP (class G) 40
  ATTACH (class B) 47
  ATTN (class G) 48
  ATTN (CMS) 48
  AUTOLOG (class A,B) 48
  BACKSPAC (class D) 49
  BEGIN (class G) 50
  CHANGE (class D) 51
  CHANGE (class G) 51
  CLOSE (class G) 54
  COMMANDS (class Any) 56
  COUPLE (class G) 59
  CP (class any) 59
  CPTRAP (class C) 61
  DCP (class C,E) 66
  DEFINE (class A,B) 69
  DEFINE (class G) 70
  DETACH (class B) 77
  DETACH (class G) 77
  DIAL (class any) 78
  DISABLE (class A,B) 82
  DISABLE (CMS) 82
  DISCONN (class any) 83
  DISPLAY (class G) 84
  DMCP (class C,E) 86
  DRAIN (class D) 88
  DUMP (class G) 90
  ECHO (class G) 90
  ENABLE (class A,B) 91
  EXTERNAL (class G) 113
  FLUSH (class D) 124
  FORCE (class A) 125
  FORCE (CMS) 125
CP (Control Program) commands *(continued)*
  FREE (class D) 127
  GIVE (class B) 132
  HALT (class A) 136
  HOLD (class D) 139
  INDICATE (class A) 142
  INDICATE (class E) 143
  INDICATE (class G) 143
  IPL (class G) 160
  LINK (class G) 164
  LOADBUF (class D) 167
  LOADVFCB (class G) 169
  LOCATE (class E) 169
  LOCK (class A) 169
  LOGOFF (class any) 170
  LOGON (class any) 170
  MESSAGE (class any) 174
  MESSAGE (class A,B) 173
  MIGRATE (class A) 174
  MONITOR (class A,E) 176
  MSGNOH (class A,B) 178
  NETWORK (class A) 181
  NETWORK (class B) 182
  NOTREADY (class G) 183
  ORDER (class D) 186
  ORDER (class G) 186
  PER (class A,B,C,D,E,F,G) 190
  PURGE (class D) 199
  PURGE (class G) 199
  QUERY (class A) 205
  QUERY (class B) 206
  QUERY (class C) 207
  QUERY (class D) 208
  QUERY (class E) 209
  QUERY (class F) 210
  QUERY (class G) 211
  QVM (class A) 216
  READY (class G) 219
  REPEAT (class D) 222
  REQUEST (class G) 223

CP (Control Program) commands *(continued)*
  RESET (class G) 224
  REWIND (class G) 225
  SAVESYS (class E) 250
  SCREEN (class G) 251
  SEND (class G) 255
  SET (class A) 274
  SET (class B) 275
  SET (class E) 276
  SET (class F) 276
  SET (class G) 277
  SHUTDOWN (class A) 281
  SLEEP (class any) 282
  SMSG (class G) 282
  SMSG (RSCS) 282
  SPACE (class D) 284
  SPMODE (class A) 284
  SPOOL (class G) 285
  SPTAPE (class D) 286
  START (class D) 287
  STCP (class C) 291
  STORE (class G) 293
  SYSTEM (class G) 294
  TAG (class G) 295
  TERMINAL (class G) 299
  TRACE (class G) 300
  TRANSFER (class D) 301
  TRANSFER (class G) 301
  UNLOCK (class A) 304
  VARY (class B) 307
  VMDUMP (class G) 307
  WARNING (class A,B) 317
  * (class any) 38
  #CP (class any) 38
CPQUERY command (RSCS) 60
CPTRAP command (CP class C) 61
CREATE ALIAS command (CMS) 62
CREATE DIRECTORY command (CMS) 62
CREATE LOCK command (CMS) 63
CREATE NAMEDEF command (CMS) 63
CSLGEN command (CMS) 64
CSLLIST command (CMS) 64
CURSOR VSCREEN command (CMS) 65

D

D (CMS Border Command) 65
DCP command (CP class C,E) 66
DCSSGEN command (CMS) 66
DDR command return codes (CMS) 10
DDR command (CMS) 67
DEBUG command (CMS) 68
DEFAULTS command (CMS) 68
DEFINE command (CP class A,B) 69
DEFINE command (CP class G) 70
DEFINE command (RSCS) 72
DEFINE VSCREEN command (CMS) 73
DEFINE WINDOW command (CMS) 73
DELETE ADMINISTRATOR command (CMS) 74
DELETE command (RSCS) 74
DELETE LINK command (TSAF) 74
DELETE LOCK command (CMS) 75
DELETE NAMEDEF command (CMS) 75
DELETE PUBLIC command (CMS) 75
DELETE USER command (CMS) 76
DELETE VSCREEN command (CMS) 76
DELETE WINDOW command (CMS) 76
DESBUF command (CMS) 77
DETACH command (CP class B) 77
DETACH command (CP class G) 77
DIAL command (CP class any) 78
DIRECT command (CMS) 78
  Account number 78
  ACIgroup 78
  CLass 78
  Console 78
  control statements
  Dedicate 78

DIRECT command (CMS) *(continued)*
  DIRectory 79
  INclude 79
  IPL 79
  IUCV 79
  Link 79
  Mdisk 79
  Option 79
  Profile 80
  SCReen 80
  SPEcial 80
  Spool 80
  User 80
dirid 31
DIRLIST command (CMS) 81
dirname 30
DISABLE command (CMS) 82
DISABLE command (CP class A,B) 82
DISABLE command (RSCS) 82
DISCARD command
  usage with DIRLIST command 81
  usage with FILELIST command 119
  usage with MACLIST command 172
  usage with PEEK command 189
  usage with RDRLIST command 218
DISCONN command (CP class any) 83
DISCONN command (RSCS) 83
DISK command (CMS) 83
DISKMAP command (CMS) 84
DISPLAY command (CP class G) 84
DLBL command (CMS) 85
DLBL command (GCS) 86
DMCP command (CP class C,E) 86
DOSGEN command (CMS) 87
DOSLIB command (CMS) 87
DOSLKED command (CMS) 87
DRAIN command (CP class D) 88
DRAIN command (RSCS) 88
DROP WINDOW command (CMS) 88
DROPBUF command (CMS) 89
DSERV command (CMS) 89
DUMP command (CP class G) 90

E

ECHO command (CP class G) 90
EDIT command (CMS) 90
ENABLE command (CMS) 90
ENABLE command (CP class A,B) 91
ENABLE command (RSCS) 91
ENROLL ADMINISTRATOR command (CMS) 91
ENROLL PUBLIC command (CMS) 92
ENROLL USER command (CMS) 92
ERASE command (CMS) 93
ESERV command (CMS) 94
ESTATE, ESTATEW command (CMS) 94
ETRACE command (CMS) 95
ETRACE command (GCS) 95
EXEC command (CMS) 96
EXEC command (RSCS) 96
EXEC 2 command (CMS) 97—109
  control statements 100
    &ARGS 100
    &BEGPRINT 100
    &BEGSTACK 101
    &BEGTYPE 100
    &BUFFER 101
    &CALL 101
    &CASE 101
    &COMMAND 102
    &DUMP 102
    &ERROR 102
    &EXIT 102
    &GOTO 102
    &IF 103
    &LOOP 103
    &PRESUME 103
    &PRINT 103
    &RTURN 104

- EXEC 2 command (CMS) *(continued)*
  - control statements *(continued)*
    - &SKIP 104
    - &STACK 104
    - &SUBCOMMAND 105
    - &TRACE 105
    - &TRUNC 105
    - &TYPE 103
    - &UPPER 105
  - predefined functions 106
    - &CONCAT 106
    - &CONCATENATION 106
    - &DATATYPE 106
    - &DIV 106
    - &DIVISION 106
    - &LEFT OF 106
    - &LENGTH OF 106
    - &LITERAL OF 107
    - &LOCATION OF 107
    - &MULT 107
    - &MULTIPLICATION OF 107
    - &PIECE 107
    - &POSITION OF 108
    - &RANGE OF 108
    - &RIGHT OF 108
    - &STRING OF 108
    - &SUBSTR 107
    - &TRANS 108
    - &TRANSLATION 108
    - &TRIM OF 109
    - &TYPE 106
    - &WORD OF 109
  - predefined variables 97
    - & 97
    - &ARGSTRING 97
    - &BLANK 98
    - &CMDSTRING 98
    - &COMLINE 98
    - &DATE 98
    - &DEPTH 98
    - &FILEMODE 98

- EXEC 2 command (CMS) *(continued)*
  - predefined variables *(continued)*
    - &FILENAME 99
    - &FILETYPE 99
    - &FROM 99
    - &LINE,&LINENUM 99
    - &LINK 99
    - &N,&INDEX 99
    - &RC,&RETCODE 100
    - &TIME 100
    - &1,&2,&n 97
    - &0 97
  - user-defined function 109
    - label 109
    - line-number OF 109
- EXECDROP command (CMS) 109
- EXECIO command (CMS) 109
- EXECLOAD command (CMS) 110
- EXECMAP command (CMS) 111
- EXECOS command (CMS) 112
- EXECSTAT command (CMS) 112
- EXECUPDT command (CMS) 112
- EXECUTE command
  - usage with CSLLIST command 64
  - usage with DIRLIST command 81
  - usage with FILELIST command 119
  - usage with MACLIST command 172
  - usage with RDRLIST command 218
- EXIT command (RSCS) 113
- EXPAND command (CMS) 113
- EXTERNAL command (CP class G) 113

F

- F (CMS Border Command) 114
- FETCH command (CMS) 114
- file identifiers 19
- file identifiers (CMS)
- file modes 29

file names 19
  pattern matching with 19
  special characters in 19
file types 19
file types, reserved (CMS) 20
FILEDEF command (CMS) 115
FILEDEF command (GCS) 117
FILELIST command (CMS) 118
  DISCARD command 119
  EXECUTE command 119
FILEPOOL BACKUP command (CMS) 119
FILEPOOL CLEANUP (CMS) 119
FILEPOOL FORMAT AUDIT (CMS) 120
FILEPOOL RESTORE (CMS) 120
FILESERV BACKUP (CMS) 120
FILESERV DEFAUDIT (CMS) 121
FILESERV DEFBACKUP (CMS) 121
FILESERV GENERATE (CMS) 121
FILESERV LIST (CMS) 122
FILESERV LOG (CMS) 122
FILESERV MINIDISK (CMS) 122
FILESERV MOVEUSER (CMS) 123
FILESERV REGENERATE (CMS) 123
FILESERV REORG (CMS) 123
FILESERV START (CMS) 124
FINIS command (CMS) 124
FLUSH command (CP class D) 124
FLUSH command (RSCS) 125
FORCE command (CMS) 125
FORCE command (CP class A) 125
FORCE command (RSCS) 126
FORMAT command (CMS) 126
format of VM/SP messages 5
FORMAT/ALLOCATE command (service aid) 127
FREE command (CP class D) 127
FREE command (RSCS) 128
full-screen CMS
  message routing defaults with 33
  virtual screens, default settings 32
FWDSPACE command (RSCS) 128

G

GCS (Group Control System) commands
  ACCESS 39
  DLBL 86
  ETRACE 95
  FILEDEF 117
  GDUMP 128
  GLOBAL 133
  HX 140
  ITRACE (CMS) 162
  ITRACE (GCS) 162
  LOADCMD 168
  OSRUN 187
  QUERY 212
  RELEASE 221
  REPLY 223
  SET 279
  * 38
GDUMP command (GCS) 128
GENDIRT command (CMS) 129
GENIMAGE command (CMS) 129
GENMOD command (CMS) 130
GENMSG command (CMS) 130
GENSERVE command (CMS) 131
GENTSAF command (CMS) 131
GEN3705 command (CMS) 131
GET VSCREEN command (CMS) 132
GIVE command (CP class B) 132
GLOBAL command (CMS) 132
GLOBAL command (GCS) 133
GLOBALV command (CMS) 134
GRANT ADMIN command (CMS) 135
GRANT AUTHORITY command (CMS) 135
GROUP command (CMS) 135
Group Control System (GCS)
  See GCS (Group Control System) commands

GTRACE command (RSCS) 136

H

H (CMS Border Command) 136
HALT command (CP class A) 136
HB command (CMS) 136
HELP command (CMS) 137
HELP facility, VM/SP 16
HELPCONV command (CMS) 138
HELP, VM/SP online 16
HI command (CMS) 138
HIDE WINDOW command (CMS) 138
HO command (CMS) 139
HOLD command (CP class D) 139
HOLD command (RSCS) 139
HT command (CMS) 139
HX command (CMS) 140
HX command (GCS) 140

I

IDENTIFY command (CMS) 140
IMAGELIB command (CMS) 141
IMAGEMOD command (CMS) 141
IMMCMD command (CMS) 141
INCLUDE command (CMS) 142
INDICATE command (CP class A) 142
INDICATE command (CP class E) 143
INDICATE command (CP class G) 143
INIT command (RSCS) 143
  Interactive Problem Control System (IPCS) commands
    See IPCS commands
Interactive Problem Control System
  See IPCS commands
Interactive Problem Control System command return codes 12
IPCS 12
IPCS command return codes 12
IPCS commands
  APAR 45
  CONVERT 57
  CONVIPCS 57
  IPCSDUMP 144
  IPCSPRT 144
    END 145
    FORMAT 145
    HELP 146
    HEX 146
    HX 146
    PROCESS 146
    QUIT 146
    SELect 147
    TIMESPAN 147
  IPCSSCAN 147
    - (decrement) 149
    Aregs 149
    ARIOBLOK 149
    BOTTOM 149
    C 150
    CHain 150
    CMS 150
    CMSPoint 150
    CORtable 150
    DISPLAY 151
    DOSPoint 151
    DOWN 151
    DUMPID 151
    END 151
    FDISPLAY 152
    FORMAT 152, 156
    G 152
    GDISPLAY 153
    HELP 153
    HEX 153, 156
    HX 153
    IDENTIFY 154
    IPCSMAP 154
    IUCV 154

IPCS commands *(continued)*
  IPCSSCAN *(continued)*
    Locate 154
    Locate Up 154
    LUNAME 155
    MAPA 155
    Mregs 155
    MRIOBLOK 155
    OSPoint 155
    PRINT 156
    QUIT 156
    Regs 156
    RIOblok 156
    Scroll 156
    Scrollu 156
    SELect 157
    SYMPtom 157
    TACtive 157
    TIME 157
    TLoadl 158
    TOP 158
    Trace 158
    TSab 158
    UP 159
    USERMAP 159
    VIOblok 159
    Vmblok 159
    VmLoadl 159
    (null line) 148
    + (increment) 149
    & 149
    &name 149
    ? 148
  MAP 173
  PRB 192
  PROB 193
  STAT 290
  TRAPFILE 302
IPCSDUMP command (IPCS) 144
IPCSPRT command (IPCS) 144
  subcommands 145—147
    END 145
    FORMAT 145
    HELP 146
    HEX 146
    HX 146
    PROCESS 146
    QUIT 146
    SELect 147
    TIMESPAN 147
IPCSSCAN command (IPCS) 147
  subcommands 148—159
    - (decrement) 149
    Aregs 149
    ARIOBLOK 149
    BOTTOM 149
    C 150
    CHain 150
    CMS 150
    CMSPoint 150
    CORtable 150
    DISPLAY 151
    DOSPoint 151
    DOWN 151
    DUMPID 151
    END 151
    FDISPLAY 152
    FORMAT 152, 156
    G 152
    GDISPLAY 153
    HELP 153
    HEX 153, 156
    HX 153
    IDENTIFY 154
    IPCSMAP 154
    IUCV 154
    Locate 154
    Locate Up 154
    LUNAME 155
    MAPA 155

IPCSSCAN command (IPCS) *(continued)*
- subcommands *(continued)*
  - MAPN 155
  - Mregs 155
  - MRIOBLOK 155
  - OSPoint 155
  - PRINT 156
  - QUIT 156
  - Regs 156
  - RIOblok 156
  - Scroll 156
  - Scrollu 156
  - SELect 157
  - SYMPtom 157
  - TACtive 157
  - TIME 157
  - TLoadl 158
  - TOP 158
  - Trace 158
  - TSab 158
  - UP 159
  - USERMAP 159
  - VIOblok 159
  - Vmblok 159
  - VmLoadl 159
  - (null line) 148
  - + (increment) 149
  - & 149
  - &name 149
  - ? 148

IPL command (CP class G) 160
ITASK command (CMS) 161
ITRACE command (CMS) 162
ITRACE command (GCS) 162

L

L (CMS Border Command) 162
LABELDEF command (CMS) 163
LANGGEN command (CMS) 163
LANGMERG command (CMS) 164
LINK command (CP class G) 164
LISTDIR command (CMS) 164
LISTDS command (CMS) 165
LISTFILE command (CMS) 165
LISTIO command (CMS) 166
LKED command (CMS) 166
LOAD command (CMS) 167
LOADBUF command (CP class D) 167
LOADCMD command (GCS) 168
LOADLIB command (CMS) 168
LOADMOD command (CMS) 168
LOADVFCB command (CP class G) 169
LOCATE command (CP class C,E) 169
LOCK command (CP class A) 169
LOGOFF command (CP class any) 170
LOGON command (CP class any) 170

M

M (CMS Border Command) 170
MACLIB command (CMS) 171
MACLIST command (CMS) 172
- DISCARD command 172
- EXECUTE command 172

MAKEBUF command (CMS) 172
MAP command (IPCS) 173
MAXIMIZE WINDOW command (CMS) 173
MESSAGE command (CP class Any) 174
MESSAGE command (CP class A,B) 173
message format (VM/SP) 4
message routing defaults with full-screen CMS 33
MIGRATE command (CP class A) 174
MINIMIZE WINDOW command (CMS) 174
MODIFY USER command (CMS) 175
MODMAP command (CMS) 175
module code in message identifier 5

MONITOR command (CP class A,E) 176
MOREHELP command (CMS) 177
MOVEFILE command (CMS) 177
MSG command (RSCS) 177
MSGNOH command (CP class A,B) 178

N

N (CMS Border Command) 178
NAMEFIND command (CMS) 179
NAMES command (CMS) 179
NCPDUMP command (CMS) 179
NETDATA command (CMS) 180
NETWORK command (CP class A) 181
NETWORK command (CP class B) 182
NETWORK command (RSCS) 183
notational conventions, command format 1
notation, command format 1
NOTE command (CMS) 183
NOTREADY command (CP class G) 183
NUCXDROP command (CMS) 184
NUCXLOAD command (CMS) 184
NUCXMAP command (CMS) 184

O

O (CMS Border Command) 185
online HELP facility, VM/SP 16
operations function type 36
OPTION command (CMS) 185
ORDER command (CP class D) 186
ORDER command (CP class G) 186
ORDER command (RSCS) 186
OSRUN command (CMS) 187
OSRUN command (GCS) 187

P

P (CMS Border Command) 188
PARSECMD command (CMS) 188
pattern matching with file names (CMS) 19
PEEK command (CMS) 189
  DISCARD command 189
PER command (CP class A,B,C,D,E,F,G) 190
POP WINDOW command (CMS) 191
PORT command (RSCS) 191
POSITION WINDOW command (CMS) 191
PRB command (IPCS) 192
prefix in message identifier 5
PRELOAD command (CMS) 192
PRINT command (CMS) 193
privilege classes, CP commands 35
PROB command (IPCS) 193
PROGMAP command (CMS) 194
PROGRAMMABLE OPERATOR (CMS) 195
  commands 195—197
    CMD 196
    FB 196
    FEEDBACK 196
    GET 196
    LGLOPR 196
    LOADTBL 196
    LOG 197
    QUERY 197
    SET 197
PSERV command (CMS) 198
PUNCH command (CMS) 198
PURGE command (CP class D) 199
PURGE command (CP class G) 199
PURGE command (RSCS) 200
PUT SCREEN command (CMS) 200
PUT VSCREEN command (CMS) 200

Q

QUERY command (CMS) 201
QUERY command (CP class A) 205
QUERY command (CP class B) 206
QUERY command (CP class C) 207

QUERY command (CP class D) 208
QUERY command (CP class E) 209
QUERY command (CP class F) 210
QUERY command (CP class G) 211
QUERY command (GCS) 212
QUERY command (RSCS) 214
QUERY command (TSAF) 213
QUERY DISABLE command (CMS) 214
QUERY FILEPOOL CONFLICT command (CMS) 215
QUERY FILEPOOL STATUS command (CMS) 215
QUERY LIMITS command (CMS) 216
QVM command (CP class A) 216

R

R (CMS Border Command) 217
RDR command (CMS) 217
RDRLIST command (CMS) 218
  DISCARD command 218
  EXECUTE command 218
READCARD command (CMS) 219
READY command (CP class G) 219
READY command (RSCS) 219
RECEIVE command (CMS) 220
RECONN command (RSCS) 220
REFRESH command (CMS) 220
RELEASE command (CMS) 221
RELEASE command (GCS) 221
RELOCATE command (CMS) 221
Remote Spooling Communications Subsystem (RSCS)
  See RSCS (Remote Spooling Communications Subsystem) commands
RENAME command (CMS) 222
REORDER command (RSCS) 222
REPEAT command (CP class D) 222
REPLY command (GCS) 223
REQUEST command (CP class G) 223
REROUTE command (RSCS) 223
RESERVE command (CMS) 224
reserved file type descriptions 20
reserved file types (CMS) 20
RESET command (CP class G) 224
RESTORE WINDOW command (CMS) 224
return codes
REVOKE ADMIN command (CMS) 225
REVOKE AUTHORITY command (CMS) 225
REWIND command (CP class G) 225
REXX language (CMS) 226
  REXX built-in functions 232—243
    ABBREV 232
    ABS 233
    Address 233
    Arg 233
    BITAND 233
    BITOR 233
    BITXOR 233
    CENTER 234
    CENTRE 234
    COMPARE 234
    COPIES 234
    C2D 234
    C2X 234
    DATATYPE 235
    DATE 235
    DELSTR 235
    DELWORD 235
    D2C 235
    D2X 236
    ERRORTEXT 236
    EXTERNALS 236
    FIND 236
    FORM 236
    FORMAT 237
    FUZZ 237
    INDEX 237
    INSERT 237
    JUSTIFY 237

REXX language (CMS) *(continued)*
  REXX built-in functions *(continued)*
    LASTPOS 238
    LEFT 238
    LENGTH 238
    LINESIZE 238
    MAX 238
    MIN 238
    OVERLAY 239
    POS 239
    QUEUED 239
    RANDOM 239
    REVERSE 239
    RIGHT 239
    SIGN 240
    SOURCELINE 240
    SPACE 240
    STRIP 240
    SUBSTR 240
    SUBWORD 241
    SYMBOL 241
    TIME 241
    TRACE 241
    TRANSLATE 241
    TRUNC 242
    USERID 242
    VALUE 242
    VERIFY 242
    WORD 242
    WORDINDEX 242
    WORDLENGTH 243
    WORDPOS 243
    WORDS 243
    XRANGE 243
    X2C 243
    X2D 243
  REXX instructions 226—232
    ADDRESS 226
    ARG 226
    CALL 226
    DO 227

REXX language (CMS) *(continued)*
  REXX instructions *(continued)*
    DROP 227
    EXIT 227
    IF-THEN-ELSE 227
    INTERPRET 228
    ITERATE 228
    LEAVE 228
    NOP 228
    NUMERIC DIGITS 229
    NUMERIC FORM 229
    NUMERIC FUZZ 229
    PROCEDURE 230
    PULL 230
    PUSH 230
    QUEUE 230
    RETURN 230
    SAY 231
    SELECT 231
    SIGNAL 231
    TRACE 232
    UPPER 232
  RSXYSFN package 244
    CMSFLAG 244
    CSL 244
    DIAG 244
    DIAGRC 244
    STORAGE 245
RO command (CMS) 245
ROUTE command (CMS) 245
ROUTE command (RSCS) 245
RSCS (Remote Spooling Communications Subsystem) commands
  BACKSPAC 49
  CHANGE 52
  CMD 54
  CP 59
  CPQUERY 60
  DEFINE 72
  DELETE 74

- RSCS (Remote Spooling Communications Subsystem) commands *(continued)*
  - DISABLE 82
  - DISCONN 83
  - DRAIN 88
  - ENABLE 91
  - EXEC 96
  - EXIT 113
  - FLUSH 125
  - FORCE 126
  - FREE 128
  - FWDSPACE 128
  - GTRACE 136
  - HOLD 139
  - INIT 143
  - MSG 177
  - NETWORK 183
  - ORDER 186
  - PORT 191
  - PURGE 200
  - QUERY 214
  - READY 219
  - RECONN 220
  - REORDER 222
  - REROUTE 223
  - ROUTE 245
  - SET 279
  - SHUTDOWN 281
  - START 289
  - STOP 292
  - TRACE 300
  - TRANSFER 301
  - * 38
- RSERV command (CMS) 246
- RT command (CMS) 248
- RTNDROP command (CMS) 246
- RTNLOAD command (CMS) 247
- RTNMAP command (CMS) 247
- RTNSTATE command (CMS) 248
- RUN command (CMS) 248
- RUNTSAF command (TSAF) 249

S

- S (CMS Border Command) 249
- SAMGEN command (CMS) 249
- SAVEFD command (CMS) 250
- SAVENCP command (CMS) 250
- SAVESYS command (CP class E) 250
- SCREEN command (CP class G) 251
- SCROLL command (CMS) 252
- SEGGEN command (CMS) 253
- SEGMENT ASSIGN command (CMS) 253
- SEGMENT LOAD command (CMS) 253
- SEGMENT PURGE command (CMS) 254
- SEGMENT RELEASE command (CMS) 254
- SEGMENT RESERVE command (CMS) 254
- SEND command (CP class G) 255
- SENDFILE command (CMS) 255
- SENTRIES command (CMS) 255
- service aid
  - FORMAT/ALLOCATE 127
- SET ABBREV command (CMS) 257
- SET APL command (CMS) 257
- SET AUTOREAD command (CMS) 257
- SET BLIP command (CMS) 258
- SET BORDER command (CMS) 258
- SET CHARMODE command (CMS) 258
- SET CMSPF command (CMS) 259
- SET CMSTYPE command (CMS) 259
- SET COMDIR command (CMS) 260
- SET command (CMS) 256
- SET command (CP class A) 274
- SET command (CP class B) 275
- SET command (CP class E) 276
- SET command (CP class F) 276
- SET command (CP class G) 277
- SET command (RSCS) 279

SET command (SET) 279
SET DOS command (CMS) 260
SET DOSLNCNT command (CMS) 261
SET DOSPART command (CMS) 261
SET ETRACE command (TSAF) 279
SET EXECTRAC command (CMS) 261
SET FILEPOOL command (CMS) 262
SET FILEWAIT command (CMS) 262
SET FULLREAD command (CMS) 262
SET FULLSCREEN command (CMS) 263
SET IMESCAPE command (CMS) 263
SET IMPCP command (CMS) 264
SET IMPEX command (CMS) 264
SET INPUT command (CMS) 264
SET INSTSEG command (CMS) 265
SET KEYPROTECT command (CMS) 265
SET LANGUAGE command (CMS) 265
SET LDRTBLS command (CMS) 266
SET LINEND command (CMS) 266
SET LOADAREA command (CMS) 267
SET LOCATION command (CMS) 266
SET LOGFILE command (CMS) 267
SET NONDISP command (CMS) 267
SET NONSHARE command (CMS) 268
SET OUTPUT command (CMS) 268
SET PROTECT command (CMS) 268
SET RDYMSG command (CMS) 269
SET REDTYPE command (CMS) 269
SET RELPAGE command (CMS) 269
SET REMOTE command (CMS) 270
SET RESERVED command (CMS) 270
SET SERVER command (CMS) 270
SET STORECLR command (CMS) 271
SET SYSNAME command (CMS) 271
SET TEXT command (CMS) 271
SET THRESHOLD command (CMS) 272
SET TRANSLATE command (CMS) 272
SET UPSI command (CMS) 272
SET VSCREEN command (CMS) 273
SET WINDOW command (CMS) 273
SET WMPF command (CMS) 273
SETKEY command (CMS) 280
SETPRT command (CMS) 280
severity codes in message identifier 5
SFS (Shared File System) Directories, naming
  See Shared File System (SFS) Directories, naming
Shared File System (SFS) Directories, naming
  specifying dirid 31
SHOW WINDOW command (CMS) 280
SHUTDOWN command (CP class A) 281
SHUTDOWN command (RSCS) 281
SIZE WINDOW command (CMS) 281
SLEEP command (CP class any) 282
SMSG command (CP class G) 282
SMSG command (RSCS) 282
SNTMAP command (CMS) 283
SO command (CMS) 283
SORT command (CMS) 283
SPACE command (CP class D) 284
SPLOAD command (CMS) 284
SPMODE command (CP class A) 284
SPOOL command (CP class G) 285
SPTAPE command (CP class D) 286
SSERV command (CMS) 287
START command (CMS) 288
START command (CP class D) 287
START command (RSCS) 289
STAT command (IPCS) 290
STATE/STATEW (ESTATE/ESTATEW) command (CMS) 291
STCP command (CP class C) 291
STOP command (CMS) 292
STOP command (RSCS) 292
STOP TSAF command (TSAF) 292
STORE command (CP class G) 293
SVCTRACE command (CMS) 293
SYNONYM command (CMS) 294

SYSTEM command (CP class G) 294
System Product Editor command return codes 12

T

TAG command (CP class G) 295
TAPE command (CMS) 296
TAPEMAC command (CMS) 297
TAPPDS command (CMS) 297
TE command (CMS) 297
TELL command (CMS) 298
TERMINAL command (CP class G) 299
TRACE command (CP class G) 300
TRACE command (RSCS) 300
TRANSFER command (CP class D) 301
TRANSFER command (CP class G) 301
TRANSFER command (RSCS) 301
Transparent Services Access Facility (TSAF)
  See TSAF (Transparent Services Access Facility) commands
TRAPFILE command (IPCS) 302
TS command (CMS) 302
TSAF (Transparent Services Access Facility) commands
  ADD LINK 40
  DELETE LINK 74
  QUERY 213
  RUNTSAF 249
  SET ETRACE 279
  STOP TSAF 292
TXTLIB command (CMS) 303
type code in message identifier 5
TYPE command (CMS) 303

U

UNLOCK command (CP class A) 304
UPDATE command (CMS) 304
  Comment 305

UPDATE command (CMS) *(continued)*
  control statements
  Delete 304
  Insert 304
  Replace 305
  Sequence 304
UTILITY command (CMS) 306

V

VALIDATE command (CMS) 306
VARY command (CP class B) 307
virtual screens, default settings (full-screen CMS) 32
VMDUMP command (CP class G) 307
VMFAPPLY command (CMS) 308
VMFASM command (CMS) 308
VMFBLD command (CMS) 309
VMFDOS command (CMS) 309
VMFHASM command (CMS) 310
VMFLKED command (CMS) 310
VMFLOAD command (CMS) 311
VMFMAC command (CMS) 311
VMFMERGE command (CMS) 311
VMFNLS command (CMS) 311
VMFOVER command (CMS) 312
VMFPLC2 command (CMS) 313
VMFREC command (CMS) 314
VMFREMOV command (CMS) 314
VMFSETUP command (CMS) 315
VMFTXT command (CMS) 315
VMFZAP command (CMS) 315
VRSIZE command (CMS) 316
VSAMGEN command (CMS) 316
VSEVSAM command (CMS) 316

W

WAITREAD VSCREEN command (CMS) 317

WAITT VSCREEN command (CMS) 317
WARNING command (CP class A,B) 317
windows, default settings
WRITE VSCREEN command (CMS) 318

X

X (CMS Border Command) 318
XEDIT 12
XEDIT command return codes 12
XEDIT command (CMS) 319—361
  prefix subcommands 357
    A 357
    C 357
    D 358
    E 358
    F 358
    I 358
    M 358
    P 359
    S 359
    SCALE 359
    SI 359
    TABL 359
    X 360
    .xxxx 360
    < 360, 361
    /[n] 360
    > 360
    ″ 361
  subcommands and macros 319
    Add 320
    ALL 320
    ALter 320
    Backward 321
    Bottom 321
    CANcel 321
    CAppend 321
    CDelete 321
    CFirst 321
    Change 322

XEDIT command (CMS) *(continued)*
  subcommands and macros *(continued)*
    CInsert 322
    CLAst 322
    CLocate 322
    CMS 322
    CMSG 323
    COMMAND 323
    COMpress 323
    COpy 323
    COUnt 323
    COVerlay 324
    CP 324
    CREplace 324
    CURsor 324
    DELete 324
    Down 325
    DUPlicat 325
    EMSG 325
    EXPand 325
    EXTract 326
    FILE 326
    Find 326
    FINDUp 327
    FOrward 327
    FUp 327
    GET 327
    Help 327
    HEXType 327
    Input 328
    Join 328
    LEft 328
    LOAD 329
    Locate 329
    LOWercas 329
    LPrefix 329
    MACRO 330
    MErge 330
    MOdify 330
    MOve 331
    MSG 331

XEDIT command (CMS) *(continued)*
  subcommands and macros *(continued)*
    Next 331
    NFind 331
    NFindUp 331
    NFU 331
    Overlay 332
    PARSE 332
    POWerinp 332
    PREServe 333
    PURge 333
    PUT 333
    PUTD 334
    Query 335
    QUIT 336
    READ 336
    RECover 336
    REFRESH 336
    RENum 336
    REPEat 337
    Replace 337
    RESet 337
    RESTore 337
    RGTLEFT 337
    RIght 338
    SAVE 338
    SCHANGE 338
    SET 339—353
    SHift 353
    SI 354
    SORT 354
    SOS 354
    SPlit 355
    SPLTJOIN 355
    STAck 355
    STATus 355
    TOP 356
    TRAnsfer 356
    Type 356
    Up 356
    UPPercas 357

XEDIT command (CMS) *(continued)*
  subcommands and macros *(continued)*
    Xedit 357
    & 319
    ? 320
    = 320
XMITMSG command (CMS) 361

Z

ZAP command (CMS) 362
ZAPTEXT command (CMS) 362

Special Characters

* command (CMS) 38
* command (CP class any) 38
* command (GCS) 38
* command (RSCS) 38
#CP command (CP class any) 38
#WM command
  usage with SET FULLSCREEN command 263

Program Number
5664-167

File Number
S370/4300-36

Printed in U.S.A.

VM/SP
Quick Reference
Order No. SX20-4400-05

READER'S
COMMENT
FORM

Is there anything you especially like or dislike about this book? Feel free to comment on specific errors or omissions, accuracy, organization, or completeness of this book.

IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you, and all such information will be considered nonconfidential.

Note: Do not use this form to report system problems or to request copies of publications. Instead, contact your IBM representative or the IBM branch office serving you.

Would you like a reply? **___YES** **___NO**

Please print your name, company name, and address:

____

____

____

____

IBM Branch Office serving you: ____

Thank you for your cooperation. You can either mail this form directly to us or give this form to an IBM representative who will forward it to us.

SX20-4400-05

Fold and tape Please Do Not Staple Fold and tape

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL
FIRST-CLASS MAIL PERMIT NO. 40 ARMONK, NY

POSTAGE WILL BE PAID BY ADDRESSEE:

IBM

INTERNATIONAL BUSINESS MACHINES CORPORATION
DEPARTMENT G60
PO BOX 6
ENDICOTT NY 13760-9987

Fold and tape Please Do Not Staple Fold and tape

IBM®

Program Number
5664-167

File Number
S370/4300-36

2 - DUPE OK

SX20-4400-05

Printed in U.S.A.