

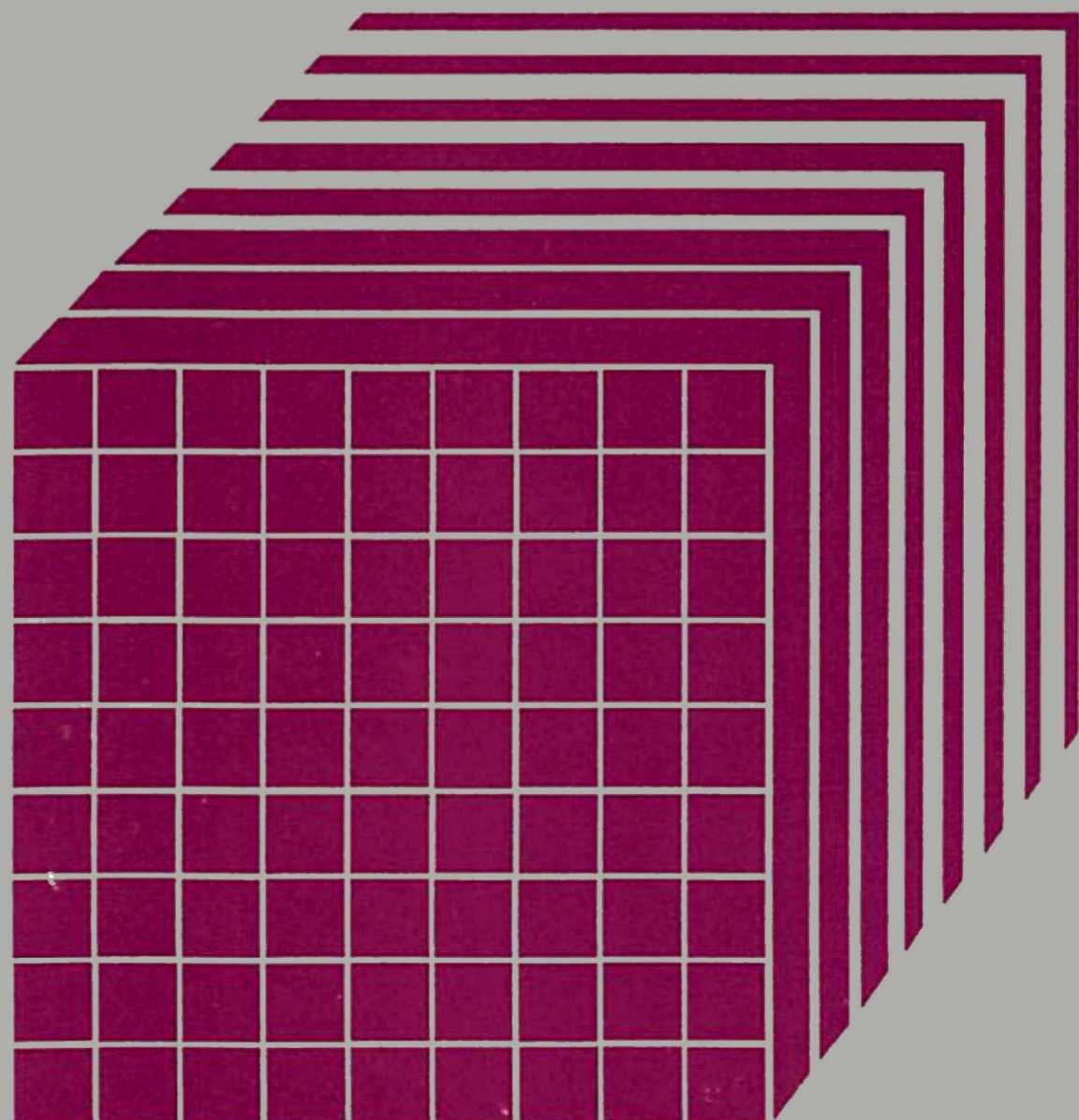


Virtual Machine/ System Product

Quick Reference

Release 5

SX20-4400-4



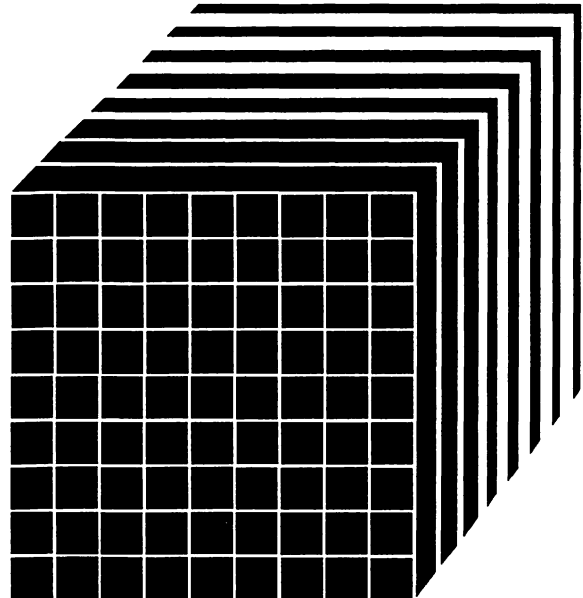


Virtual Machine/ System Product

Quick Reference

Release 5

SX20-4400-4



Fifth Edition (December 1986)

This edition, SX20-4400-4, is a major revision of SX20-4400-3. It applies to Release 5 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, and 4300 Processors Bibliography*, GC20-0001, for the editions that are applicable and current.

Summary of Changes

For a list of changes, see "Summary of Changes" on page 273.

Changes or additions to the text and illustrations are indicated by a vertical line to the left of the change.

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

Ordering Publications

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

A form for reader's comments is provided at the back of this publication. If the form has been removed, comments may be addressed to IBM Corporation, Information Development, Dept. G60, P.O. Box 6, Endicott, 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.



Preface

This publication is a quick reference for all users of VM/SP, from general users to experienced system programmers. It is the one book in the VM/SP library that contains all the commands (CP, CMS, GCS, TSAF, IPCS, and RSCS) that are documented in the rest of the library. The coverage is complete so that experienced programmers can find infrequently used command formats, such as privileged CP commands, without delving into the entire library. Since all commands are clearly labeled, each type of user can quickly find the particular one he needs and ignore the others. The addition of an index and running headings make this an easy task.

New users who need more guidance will find the *VM/SP CMS Primer* and the *VM/SP CMS Primer for Line-Oriented Terminals* most helpful. These primers give you detailed, easy-to-follow directions on how to do your work using the VM/SP system.

The section "Using VM/SP System" is a memory jogger describing how to access the VM/SP system and control terminal operations, how to create files and manipulate data, and how to code and debug programs using CMS.

The section "Summary of VM/SP Commands" is an alphameric listing of all the VM/SP commands and service aids. For each command a brief description of its function is included, as well as the complete command format. The section also describes the syntax conventions used in the command formats and summarizes the CP privilege classes.

This publication is part of a set of reference summaries that may be ordered as a group under Order No. SBOF3242.

Terminology

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

Using This Publication

This publication includes commands and operands that pertain to the VM/SP System Control Program (5664-167).

See the Bibliography for lists of the VM/SP publications that should be used in conjunction with the first sections of this publication.

Contents

Chapter 1. Using the VM/SP System	1
Getting Started	2
Terminal Operating Procedures	3
Logging On	3
Logon Exceptions	6
Logging Off	8
Input Conventions	10
Chapter 2. Using CMS	15
How To Initial Program Load (IPL) CMS	16
System Product Editor	16
EXEC 2 Interpreter	17
System Product Interpreter	18
CMS Session Services	19
Enhanced Connectivity Facilities on VM/SP	21
Example of CMS Program Development Facilities	22
Creating an Assembler Language Source File	22
Assembling a Source File	24
Correcting Errors	25
Creating a Load Module	26
Testing and Correcting a Program	26
Erasing Unwanted Files	28
CMS File System	29
Printing, Sending, and Reading Files	29
CMS Fileids	32
Return Codes	39
CMS Return Codes	39
CP DIRECT Command Return Codes	41
CMS DDR Command Return Codes	42
System Product Editor (XEDIT) Command Return Codes	42
IPCS Commands Return Codes	45
Example of a Return Code from a CP Command	45
Chapter 3. Summary of VM/SP Commands and Service Aids	47

Notational Conventions	48
CP Privilege Classes	50
Chapter 4. CP, CMS, GCS, TSAF, RSCS, and IPCS Commands	53
Summary of Changes	273
The HELP Facility	273
NLS Parser Summary	273
Enhancements for EXECs in Storage	274
Advanced Printer Subsystem Support	274
Transparent Services Access Facility	274
Enhanced 3270 Usability	275
New Commands for Release 5	275
TSAF Commands	276
Changed Commands for Release 5	276
Bibliography	277
Prerequisite Publications	277
Corequisite Publications	278
Index	281



Figures

1.	VM/SP Default Logical Line Editing Characters and Their Use	11
2.	Summary of Status Action on Display Terminals	12
3.	Sample Assembler Language Program Used for Creating a Source File	23
4.	Reserved Filetypes	34
5.	Return Codes Produced by CMS	39
6.	Return Codes Produced by the CP DIRECT Command	41
7.	Return Codes Produced by the CMS DDR Command	42
8.	Return Codes Produced by the XEDIT Command	42
9.	Return Codes Produced by the IPCS Commands	45
10.	Syntax Abbreviations Used	49

Chapter 1. Using the VM/SP System

IBM Virtual Machine/System Product (VM/SP) is a system control program that controls virtual machines. A virtual machine is a functional equivalent of a real machine that you control from your terminal using a command language. VM/SP builds and maintains, for each user, a virtual System/370 machine from a predefined configuration.

The virtual machine configuration includes components corresponding to a real System/370:

1. A virtual operator's console
2. Virtual storage
3. A virtual processor
4. Virtual channels and I/O devices.

However, since the virtual machines are simulated, their configurations may differ from that of the real machine and from each other. For example, the real machine may have 512K bytes of real storage and eight real disk drives, while a virtual machine may have 768K bytes of virtual storage and two virtual disk drives.

The command languages correspond to the components of VM/SP. CP (Control Program) controls the virtual machine while CMS (Conversational Monitor System) is the conversational operating system designed to run under CP. You use CP commands to communicate with the Control Program and control the devices attached to your virtual machine. CP commands can be entered from either the CP or CMS environment; but CMS commands cannot be entered while in the CP environment. Thus, the CP and CMS

command languages are, for practical purposes, a single, integrated command language for CMS users.

VM/SP makes work easier with its ability to create, change, store, copy, send and print data files.

New users will find the *VM/SP CMS Primer* most helpful. It teaches you how to do your work using the VM/SP system and a video display terminal. Likewise, the *CMS Primer for Line-Oriented Terminals* is for new users who have line-oriented terminals.

Getting Started

Before you can start using VM/SP, you must have:

- A user identification and password. Your user identification (userid) identifies you to VM/SP. Your password is checked when you log on, and protects your virtual machine from unauthorized use. The VM/SP system operations group usually assigns userids and passwords.
- A virtual machine defined for your use. The virtual machine definition should include all the devices you expect to use, for example:

- A console
- Card reader
- Card punch
- Printer
- Disk space.

The VM/SP system operations group usually defines your virtual machine configuration.

- Properly formatted disk space. The VM/SP system operations group usually formats disk space, but you can format your own disk space using the FORMAT command. See the *VM/SP CMS User's Guide* for details on formatting your disk space.

Once you have your userid and password, you can communicate with the VM/SP system from a terminal such as an IBM 3270 Information Display System Terminal, IBM 2741 typewriter-like terminal, or IBM 1050 Data Communication System (or equivalent). Depending on your terminal, you can either dial the central VM/SP computer, or you are directly connected to it. For a description of the communication procedures for each type of terminal, see the *VM/SP Terminal Reference*.

Terminal Operating Procedures

For a more detailed description of specific terminal models with their specific messages, see the *VM/SP Terminal Reference*.

Logging On

First, establish communication with VM/SP by turning on your terminal's power. VM/SP sends a Virtual Machine/System Product message and/or logo to your terminal.

Note: If your terminal is not a 3270-type, use the logon procedure outlined in the Logon Exceptions section.

If your terminal is a 3270-type, with a screen size of 20x80 or larger, you may log on directly from the logo screen.

Below the actual VM/SP logo you will find instructions for filling in your userid and password. For security reasons, your password will not appear when you enter it.

Three input lines follow the instructions and are labeled USERID, PASSWORD, and COMMAND. The cursor is placed at the input line for USERID.

You may type your userid and password in the USERID and PASSWORD input areas and press enter. If all the information is entered correctly, the logo is cleared from the screen, no further prompts will appear, and you will be logged on to the system. If an invalid userid or password is entered, the logo is cleared from the screen, and the following message and prompt will appear:

```
DMKLOG050E LOGON unsuccessful - incorrect password
```

or

```
DMKLOG053E userid not in CP directory
```

Enter one of the following commands:

```
LOGON userid           (Example: LOGON VMUSER1)
DIAL userid           (Example: DIAL VMUSER2)
MSG userid message    (Example: MSG VMUSER GOOD MORNING)
LOGOFF
```

If you enter only your PASSWORD in the input area or if your USERID, as entered, contains one or more blanks (V MUSER1), the following error will be issued, followed by the LOGON prompts:

```
DMKLOG050E userid not in CP directory
```

```
Enter password (It will not appear when typed) :
```

You may also enter your userid in the USERID input area without your password or enter the LOGON command, followed by your userid, in the COMMAND input area. The following prompt will appear:

| Enter password (It will not appear when typed) :

| **If you have entered the information correctly, the logo is cleared**
| **from the screen and you will be logged on to the system.**

Logon Exceptions

If your terminal is not a 3270-type terminal, use the following logon procedures once you have established communication with VM/SP.

Press the Enter key (or Clear key, or Attn key, or equivalent). Then the system prompts you:

Enter one of the following commands:

LOGON userid	(Example: LOGON VMUSER)
DIAL userid	(Example: DIAL VMUSER2)
MSG userid message	(Example: MSG VMUSER2 GOOD MORNING)
LOGOFF	

Identify yourself to the system by entering one of the above choices. For example:

```
l userid      (short for LOGON userid)
```

Then press the Enter (or Return) key.

If VM/SP accepts your userid, it responds with:

```
Enter password (it will not appear when typed):
```

Then type your password and press the Enter key. Each time you press Enter the command you typed is sent to the system.

Note: Not seeing your password as you type it is a security measure that prevents other people from learning your password and using your system without authorization. For more information on this masking technique or using the print inhibiting feature, see the *VM/SP Terminal Reference*.

If you make an error during the logon procedure the system will give you an error message and you **must** start the logon procedure from the beginning by entering "l userid" again.

For example, if you type your password incorrectly VM/SP will give you this error message:

```
DMKLOG050E LOGON unsuccessful--incorrect password
```

Enter one of the following commands:

Or if the userid and password you enter are valid, but someone else has already logged on with this userid, VM/SP issues this message:

```
DMKLOG054E Already logged on devtype rdev
```

Enter one of the following commands:

If you want to find out why the userid you just entered is in use, issue the MSG command to send a message to the operator or to the other user. You should either logon with another userid (if another userid is reserved for your use) or try again later.

If you are currently logged on when you issue an invalid command, you will receive the message:

```
Unknown CP command: name
```

Once you have successfully logged on and entered the system, VM/SP replies with the following kinds of messages:

- Informational messages regarding linkage status of minidisks attached to your virtual machine
- LOGMSG setup time (the time and date that the operator created a log message)
- The log message line that the operator created
- The FILES message (which tells you if you have any virtual reader, print, or punch files)
- A LOGON message, such as:

LOGON AT 11:24:35 EST THURSDAY mm/dd/yy

Now you can start using the virtual machine that you have set up for your userid.

Logging Off

When you want to end your terminal session, you do so by logging off the VM/SP control program (CP). Even if you are in CMS mode, you need only enter the command:

```
log      (short for LOGOFF)
```

and press the ENTER key. VM/SP responds with:

```
CONNECT=hh:mm:ss VIRTCPU=mmm:ss.ss TOTCPU=mmm:ss.ss  
LOGOFF AT hh:mm:ss zzz weekday mm/dd/yy
```

and the connection with VM/SP terminates. The connect time is in hours, minutes, and seconds. The system displays the use of the virtual processor and total processor in minutes, seconds, and hundredths of a second. Pressing the appropriate key on your terminal causes the logo to appear on the terminal in preparation for the next session. Only when the logoff procedure is completed should you turn off terminal power.

Note: If you logged off over a dialed line, you may specify that the communication line be left connected so that another user can logon immediately from the same terminal. When you issue:

```
log hold      (short for LOGOFF HOLD)
```

the next user can then logon to VM/SP without having to reestablish a line connection.

Disconnecting Your Terminal

There are situations when it is to your advantage to **disconnect** your terminal from your virtual machine.

The disconnect function is described in *VM/SP Terminal Reference* and the *CP Command Reference*.

Logoff and Security

When telecommunication line failures affect the system, VM/SP places your virtual machine in a DISCONNECT status for 15 minutes before it automatically logs you off the system.

If you turn off your terminal instead of issuing the LOGOFF command, your virtual machine is still logged on until:

- CP attempts to write data to your terminal while it is turned off, or
- Your terminal is turned on again.

If this happens, VM/SP places your virtual machine in DISCONNECT status for 15 minutes (unless you reconnect), and then logs you off the system. Even if another user turns on the same terminal he cannot use your virtual machine; he must log on using his own userid. Your virtual machine can be reconnected only by the normal logon procedure, requiring your password identification.

Note: The same sequence of events occurs:

- If your terminal is physically disconnected from the control unit, or
- If, on some terminals, the security key is turned to the locked position and then turned back to the unlocked position, or

- When, on some display terminals, the unit is switched to TEST mode and then back.

Input Conventions

You may enter data in either uppercase or lowercase. The examples in this book use lowercase characters for user entries and uppercase characters for system responses.

To correct typing errors on a full screen display terminal you can simply type over the data or use the Delete key for character deletion.

To correct typing errors in entering commands on a line-editing terminal, VM/SP assigns logical line-editing functions to four special characters. These characters allow you to easily correct mistakes as you enter data.

Using Logical Line Editing Characters

If you make a typing error when entering a command on a line-editing terminal, you can correct it by using one of four special characters. These four “logical line-editing” characters are more useful with typewriter-like terminals. (which do not have cursor-controlled line-editing) than with display terminals, but are valid for use with both. The characters listed in Figure 1 are the default characters in the system. They can be changed to others if your terminal keyboard does not have these characters. You can define some other infrequently used keyboard character to perform these editing functions, by using the CP TERMINAL command. For details on using the CP TERMINAL command to change default values, see *VM/SP CP Command Reference*.

Character	Function	Usage and Result Obtained
@	Logical Character Delete deletes the preceding character(s).	abc#@@ results in ab abc@d results in abd ç@def results in def abc@@@ deletes the whole line
#	Logical Line End divides a single physical input line into several logical input lines.	down 1#type 1#top results in execution of the commands as though entered: down 1 type 1 top
ç	Logical Line Delete deletes the preceding <i>logical</i> input line.	abcç results in abc#defç results in abc abc#def#ç results in abc def abc#def#çghi results in abc defghi Note: The logical line delete symbol (ç) deletes all typed characters back to and including the previous logical line end symbol (#). Use the logical line delete symbol when you have made several errors in a line.
"	Logical Escape accepts the following @, #, or ç character as data, not as a logical line-editing character.	abc"çd results in abcçd ""abc"" results in "abc" Note: The editor ignores any quotation mark ("") that appears as the last character of a line.

Figure 1. VM/SP Default Logical Line Editing Characters and Their Use

Line Length

For all VM/SP commands, input line length is restricted by the physical limitations of the terminal device, or by the default record length. Lines exceeding the maximum number of characters (including blanks, backspaces, underscores, the line-editing characters, and the tab character) are truncated to that line length value.

Line Termination

An input data line from an IBM 2741 Communications Terminal is transmitted to the processor by pressing the Return key. The same function is performed on the 3270 Information Display System Terminal by pressing the Enter key. Other terminals have similar line termination keys. As you do work on your terminal, the lower right corner of the screen will display various status notices. This

tells you what is happening in the system at the present time. This figure shows you how to change the status.

If the initial Status was:	and you pressed	(any data or a command)?	VM/SP does this:	and the Status becomes:
RUNNING (CP mode)	ENTER	no	Enters console function mode.	VM READ
		yes	Processes console function.	RUNNING
	CANCEL	don't care	Clears the Output Area.	RUNNING
RUNNING (VM mode)	ENTER	no	Attention pending. VM running.	RUNNING ¹
		yes	Attention pending. VM running, data stacked.	RUNNING ²
	CANCEL	don't care	Clears the Output Area.	RUNNING ²
MORE...	ENTER	no	Holds the Output Area display.	HOLDING
		yes	Same as for RUNNING (in CP or VM mode).	MORE...
	CANCEL	don't care	Clears the Output Area, continues output.	RUNNING
HOLDING	ENTER	no	Holds the Output Area display	MORE...
		yes	Same as for RUNNING (in CP or VM mode).	HOLDING
	CANCEL	don't care	Clears the Output Area, continues output.	RUNNING ³
CP READ	ENTER	no	"Null" line return.	RUNNING ⁴
		yes	Reads the data or command.	RUNNING
	CANCEL	don't care	Clears the Output Area.	CP READ
VM READ	ENTER	no	"Null" line, VM continues running	RUNNING ⁵
		yes	Reads data or command, VM continues running	RUNNING
	CANCEL	don't care	Clears the Output Area. If APL is on, interrupts the virtual machine.	VM READ

Notes:

- ¹ The status shown is **RUNNING**, however, the virtual machine should respond to the ATTN with a read, whereupon the status goes to **VM READ**
- ² If a data buffer is already stacked for a virtual machine, the terminal displays **NOT ACCEPTED** status before returning to the **RUNNING** status

Figure 2 (Part 1 of 2). Summary of Status Action on Display Terminals

- 3 If you are running with `TERMINAL MODE CP` (the default for the primary system operator), an attention return also occurs that cancels the function. You can use this to terminate certain `QUERY` or `DISPLAY` functions on consoles that do not have a PA1 key.
- 4 Unless you are the VM/SP primary system operator or are running with the `SET RUN ON` option, the status returns to `CP READ` for another console function if the previous read was for a console function.
- 5 The status remains `VM READ` if you have `SET AUTOREAD ON`.

Figure 2 (Part 2 of 2). Summary of Status Action on Display Terminals



Chapter 2. Using CMS

The CMS (Conversational Monitor System) component of VM/SP allows you to create, compile, execute and test programs, and to create and manage data files.

Before you can use CMS, you must do the following:

- Logon with a valid user identification and password. The user identification should have a virtual machine defined for its use.
- Initial program load (IPL) the CMS system by specifying the name of the CMS system or the device address of the CMS system disk.
- Have disk space available that is formatted for use by CMS.

The logging on procedure is discussed in "Terminal Operating Procedures" on page 3. The IPL procedure is described in this section. Disk formatting is detailed in the *VM/SP CMS User's Guide*.

How To Initial Program Load (IPL) CMS

If your system does not have “automatic IPL,” you need to load CMS into your virtual machine by entering a command. Type:

```
ipl cms
```

```
- or -
```

```
ipl 190
```

Press enter. VM/SP responds with a ready message which usually looks like:

```
Ready; T=7.36/19.87 09:26:12
```

If you have properly formatted disk space available you can now use the facilities of CMS:

System Product Editor

The part of the VM/SP system that helps you manipulate files is called the System Product Editor, invoked by the XEDIT command. Its enhancements include:

- Full screen editing
- Multiple views of the same file
- Multiple files viewed on the same screen
- Addition of color and extended highlighting
- Ability to issue selected commands directly from the displayed line

- Ability to define screen formats with flexibility, including splitting the screen vertically
- Extended use of the prefix area
- Extended string search to facilitate text processing
- Column pointer for intraline-editing
- Line splitting and/or joining functions
- Macro writing capabilities to expand the basic subcommand language and to tailor the language to your own application.

For a thorough understanding of the System Product Editor and its diverse uses, see *VM/SP System Product Editor User's Guide* and *VM/SP System Product Editor Command and Macro Reference*.

EXEC 2 Interpreter

An EXEC is a CMS function that allows you to create new commands by setting up frequently used CP or CMS commands, together with conditional branching facilities. This eliminates the repetitious re-keying of those commands. The sequence of commands is executed when you enter the filename of the EXEC. Coexisting with the EXEC processor currently in use, the EXEC 2 interpreter features:

- Acceptance of 255-character words
- Issuance of commands to either CMS or to specified other subcommand environments
- String manipulation functions

- Arithmetic functions such as multiplication and division
- Debugging facilities
- Support for user-defined functions and subroutines.

For a thorough understanding of the system product EXEC 2 interpreter, see *VM/SP EXEC 2 Reference*.

System Product Interpreter

The System Product Interpreter is an interpretive command and macro processor. Everyone, from experienced programmers to novices, may use the System Product Interpreter to create and execute EXEC-type command procedures and programs. It is easy to learn and use. EXEC procedures allow you to create new commands by setting up frequently used commands, together with conditional branching, to eliminate the repetitious re-keying of those commands. The System Product Interpreter's functions are written in a high-level language, similar to PL/I, known as the Restructured Extended Executor (REXX) language.

Some advantages of using the System Product Interpreter are that it:

- Uses a general-purpose high-level language, REXX.
- Supports structured programming concepts.
- Has many built-in functions.
- Accepts programs written in mixed case (which makes them easier to read).
- Has extensive mathematical capabilities, such as decimal, exponential, and scientific arithmetic.

- Uses a full set of arithmetic, character, and logical operators with algebraic precedence and parentheses.
- Accepts programs in free format.
- Makes XEDIT macros and system EXECs easier to maintain. It easily handles subroutine calls to other EXECs, modules, or internal routines.

The System Product Interpreter coexists with the CMS EXEC and EXEC 2 processors. It is functionally a superset of CMS EXEC and EXEC 2, but uses a completely different language and syntax.

You can find complete information about the System Product Interpreter and REXX in the *VM/SP System Product Interpreter User's Guide* and *VM/SP System Product Interpreter Reference*.

CMS Session Services

This support improves the usability of VM/SP on 3270-type display terminals. It includes:

1. Window functions for the CMS user
2. Full-screen CMS

The support also provides functions for:

- Defining and deleting windows
- Positioning windows anywhere on the screen
- Overlaying one window with another
- Scrolling backward and forward through data in a window

- Changing color, highlighting, and other characteristics of virtual screens
- Writing data into virtual screens
- Logging data in a CMS file

Full-screen CMS has the following characteristics:

- Routing VM output and messages to the appropriate windows
- Using extended highlighting for CMS output
- Defining CMS PF keys
- Controlling the display of messages

You can find complete information on this support and its functions in the *VM/SP CMS User's Guide* and the *VM/SP CMS Command Reference*.

Enhanced Connectivity Facilities on VM/SP

Enhanced Connectivity Facilities on VM/SP is part of IBM System/370 to IBM Personal Computer Enhanced Connectivity Facilities.

Enhanced Connectivity Facilities on VM/SP provides:

- A means for VM/SP to communicate with your work station (for example, IBM Personal Computer).

With a communications program running on your work station, you can enter the CMS command, CMSSERV, to start communications between VM/SP and your work station. With this, you have access to the services of the IBM System/370 to IBM Personal Computer Enhanced Connectivity Facilities.

- The Server-Requester Programming Interface (SRPI).

If you're an application programmer, you can write server programs for VM/SP that use the SRPI. A companion requester program, typically on the work station, can then ask the server to perform needed functions on VM/SP and pass the results back to the requester.

The SRPI has five subcommands (ADDENTRY, DELENTY, GETREQ, SENDREQ, and SETREPLY) that you can use when writing server programs.

You can find more information on how you can use the services of the IBM System/370 to IBM Personal Computer Enhanced Connectivity Facilities in the *VM/SP Introduction* or the *Introduction to IBM System/370 to IBM Personal Computer Enhanced Connectivity Facilities*, GC23-0957. If you are an application programmer and want to learn more about writing "servers," see the *VM/SP Programmer's Guide to the Server-Requester Programming Interface for VM/SP*.

Example of CMS Program Development Facilities

This section illustrates several CMS functions that are useful in creating and manipulating CMS files. For detailed descriptions for developing application programs, see the *Application Development Guide*.

Creating an Assembler Language Source File

The sample program that follows is an Assembler language program that reads data from one CMS file and writes it to another CMS file. After you have logged on the system and issued IPL CMS, you can create the program using the CMS XEDIT facility.

```

xedit manip assemble
CREATING NEW FILE:
input
INPUT:
MANIP      CSECT
           PRINT NOGEN
           SAVE  (14,12),,*
           BALR  12,0
           USING *,12      ESTABLISH ADDRESSABILITY
           LA   2,8(,1)    R2=ADDR OF INPUT FILE IN PLIST
           LA   3,32(,1)  R3=ADDR OF OUTPUT FILE IN PLIST
* DETERMINE IF INPUT FILE EXISTS
           FSSTATE (2),ERROR=ERR1
* READ A RECORD FROM INPUT FILE AND WRITE ON OUTPUT FILE
RD         FSREAD (2),ERROR=EOF,BUFFER=BUFF1,BSIZE=80
           FSWRITE (3),ERROR=ERR2,BUFFER=BUFF1,BSIZE=80
           B     RD        LOOP BACK FOR NEXT RECORD
* COME HERE IF ERROR READING INPUT FILE
EOF        EQU   *
           LA   15,7      TEST CODE FOR READ ERROR
           C    15,=F'12'  END OF FILE?
           BNE  ERR3      ERROR IF NOT
           RETURN (14,12),RC=0
* IF INPUT FILE DOES NOT EXIST
ERR1       WRTERM 'FILE NOT FOUND',EIDT=YES
           B     ERRET
* IF ERROR WRITING FILE
ERR2       LINEDIT TEXT='ERROR CODE ..... IN WRITING FILE',    $
           SUB=(DEC,(15))
           B     ERRT
* IF READING ERROR WAS NOT NORMAL END OF FILE
ERR3       LINEDIT TEXT='ERROR CODE ..... IN READING FILE',    $
           SUB=(DEC,(15))
ERRET      RETURN (14,12),RC=1  RETURN TO CALLER
BUFF1     DS    CL80
           END    MANIP

(Press the ENTER/RETURN key to leave Input mode.)
(Remember that the continuation character '$' must be in
in column 72)

XEDIT:
file
Ready;

```

Figure 3. Sample Assembler Language Program Used for Creating a Source File

The Editor (the term applied to the edit program that is used by the XEDIT command) did not find a file with the filename and filetype of MANIP ASSEMBLE, so it created the file for you. Enter the INPUT subcommand so that you can enter your program code into the file. You must issue the FILE subcommand to save your program.

This program (MANIP CSECT) uses several CMS macros; when it is assembled, this program requires the CMS macro library. To identify the macro libraries to be searched, issue:

```
global maclib dmssp tsomac cmslib osmacro
```

Assembling a Source File

To assemble the MANIP program, you enter the "ASSEMBLE MANIP" command, then wait for the assembler to complete processing. The assembler expects to find a CMS file with the filetype ASSEMBLE.

```
assemble manip
```

```
ASSEMBLER (XF) DONE
```

```
MAN00331          B      ERRT  
IFO024 NEAR OPERAND COLUMN 1-UNDEF SYMBOL
```

```
    1 STATEMENT FLAGGED IN THIS ASSEMBLY  
    8 WAS HIGHEST SEVERITY CODE
```

```
Ready (00008);
```

The message IFO024 indicates an error in your program. The line in your program containing the error has a sequence number of MAN00331. Display or print your listing file to find this line.

At this point, three files are associated with your program:

1. The MANIP ASSEMBLE file contains the source statements of your program. This file was the input used by the Assembler

Language program. The output from the assembler is two permanent files, MANIP TEXT and MANIP LISTING.

2. The MANIP TEXT file contains the object module.
3. The MANIP LISTING file contains a listing of the source statements, assembled machine code, and other associated information based on the options selected for the ASSEMBLE command.

Correcting Errors

Since the assembler has detected an error in the source code, you must correct the error before attempting to execute the program. Just as you used the editor to create the assembler file, you also use the editor (either the CMS Editor or the System Product Editor (XEDIT)) to change or correct the assembler file. When you issue the XEDIT MANIP ASSEMBLE command this time, the editor finds your file and enters edit mode. Then issue the LOCATE subcommand to find the line in error. Issue the CHANGE subcommand to correct the error and then issue FILE to save the corrected program. The terminal output is as follows:

```
xedit manip assemble
XEDIT:
locate /errt/
B      ERRT
change /errt/erret/
B      ERRET
file
Ready;
```

Now that the error has been corrected, you can assemble the file again:

```
assemble manip
```

```
ASSEMBLER (XF) DONE

NO STATEMENTS FLAGGED IN THIS ASSEMBLY
Ready;
```

This time, the program assembled without any assembler-detected errors. The TEXT and LISTING files from the previous assembly are erased automatically and replaced by the new ones from the current assembly.

Creating a Load Module

You can now create a load module from the TEXT file that was created by the assembler. The resulting MODULE file can then be executed.

```
load manip
Ready;

genmod manip
Ready;
```

Now a fourth file, MANIP MODULE, exists. This file is in executable form.

Testing and Correcting a Program

Once the MODULE file has been created, you can begin testing. To execute the MANIP MODULE file, issue the MANIP command name, plus the file identifiers for the input and output files. The input file (MANIP ASSEMBLE A1) is to be copied and the resulting file is to be called MANIP1 ASSEMBLE A1. The first test should take the branch on the FSREAD error. The following error message appears on the terminal:

```
manip manip assemble a1 manip1 assemble a1
ERROR CODE 7 IN READING FILE.
Ready (00001);
```

You should then use the Editor to correct the program so that this branch is no longer taken.

```
xedit manip assemble
XEDIT:
find eof
EOF    EQU *
next
LA 15,7 TEST CODE FOR READ ERROR
delete
file
Ready;
```

After the corrected version of the program is filed, assemble and execute the program again.

```
assemble manip

ASSEMBLER (XF) DONE

NO STATEMENTS FLAGGED IN THIS ASSEMBLY
Ready;

load manip
Ready;

genmod manip
Ready;
```

This command produces a file called "MANIP MODULE."

Now that the testing statement has been deleted, and a new MODULE file created, further testing of the program can begin. First, attempt to copy a file that does not exist. The file is not found.

```
manip file1 assemble a1 file2 assemble a1
FILE NOT FOUND
Ready (00001);
```


Then, attempt to copy a file to itself. Your program is not equipped to do this; an error occurs.

```
manip manip assemble a1 manip assemble a1
ERROR CODE 9 IN WRITING FILE.
Ready (00001);
```

Finally, create a new file (MANIP1) from your MANIP file.

```
manip manip assemble a1 manip1 assemble a1
Ready;
```

Erasing Unwanted Files

Once testing is complete, display the beginning of MANIP1 to make sure that it was copied correctly, then delete the MANIP1 file:

```
type manip1 assemble 1 5
MANIP  CSECT
      PRINT NOGEN
      SAVE      (14,12),,*
      BALR     12,0
      USING *,12 ESTABLISH ADDRESSABILITY
```

```
Ready;
```

```
erase manip1 assemble
Ready;
```

Or type DISCARD in the "Cmd" space next to the file you no longer want from the FILELIST screen. Then press the **Enter key** to execute it.

The LISTFILE or FILELIST command can then be issued to make sure the file was erased:

```
listfile * assemble
MANIP  ASSEMBLE A1
Ready;
```

CMS File System

Printing, Sending, and Reading Files

Printing Files

When you want to print your program listing, you should first check the output status of your virtual printer by entering:

```
query 00e
PRT 00E CL A NOCONT NOHOLD COPY 01 READY FORM FORM
      00E FOR USERID DIST distcode FLASHC 000 DEST OFF
      00E FLASH CHAR MDFY FCB
Ready;
```

Since output class A is acceptable for program listings, print the LISTING file:

```
print manip listing
Ready;
```

You can also print the LISTING file by specifying the PRINT option when you issue the ASSEMBLE command. Once the LISTING file is printed, it can be erased. Also, you may want to erase the TEXT file from which the MODULE file was generated:

```
erase manip listing
Ready;
```

```
erase manip text
Ready;
```

Sending Files

If other users want to use your MANIP program, send it to them by using the SENDFILE command. Type SENDFILE, and a screen for data entry is displayed. Then type the name of the file to be sent and the name of the recipient, directly on the screen. You can also select certain options from the list on the screen by answering YES or NO. Press PF5 to send the file and exit from the sendfile screen. Pressing the ENTER key sends the file but keeps the screen (so that you can use it to send another file).

Reading Files

When the user PAYROLL logs on the VM/SP system, the following message types during the logon procedure:

```
FILES: 001 RDR, NO PRT, NO PUN
```

The PAYROLL user can decide whether or not he wants the file before he reads it by invoking the command:

```
query reader all
```

```
ORIGINID FILE CLASS RECORDS CPY HOLD DATE TIME NAME TYPE DIST
NET4      1014 A PUN 0000051 001 NONE 08/17 12:36:03 MANIP TEXT G41/
```

To read in this file, the PAYROLL user must IPL CMS and issue the command:

```
RECEIVE
```

Or issue

```
READ *
```

```
READ CONTROL CARD MISSING. FOLLOWING ASSUMED:..
:READ READCARD CMSUT1 A1
```

CMS reads the first spool reader file in the queue and if there are READ control cards in the input stream, it names the files as indicated on the control cards.

The first card in the deck may not be a READ control card. If it isn't, CMS writes a file named READCARD CMSUT1 A1 to contain the data, until a READ control card is encountered or until the end-of-file is reached.

If there is only one file in the reader and you use the READ * command, the file is moved to your A-disk and is named READCARD CMSUT1 A1.

If there is more than one file in the reader and you use the READ * command to read a second file, the second file is moved to your A-disk and replaces the first file. You can save the first file simply by renaming the READCARD CMSUT1 A1 file before you use the READ * again to read in additional files.

The RECEIVE command does not read in the first file and then replace the previous file with each successive use of the command. Instead, the RECEIVE command receives the file and moves it to your A-disk.

If the PAYROLL user does not want the file, he can purge it from his reader, as follows:

```
purge reader          (or purge reader 1014)
0001 FILE PURGED
```

Or issue the RDRLIST command to display information about files in your reader. Then choose to receive, discard, replace, rename, or peek at the file from the "Cmd" space.

CMS can be used for many other purposes. Those functions illustrated in the previous discussion are intended to help VM/SP users become acquainted with the system and its capabilities. Once

you are familiar with these commands and functions, you have a sound base upon which to build a more thorough understanding of the VM/SP system.

CMS Fileids

The CMS file is the essential unit of data in the CMS system. When you create a file in CMS, you name it using a file identifier (fileid).

A **fileid** consists of:

filename filetype filemode.

The fileid is associated with a particular file when the file is created, defined or renamed under CMS.

A valid **filename** consists of a 1- 8-character alphanumeric field, comprised of A-Z, a-z, 0-9. and special characters \$ # @ + - (hyphen) : (colon) _ (underscore), that is part of the CMS file identifier and serves to identify the file for the user.

A valid **filetype** consists of a 1- 8-character alphanumeric field, comprised of A-Z, a-z, 0-9. and special characters \$ # @ + - (hyphen) : (colon) _ (underscore), that is part of the CMS file identifier and serves to identify the file for the user.

A valid **filemode** is a 2-character CMS file identifier field comprising the filemode letter (A through Z) followed by the filemode number (0 through 6). The filemode letter indicates the CMS file directory on which the file resides and whether or not the disk is a user virtual disk or a CMS system disk. The filemode number indicates the access mode of the disk.

Filemode 0 Makes that file private. No other users may access it unless they have read/write access to your disk.

Filemode 1 Most common for reading and writing

Filemode 2 Essentially the same as filemode 1. Usually is assigned to files that are shared by users who are linked to a common disk, like the system disk.

Filemode 3 Erased after it is read. A filemode of 3 should not be used with EXECs (it may be erased before it completes execution).

Filemode 4 OS simulated data set format, created by OS macros in programs running in CMS.

Filemode 5 Essentially the same as filemode 1 for reading and writing. Used to maintain logical groups so that you can manipulate them easily in groups.

Filemode 6 Indicates that the "update-in-place" attribute of a CMS file is in effect. This means that the existing records of a file are written back to their previous location on disk rather than in a new slot. This only applies to files located on 512-, 1K-, 2K-, or 4K-byte block formatted minidisks.

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

Reserved Filetype Descriptions

The following figure lists the filetypes used by CMS and CMS/DOS commands.

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

Filetype	Command	Usage	Filename	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 DISK LOAD TAPE LOAD UPDATE INCLUDE LOAD MACLIB EDIT XEDIT	Intermediate work files	READCARD COPYFILE DISK TAPE fn DMSLDR DMSLDR DMSLBM EDIT XEDIT	F 80	
CNTRL	UPDATE XEDIT	Input	fn	F 80	Control file update
COBOL	COBOL	Input	fn	F 80	COBOL source statements
COPY	MACLIB SSERV	Input Output	fn fn	F 80	COPY control cards and macro definitions A book from a DOS/VS source library
DIRECT	DIRECT	Input	fn	F 80	User directory entries

Figure 4 (Part 1 of 5). Reserved Filetypes

Filetype	Command	Usage	Filename	Format RECFM LRECL	Contents
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
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 sLISTIO	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
HELPABBR	HELP	Input	fn	V/F 79	Input files for HELP facility
HELPCMS	HELP	Input	fn	V 79	Input files for HELP facility
HELPCMQ	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
HELPCPQU	HELP	Input	fn	V 79	Input files for HELP facility
HELPCPSE	HELP	Input	fn	V 79	Input files for HELP facility
HELPDEBU	HELP	Input	fn	V 79	Input files for HELP facility
HELPEEDIT	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

Figure 4 (Part 2 of 5). Reserved Filetypes

Filetype	Command	Usage	Filename	Format RECFM LRECL	Contents
HELPHLP	HELP	Input	fn	V 79	Input files for HELP facility
HELPIPCS	HELP	Input	fn	V 79	Input files for HELP facility
HELPMENU	HELP	Input	fn	V 77	Input files for HELP facility
HELPMMSG	HELP	Input	fn	V 79	Input files for HELP facility
HELPPF	HELP	Input	fn	V 79	Input files for HELP facility
HELPPREF	HELP	Input	fn	V 79	Input files for HELP facility
HELPPQUER	HELP	Input	fn	V 79	Input files for HELP facility
HELPPREXX	HELP	Input	fn	V 79	Input files for HELP facility
HELPPSET	HELP	Input	fn	V 79	Input files for HELP facility
HELPPSQLD	HELP	Input	fn	V 79	Input files for HELP facility
HELPPSRPI	HELP	Input	fn	V 79	Input files for HELP facility
HELPTASK	HELP	Input	fn	V 105	Input files for HELP facility
HELPTSFAF	HELP	Input	fn	V 79	Input files for HELP facility
HELPPXEDI	HELP	Input	fn	V 79	Input files for HELP facility
\$HLPxxxx	HELPCONV	Output	fn	same as input	Converted file 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	Output	fn		
LISTING	FORTGI	Output	fn	V 121	
	FORTHX				
	COBOL				
	PLIC				
LISTING	PLICR	Output	fn	V 121	
	PLIOPT	Input	fn		
TESTCOB	Input	fn			
LKEDIT	LKED	Output	fn	F 121	Listing
LOGFILE	SET LOGFILE	Output	fn vscreen name	V none	Log of data written to virtual screen.
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

Figure 4 (Part 3 of 5). Reserved Filetypes

Filetype	Command	Usage	Filename	Format RECFM LRECL	Contents
MAP	DOSLIB DOSLKED	Output Output	libname fn	F 80	Library map DOS/VS linkage editor map Directory information from DOS/VS private or system libraries Module map Module map Library Map Library Map Library Map
	DSEVR INCLUDE LOAD MACLIB TXTLIB TAPE	Output Output Output Output Output	DSEVR LOAD LOAD fn fn fn		
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.
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	PSEVR	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 ¹	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		
SYSUT4	COBOL LKED PLIC PLICR TESTCOB	Work	fn	F 80	Used as input to TESTCOB
		Input		512	
TESTFORT	TESTFORT	Output	fn	VB 125	Processor printed output

Figure 4 (Part 4 of 5). Reserved Filetypes

Filetype	Command	Usage	Filename	Format RECFM LRECL	Contents
TEXT	ASSEMBLE ASM3705	Output Output	fn fn	F 80	Object code 3704/3705 source code and job control language statements Object code Object code Linkage editor control statements for 3704/3705 control programs Object code Object code and LKED control cards Object code Object code Object code Object file
	COBOL DOSLKED GEN3705	Output Input Output	fn fn fn(Ln)		
	INCLUDE LKED	Input Input	fn fn		
	LOAD PLIOPT TXTLIB GOFORT FORTGI FORTHX RSERV TEXTFORT	Input Output Input Output Output Output Output Input	fn fn fn fn fn fn		
TXTLangid	GENMSG	Output	fn	F 80	Object code for language files.
TXTLIB	GLOBAL TXTLIB	Library Output	fn fn		Object decks (dictionary and members)
UPDATE	UPDATE	Input	fn	F 80	UPDATE control cards
UPDLOG	UPDATE	Input	fn	F	UPDATE log
VS BASIC	VS BASIC	Input	fn	F ≤256	VS BASIC language source statements
VS BDATA	VS BDATA	Execution time files	fn	V ≤140	VS BASIC 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

Figure 4 (Part 5 of 5). Reserved Filetypes

- ¹ SCRIPT/VS is a component of the IBM Document Composition Facility program product, which is available from IBM for a license fee. For additional information on SCRIPT/VS usage, see *Document Composition Facility: User's Guide*, SH20-9161.

Return Codes

If a condition arises during execution of a command that results in the display of a warning, error, severe error, or terminal error message, the command passes a nonzero return code back in register 15.

If no warning, error, severe error, or terminal error messages are generated during execution of the command, the return code passed back in register 15 is zero.

Commands that invoke program products pass the return code set by that program in register 15. This code may have the same number as a CMS code described above; however, it will have been redefined by the program product or compiler in operation.

CMS Return Codes

The following figure shows the return codes passed by CMS commands.

-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).
0	Normal.
1	Device disconnected.
1	Top or bottom of virtual screen reached.
3	Virtual screen, window, or queue already exists.

Figure 5 (Part 1 of 2). Return Codes Produced by CMS

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	Window name of "*" or "=" not allowed.
20	An invalid character is in the fileid. Valid characters are: 0-9, A-Z, a-z, \$, @, #, +, -(hyphen), :(colon), _(underscore).
24	CMS virtual screen or window cannot be deleted.
24	The user did not specify the command line correctly.
28	Virtual screen, window, or queue not defined.
28	An error occurred while trying to access, or manipulate, a user's files; for example, file not found.
32	Invalid position specified.
32	The user's file is not in the expected format, or the user's file does not contain the expected information.
36	Window not connected or displaying virtual screen.
36	No field to write data/color/exthi/PSset.
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.
40	A functional error for which the user is responsible occurred during execution of the command, or the user failed to supply all the necessary conditions for executing the command, or end-of-file, end-of-tape (where applicable).
41	Insufficient storage was available to execute the command.
88	TTY device.
88	A CMS system restriction prevented execution of the command, or the function requested is an unsupported feature, or the device requested is an unsupported device.
100	I/O error.
100	Input/output device errors.
104	Insufficient storage.
104	A functional error for which the system is responsible occurred during execution of the command.
256	Request rejected by IUCV.
256	All unexpected errors for which the system is responsible; that is, terminal error messages.

Figure 5 (Part 2 of 2). Return Codes Produced by CMS

CP DIRECT Command Return Codes

The following figure shows the return codes passed by the CP DIRECT command.

1	Invalid filename, or file not found.
2	Error loading the directory.
3	Invalid option from CMS.
4	Directory not swapped, user not privilege class A, B, or C.
5	Directory not swapped, system (old) directory locked.
6	Directory not swapped, the directory in use by the system is not the updated directory.
1xx ¹	Error in the CMS RDBUF routine.
2xx ¹	Error in the CMS TYPLIN routine.

Figure 6. Return Codes Produced by the CP DIRECT Command

¹ xx is the CMS routine return code.

CMS DDR Command Return Codes

The following figure shows the return codes passed by the CMS DDR command.

1	Invalid filename, 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.
1xx ¹	Error in the PRINTIO routine.
2xx ¹	Error in the CONREAD routine.
3xx ¹	Error in the RDBUF routine.
4xx ¹	Error in the TYPLIN routine.

Figure 7. Return Codes Produced by the CMS DDR Command

¹ xx is the CMS routine return code.

System Product Editor (XEDIT) Command Return Codes

The following figure shows the return codes passed by the XEDIT command.

-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

Figure 8 (Part 1 of 3). Return Codes Produced by the XEDIT Command

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 during execution; File has been filed, was 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
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; Lrcel 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 or QUIT issued in macro
7	Error building the update file
8	Prefix area contains pending subcommand or macro; I/O error or modifications lost because PA key pressed when message pending
12	Disk defined in filemode 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 filename or filetype
24	Invalid filemode; Invalid parameters, or options; Invalid columns defined
28	Source file not found (UPDATE MODE), or library not found (MEMBER option), or specified profile macro does not exist, or file XEDTEMP CMSUTI already exists; Filename 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, 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; or previous MACLIB function not found

Figure 8 (Part 2 of 3). Return Codes Produced by the XEDIT Command

100	Error reading/writing file to disk; Error from rdbuf; Error occurred while creating the file
104	No storage is available
nn	Command's RC 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
any number > 10	Standard CMS HELP command return codes

Figure 8 (Part 3 of 3). Return Codes Produced by the XEDIT Command

IPCS Commands Return Codes

The following figure shows the return codes passed by the IPCS commands.

4	An incorrectly entered parameter.
8	System failure; a read/write error or an invalid internal parameter.

Figure 9. Return Codes Produced by the IPCS Commands

Example of a Return Code from a CP Command

Commands or operands of commands passed to CP pass the return code sent back by CP to register 15. For example, suppose the user is in CMS mode and invokes the CP command LINK:

```
ipl cms  
  
CMS VERSION n.n mm/dd/yy  
-----  
-----  
cp link to * vaddr1 as vaddr2 r
```

The user has entered the CP command LINK to userid *. That means the user's own directory is searched for device vaddr1. Vaddr2 is the virtual address to be assigned to the device for this virtual machine. Read-only access is requested. No password is required because the user has linked to one of his own disks.

The result may be either

Ready; (or Ready(0);)

which indicates successful execution.

or

Ready(nnnnn);

which indicates an error. If, as previously described, 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.

The return code may be used by a system programmer in the DEBUG subcommand and also in EXEC procedures. See the *VM/SP CMS Command Reference* or *EXEC 2 Reference* for a description of the 'RC' special variable.

Chapter 3. Summary of VM/SP Commands and Service Aids

The remainder of this book details command formats and descriptions. Also included are macro formats and service aids.

It is recommended that VM/SP users of RSCS obtain the RSCS Networking Version 2 (5664-188) program product. This program product takes advantage of the enhanced functions of VM/SP. The commands listed are for the RSCS program product.

Users of VM/SP who intend to use the RSCS component of VM/370 Release 6 should retain the Release 6 reference library.

VM/SP has an enhanced Interactive Problem Control System (VM/SP IPCS) component. This component replaces the unmodified VM/370 interactive problem control system. Details of this major component are found in the *VM Distributed Data Processing Guide*, LY24-5241.

Note: The VM/SP Release 4 base product is enhanced to include function equivalent to that within the VM/Interactive Problem Control System Extension (VM/IPCS/E) licensed program product (5748-SA1). The IPCS commands listed here are those of VM/SP IPCS.

Notational Conventions

The following symbols should be coded as they appear in the command format:

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

The following symbols are used to define the command format and should not be coded as part of the command.

braces { }

indicate choices, one of which **must** be selected. If a list of choices is enclosed by neither brackets or braces it is to be treated as if enclosed by braces.

brackets []

indicate optional choices, one of which may be selected.

underscore _

indicates a default option. If the underscored option is selected, it need not be specified.

vertical bar |

separates the operand alternatives within the brackets and braces.

ellipsis

indicates that the preceding item may be repeated more than once.

The commands and subcommands are shown in uppercase and lowercase. The uppercase represents the minimum truncation of the command or keyword operand that the system accepts. An all-lowercase operand indicates a user- or system-supplied variable value. Variable abbreviations used are shown in the following figure.

Short Form	Meaning
...	Alphameric information
addr	Storage address
cc	Cylinder number
col	Column
cuu	Virtual device address
cyl	Cylinder
dasd	Direct access storage device
fileid	fn ft [fm]
fm	filemode
fn	filename
ft	filetype
hexloc	hexadecimal storage location
mode	Mode letter or mode letter and mode number
nn	Decimal information
psw	Program status word
raddr	Real storage address
rec	record
vaddr	Virtual storage address

Figure 10. Syntax Abbreviations Used

For a detailed and thorough explanation of command formats and notational conventions, see the *VM/SP CMS Command Reference*.

CP Privilege Classes

The CP commands are divided into seven functional types: operations, resource, programmer, spooling, analyst, CE, and general. These types apply whether the installation retains the IBM-defined class structure or establishes its own class structure. Each functional type corresponds to one IBM-defined privilege class. For details of the function of each IBM-defined privilege class and its corresponding function type, see the *VM/SP CP Command Reference*. Each user is assigned, as part of his entry in the directory, one or more privilege classes.

This is a summary of the CP privilege classes and the functions performed by each class of user:

Class A Primary System Operator

The class A user controls the VM/SP system. Class A is assigned to the user at the VM/SP system console during IPL. The primary system operator is responsible for the availability of the VM/SP system and its communication lines and resources. In addition the class A user controls system accounting, broadcast messages, virtual machine performance options, and other command operands that affect the overall performance of VM/SP. The system operator controls operation of the real machine using the system control panel and console device.

Note: The class A system operator who is automatically logged on during CP initialization is designated as the primary system operator.

Class B System Resource Operator

The class B user controls allocation and deallocation of all the real resources of the VM/SP system, except those

controlled by the primary system operator and spooling operator.

Class C System Programmer

The class C user updates certain functions of the VM/SP system. The system programmer can modify real storage in the real machine.

Class D Spooling Operator

The class D user controls spool data files and specific functions of the system's unit record equipment.

Class E System Analyst

The class E user displays the contents of real storage, performs the functions required to generate saved systems and discontinuous saved segments, and controls the collection and recording of performance measurement data. This class of user can display the contents of specified real storage areas on the virtual operator's console or on a spooled virtual printer, but cannot modify real storage.

Class F Service Representative

The class F user obtains and examines, in detail, certain data about input and output devices connected to the VM/SP system. The service representative can establish intensive recording mode for one I/O device at a time and can cause the recording of repressible machine check errors to be initiated or resumed.

Class G General User

The class G user controls functions associated with the execution of his virtual machine. A general user cannot display or modify real storage.

Class Any The Any classification is given to certain CP commands that are available to any user. These are primarily for the purpose of gaining and relinquishing access to the VM/SP system.

Class H Reserved for IBM use.

Chapter 4. CP, CMS, GCS, TSAF, RSCS, and IPCS Commands

The remainder of the book contains CP, CMS, GCS, TSAF, IPCS, and RSCS commands and the VM/SP service aids. A brief description precedes a syntactic representation of each command.

Note: RSCS commands must be preceded by "RSCS", if issued by the console operator, or by "MSG RSCSvmid" if issued by a system-authorized alternative operator.

Note: Either the TSAF virtual console or the secondary user of the TSAF virtual machine issues the TSAF commands. If the secondary user issues the commands, TSAF commands must be preceded by **SEND userid**. The value in **userid** is the userid 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

ACCESS

CMS

Defines direct access space to a CMS virtual machine and relates it to a logical directory.

```
Access [ [cuu] mode [/ext [fn [ft [fm]]] ] [(options...{})] ]
       [ 191  A        [ * [ * [ * ] ] ] ] ]
```

options:

```
[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 virtual machine's execution.

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

ALARM VSCREEN

CMS

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

ALARM VScreen vname

AMSERV

CMS

Defines VSAM catalogs, data spaces, or clusters. Alters, lists, copies, deletes, exports, or imports VSAM catalogs and data sets.

```
AMserv fn1 [fn2|fn1][(options...{})]
```

options:

```
[PRINT] [TAPIN {18n|TAPn}] [TAPOUT {18n|TAPn}]
```

APAR

APAR

IPCS

Invokes the functions of the PRB and PROB commands and produces a hard-copy APAR form for submittal to IBM.

APAR

ASMGEND

CMS

Builds the system assembler and creates the associated auxiliary directory. (System programmers only)

ASMGEND

ASM3705

CMS

Invokes the 3705 assembler.

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:

```
[LIST|NOLIST]  
[LINECOUN 55|LINECOUN nn]  
[DISK|PRINT|NOPRINT]  
[XREF (SHORT) |XREF (FULL) |NOXREF]  
[DECK|NODECK]  
[RENT|NORENT]  
[LOAD|NOLOAD]
```

ASSEMBLE

CMS

Invokes the system assembler.

Assemble fn [(options...[])]

listing control options:

[ALOGIC|NOALOGIC] [MLOGIC|NOMLOGIC]
[ESD|NOESD] [RLD|NORLD]
[LIST|NOLIST] [LIBMAC|NOLIBMAC]
[MCALL|NOMCALL] [FLAG (0)|FLAG (nnn)]
[LINECOUN (55)|LINECOUN (nn)]
[DISK|PRINT|NOPRINT]
[XREF (FULL)|XREF (SHORT)|NOXREF]

output control options:

[DECK|NODECK] [OBJECT|NOOBJECT]
[TEST|NOTEST]

SYSTEM options:

[NUMBER|NONUM] [STMT|NOSTMT]
[TERMINAL|NOTERM]

other options:

[ALIGN|NOALIGN] [BUFSIZE (STD)|BUFSIZE (MIN)|BUFSIZE (MAX)]
[RENT|NORENT] [SYSPARM (string)|SYSPARM (?)|SYSPARM ()]
[WORKSIZE (2048K) | WORKSIZE (nnnnnK)]
[YFLAG|NOYFLAG]

ASSGN

ASSGN

CMS

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

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

options:

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

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] [VALID valid]
          {SYSTEM [AS] valid}
          {raddr...
           raddr-raddr} [TO] userid [R[/O]] [3330V]
          Laddr [TO] userid [AS] vaddr
          CHANnel c [PROC nn] [TO] {userid
                                   *}
```

ATTN

CP Class G

Makes attention interruption pending.

ATTN

AUTOLOG

AUTOLOG

CP Class A or B

Logs on any virtual machine that is 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:

Punch Format:

Backspac { raddr } [File
 { lpri } [pages [EOF]] Backspac raddr [File]
 [1]

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]

BEGIN

CP Class G

Starts the execution of a virtual machine.

Begin [hexloc]

C (CMS Border Command)

C

CMS Border Command

Clears the window of scrollable data.

c

CATCHECK

CMS

Invokes the VSE/VSAM Catalog Check Service Aid to verify a complete catalog structure. Provides a print file containing the catalog analysis.

CATCHECK [catname
 [catname/password]

CHANGE

CP Class D

Alters the attributes of a closed spool file.

```

Change [userid] {Reader} {Class c1} {
  [SYSTEM] {Printer} FORM form1 CHars name0 2 [[CHars] name1]
  [_] {PUnch} DEST dest1 [[CHars] name2]
  spoolid [[CHars] name3]
  ALL
}
Class c2
COpy[*]nnn
DIst distcode
FCB name 2
FLash name nnn
FORM form2
DEST dest2
}
{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

CHANGE

CP Class G

Alters the attributes of a closed spool file.

```
Change {Reader } {Class c1 } {Class c2 } 1 {Name {fn [ft] } }
      {Printer } {spoolid }
      {Punch } {FORM form1 } {HOLD }
              {DEST dest1 } {NOHold }
              {Dist dist }
              {COpy [*]nnn }
              {FLash name nnn }

      {MODify name [n] }
      {CHars name1 { [CHars]name2 }
                { [CHars]name3 }
                { [CHars]name4 } }
      {FCB name }
      {FORM form2 }
      {DEST dest2 }
      {UNCONV }
```

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 normally used in this publication.

CHANGE

RSCS

Alters one or more attributes of an inactive spool file.

General User Format:

```
Change [*]spoolid options...*
```

Operator Format:

```
Change [linkid] spoolid options...*
```

* You must include at least one of the following options:

```
[PRIority nn ]
[CLass c ]
[COpy [*]nnn ]
[DIst distcode ]
[FLash name nnn ]
[HOLD |NOHold ]
[MODify name [trc] ]
[CHars name1 [CHars name2 ...] ]
[FCB name ]
[FOrm ccccccc ]
[DEST {ccccccc|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 no scrollable data is displayed in the window.

```
CLEAR WINDow [wname]
[ = ]
```

CLOSE

CLOSE

CP Class G

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

```
close {
  {
    Reader
    vaddr {Hold
           NOHold}
  }
  {
    CONsole {Purge
             FORM form
             DEST dest
             vaddr {Hold [Dist distcode]
                   NOHold}
             {Name {fn[ft]
                   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 or standard PLIST command calls.

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

CMSBATCH

CMS

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

```
CMSBATCH [sysname]
```

CMSGEND

CMS

Generates a new CMS module from a text file and places the new CMS module on the specified disk.

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

CMSSERV

CMS

Starts communications between your VM/SP host system and your work station (for example, IBM Personal Computer) for IBM System/370 to IBM Personal Computer Enhanced Connectivity Facilities.

CMSSERV

COMMANDS

CP Class Any

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

COMMANds

COMPARE

CMS

Compares two existing files on a record for record basis and displays dissimilar records.

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

option:

COL

```
[mmm[-]nnn]
[1 recl ]
```

CONVERT

CONVERT

IPCS

Converts symptom summary file and PRBnnnn dumps from the system component IPCS format to that required for VM/IPCS/E.

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

CONVERT COMMANDS

CMS

Use CONVERT COMMANDS to take DLCS statements in a CMS file and create an internal form (a text deck) of the table for the parsing facility to use.

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

```
options:  [({[SYSTEM|USER|ALL]
```

```
          [CHECK|OUTmode  [*|fm]]
```

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

CONVIPCS EXEC

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 files according to operand specifications.

```
COPYfile fileid1 [fileidi2...] [fileido] [(options...)]
```

options:

```
[Type ] [NEWDate] [NEWFile] [PROMpt ]  
[NOType] [OLDDate] [REPlace] [NOPRompt ]  
  
[From recno ] [FOR numrec ] [SPecs ] [OVly ]  
[FRLabel xxxxxxxx] [TOLabel xxxxxxxx] [NOSPecs] [APpend ]  
  
[RECFm {F } ] [LRecl nnnnn] [TRUnc ] [PAck ] [Fill c ]  
[V ] [NOTRunc] [UNPack] [Fill hh ]  
[Fill 40 ]  
  
[EBcdic] [UPcase ] [TRAns] [SIngle]  
[LOWcase ]
```

COUPLE

CP Class G

Connects virtual channel to channel adapters.

```
COUPLE vaddr1 [TO] userid vaddr2
```

CP

CMS

Permits entry of CP commands without leaving the CMS environment.

```
CP [commandline]
```

CP

CP Class Any

Permits execution of CP commands within your privilege class.

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


CP

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.

CPQuery {
 INDicate
 LOGmsg
 Names
 Time
 Users [userid]
 CPuid
 CPLEVEL
}

CPTRAP

CP Class C

Use to create a file of trace table and CP and virtual machine interface records in the order they happen for problem determination. The CPTRAP READER file can be printed out or displayed at a terminal using the TRAPRED command.

```

CPTrap {
  typenum {
    Vmblok  address
    DEVaddr cuu
    CODE    code-value
    OFF
  }
  ALL {
    ON
    OFF
  }
  ALLOWid  userid
  GGroupid group-name
  STArt {
    [TO] {userid} [WRAP wrap-size]
          {
            *
            _
          }
  }
  CLOSE
  STOP
}

```

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

DCP

CP Classes C and E

Displays real processor storage on the terminal.

```
DCP { MLhexloc1 [-] [hexloc2] }  
    { NLhexloc1 [:] [END] }  
    MThexloc1  
    NThexloc1  
    Mhexloc1  
    Nhexloc1 [.] [bytecount]  
    Lhexloc1 [END]  
    Thexloc1  
    hexloc1  
    L  
    T  
    O }
```

DDR

CMS

Dumps, restores, prints, or copies data from DASD devices and tape devices.

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

I/O definition statements:

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

options:

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

DDR (continued)

CMS

SYSPRINT control statement:

SYsprint {cuu|CONS}

function control statements:

$\left\{ \begin{array}{l} \text{DUMp} \\ \text{COpy} \\ \text{REstore} \end{array} \right\}$	[FTr]	¹	$\left[\begin{array}{l} \text{cyl1 [To] [cyl2 [Reorder] [To] [cyl3]]} \\ \text{block1[To][block2[Reorder][To][block3]]} \\ \text{CPvol} \\ \text{NUcleus} \\ \text{ALL} \end{array} \right]$
--	-------	--------------	---

PRINT/TYPE function statements:

$\left\{ \begin{array}{l} \text{PRint} \\ \text{TYpe} \end{array} \right\}$	$\left[\begin{array}{l} \text{cyl1 [hh1 [rr1]] [To] cyl2 [hh2[rr2]]} \\ \text{block1 [To] block2} \end{array} \right] \left[(\text{options...}) \right]$
---	--

options:

[Hex] [Graphic] [Count]

¹ The FTr option is valid only with the DUMP control statement.

DEBUG

CMS

DEBUG

Enters the **DEBUG** environment to test and debug a program.

DEBUG

The format of each **DEBUG** subcommand is followed by its description:

Break {id symbol}
 hexloc}

Stops program execution at a specific instruction location.

CAW

Displays the Channel Address Word (CAW).

CSW

Displays the Channel Status Word (CSW).

DEfine symbol hexloc [bytecount]
 4

Assigns a symbolic name to a specific storage address.

DUmp [symbol1 hexloc1 0 [symbol2 hexloc2 * 32 [ident]]]

Dumps the contents of storage locations to the virtual printer.

GO [symbol]
 hexloc]

Exits from the **DEBUG** environment and begins program execution.

DEBUG (continued)

CMS

GPR reg1 [reg2]
Displays the contents of the specified general registers.

HX
Returns to CMS environment.

ORigin { symbol|hexloc|0 }
Sets a base address.

PSW
Displays old PSW.

RETurn
Returns to CMS environment when DEBUG environment was entered via DEBUG command.

SET { CAW hexinfo
 CSW hexinfo [hexinfo]
 PSW hexinfo [hexinfo]
 GPR reg hexinfo [hexinfo] }
Changes the contents of the specified general registers and control words.

STore { symbol } [hexinfo [hexinfo]]
 hexloc
Stores up to 12 bytes of hexadecimal information in the specified virtual location.

X { symbol [n|LENGTH]
 hexloc [n|4] }
Examines virtual storage locations.

DEFAULTS

CMS

Set or display default options for the commands: FILELIST, HELP, NOTE, RDRLIST, RECEIVE, PEEK, SENDFILE, TELL, and MACLIST.

DEFAULTS [Set command options...]
 [LIST [command]]

DEFINE

DEFINE

CP CLASS A or 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 {
  Reader  [AS] VADDR
  Printer 1403, 1443 ... 4248

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

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

```


DEFINE

DEFINE (continued)

CP CLASS G

```
{ 3800 }
{ 3800-1 } [As] vaddr [Size ww 11] [2WCGM] [BTS] [DATck]
{ 3800-3 } [4WCGM] [CFS] [NODatck]

CHANnels [As] {SEL}
           {BMX}

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

vaddr1 [As] vaddr2

GRAF cuu { 3033 }
          { 3036 }
          { 3138 }
          { 3148 }
          { 3158 }
          { 3270 }

{ TFB-512 }
{ T3310 } [As] vaddr [BLK] [nnnnnn]
{ T3370 }

{ T2305 }
{ T2314 }
{ T2319 }
{ T3330 } [As] vaddr [CYL] [nnn]
{ T3340 }
{ T3350 }
{ T3375 }
{ T3380 }

STORage [As] {nnnnnK}
           {nnM}
```

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|NOASart]
             [Class      c      ]
             [DP         dpriority ]
             [KEEP       holdslot ]
             [LINE       vaddr   ]
             [LOGMode    logmodename ]
             [LUName     luname   ]
             [Queue      (Priority|Fifo|Size) ]

             [
               TYPE          [
                               NJE
                               SNANJE
                               RJE
                               MRJE
                               3270P
                               SNA3270P ]
               ]
             [Parm [parameters... ] ]
    
```

DEFINE VSCREEN

CMS

Creates a virtual screen.

```

DEfIne VScreen vname lines cols rtop rbot [(optionA optionB optionC optionD[])]
    
```

```

optionA: [ TYpe ]
         [ NOnType ]
    
```

```

optionB: [ PRotect ] [ High ]
         [ NOPRotect ] [ NOHigh ]
    
```

```

optionC: [color] [exthi] [PSset]
    
```

```

optionD: [ USer ]
         [ SYstem ]
    
```

DEFINE WINDOW

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 LINK

TSAF

Removes a communication link from TSAF's list 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 VSCREEN

CMS

Removes a virtual screen definition.

```
DELeTe VSCreen  vname
```

DELETE WINDOW

DELETE WINDOW

CMS

Removes a window definition.

DELEte WInDow wname

DESBUF

CMS

Clears the console and program stack input and output 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 * }
 Laddr [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

DIAL

CP Class Any

Attaches a terminal device to a multiple access system.

```
DIAL userid [vaddr]
```

DIRECT

CMS

Allows creation, editing, and swapping of VM/SP user directory.

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

Control statements:

```
Account number [distribution]
```

```
CLASS classes
```

```
Console vaddr devtype [class] [userid]
```

```
Dedicate { NETWORK vaddr resource  
          vaddr rdev {[VOLID] [volser] [3330V] [R/O]}  
          {[VOLID] [volser] [3330V] [R/O]}}
```

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

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

```
IUCV [userid] [PRIORITY] [MSGLIMIT limit]  
     [*CCS  
     *SIGNAL  
     ALLOW  
     ANY]
```

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

DIRECT (continued)

CMS

```
Mdisk cuu devtype {cylr   cyls volser [mode{pr[pw[pm]]}] }
                   {T-DISK cyls
                   {blkcr blkbs
Option [Realtimer] [Ecmode] [Virt=Real] [Acct] [Svcoff] [BMX]
       [CPUID bbbbbbb] [AFFinity nn] [Isam]
       [VMsave] [STFirst] [370E] [Maxconn nnnnn] [MIH]
       [DIAG98]
       [VCUNOSHR]

SCreen area {color {highlight}} {highlight {color {Default}}} ...
            {NONE}

SPEcial vaddr devtype [IBM Tele]

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

User userid pass [stor [mstor [cl * [pri 1e [1d [cd [es [ON [ON ]]]]]]]]] ]]]]
                 [OFF [OFF [OFF [OFF ]]]]]]]]] ]]]]
```

¹ If you use *, the USER control statement should be followed immediately by the CLASS control statement.

DISABLE

CP Class A or B

Prevents communication lines from accessing VM/SP.

```
DISable {raddr...
         {Laddr...
         {SNA [userid]
         {ALL
```

DISCONN

CP Class Any

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

```
DISConn [HOLD]
```

DISCONN

DISCONN

RSCS

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

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

DISK

CMS

Dumps disk files to punched cards and restores disk files.

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

Options:

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

DISKMAP

CMS

Summarizes the MDISK statements in a CP directory. The output of the exec shows gaps and overlaps between minidisk assignments.

```
DISKMAP filename [ filetype ]  
                 [ DIRECT ]
```

DISPLAY

CP Class G

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

```
Display { hexloc1 } { - } { hexloc2 }  
        { Khexloc1 } { : } { END }  
        { Lhexloc1 }  
        { Thexloc1 }  
        { . } { bytcount }  
        { END }  
  
        { Greg1 } { - } { reg2 }  
        { Yreg1 } { : } { END }  
        { Xreg1 }  
        { . } { regcount }  
        { END }  
  
        Psw  
        CAW  
        CSW }
```


DLBL

DLBL

CMS

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

In CMS, defines and identifies VSAM catalogs, clusters, and data spaces; identifies VSAM files used for program input/output; identifies VSAM files used for program input/output; identifies input/output files for Access Method Services.

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

optionA:	optionB:	optionC:
[SYSxxx]	[PERM]	[VSAM]
	[<u>CHANGE</u>]	[<u>EXTENT</u>]
	[NOCHANGE]	[MULT]
		[CAT catdd]
		[BUFSP nnnnnn]

DLBL

GCS

Defines VSAM files used for program input/output.

```
DLBL [ ddname mode [ DSN qual1 [[.]qual2...qualn] [(option B option C [])]
      [ DSN ? ]
      ddname CLEAR
      * ]
```

optionB:

optionC:

[PERM]

[VSAM]

[MULT]

[CAT catdd]

```
[CHANGE ]
[NOCHANGE]
```

[BUFSP nnnnnn]

DMCP

CP Class C or E

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

```
DMCP [ [ MLhexloc1 ] [ { : } [ hexloc2 ] ] [ *dumpid ]
      [ NLhexloc1 ] [ { - } [ END ] ]
      [ MThexloc1 ]
      [ NThexloc1 ]
      [ Mhexloc1 ] [ { . } [ bytecount ] ]
      [ Nhexloc1 ] [ [ END ] ]
      [ Lhexloc1 ]
      [ Thexloc1 ]
      [ hexloc1 ]
      [ 0 ] ]
```

DOSLIB

DOSLIB

CMS

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

```
DOSLIB { DEL libname phasename1 [...phasenamen]
        COMP libname
        MAP libname [(options...{})] }
```

options:

```
[ TERM
  DISK
  PRINT ]
```

DOSLKED

CMS

In CMS/DOS, link-edits TEXT files from CMS disks, or object modules from DOS/VSE private or system relocatable libraries, and places them in executable form in a CMS phase library (DOSLIB).

```
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 only the most recently created program stack buffer or a specified program stack buffer and all buffers created after it.

```
DROPBUF n
```

DSERV

DSERV

CMS

Obtains information contained in DOS/VSE private or system libraries.

```
DSERV { CD [ PHASE { name [ nn ] } ] } [d2 ... dn][{options...[]}]
      { RD
        SD
        PD
        TD
        ALL } { options:
              { DISK
                TERM
                PRINT
              }
              [ SORT ]
```

DUMP

CP Class G

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

```
DUmp [ Lhexloc1
      Thexloc1
      hexloc1 ] { { - } [ hexloc2 ] } [*dumpid]
              { : } [ END ]
              { . } [ bytcount ]
              { END }
```

DUMPSCAN

GCS

This is an IPCS command that enables you to interactively examine a dump of a Group Control System (GCS) virtual machine existing as a CMS file.

The **DUMPSCAN subcommands**, supported by GCS, followed by their descriptions, are:

IUCv

Displays all entries in the IUCV path table.

TActive {taskid}
 [ALL]

Displays the task's active program list.

TLoad1 {taskid}
 [ALL]

Displays the task load list.

TSab {taskid}
 [ALL]

Displays the subpool map and chain header of a task.

VMLoad1

Displays information about all NUCCBLKs on a virtual machine's load list.

DUMPSCAN

DUMPSCAN

IPCS

Enables you to interactively examine a dump existing as a CMS file created by IPCSDUMP.

```
DUMPSCAN [ HELP [ nnnnn [ fm ] ] ]
```

The **DUMPSCAN subcommands**, followed by their descriptions, are:

- (null line) (Common)
Reissues the previous CHAIN, LOCATE, or SCROLL subcommand.
- ? (Common)
Displays last subcommand entered.
- + (increment) (Common)
- (decrement)
Adjusts the address pointer and reissues the DISPLAY command.
- &name [subcommand] (Common)
- &
Creates a synonym for frequently used subcommands.
- Aregs (CP dump only)
Displays the registers, clocks, PSW, CSW, and CAW for the attached or the non-IPL processor (AP).
- ARIOBLOK cuu (CP dump only)
Displays the RCHBLOK, RCUBLOK, and RDEVBLOK for the specified device attached to the non-IPL processor in an MP configuration.
- C (CP dump only)
Displays the control registers for the failing processor.

DUMPSCAN (continued)

IPCS

CHain fromhexloc increment endval	(Common)
Verifies the chain of homogeneous control blocks that start at the specified location.	
CMS	(Common)
Enters the CMS subset environment.	
CMSPoint	(CMS dump only)
Displays the formatted contents of 17 pointers from CMS NUCON.	
CORtable hexloc	(CP dump only)
Displays page status and the formatted contents of the CORTABLE entry for the hexadecimal location specified.	
Display { hexloc } [nnnn]	(Common)
{ hexloc% Thexloc Thexloc% }	
Displays areas in the dump. The actual address or an indirect address may be specified.	
DOSPoint	(CMS dump only)
Displays the formatted contents of five pointers used by DOS simulation.	
DUMPID	(Common)
Displays a dump identification message and the dumpid information from the VMDUMP command line (if any).	
END	(Common)
Ends the session and returns to CMS.	

DUMPSCAN

DUMPSCAN (continued)

IPCS

FDISPlay
only)

(TSAF dump

```

|   { PATH
|     SERVICE
|     CLUSTER
|     RESOURCE
|     NEIGHBOR
|     ROUTING
|     LINKDEF
|     LINKCTL
|     ALL
|   }
|
|   { BSC
|     CTCa
|   }
```

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

G (Common)
Displays the set of general registers in the failing processor or virtual machine.

HELP (Common)
Displays a summary of the DUMPSCAN subcommands.

HX (Common)
Ends the session and returns to CMS.

IPCSMAP (Common)
Adds an IPCS map to the dump being viewed.

Locate { string } fromhexloc tohexloc [increment] (Common)
Locate Up { X'string } [1]
Searches the dump for a particular string of data.

DUMPSCAN (continued)

IPCS

MAPA	hexloc	(Common)
	Locates the module that contains the address specified.	
MAPN	mmmmmmmm	(Common)
	Searches the load map for an entry point.	
Mregs		(CP dump only)
	Displays the registers, clocks, PSW, CSW, CAW and timers for the main or the IPL processor.	
MRIoblok	raddr	(CP dump only)
	Displays the RCHBLOK, RCUBLOK, and RDEVBLOK for the specified device attached to the IPL processor in an MP configuration.	
OSPoint		(CMS dump only)
	Displays the formatted contents of three pointers used in OS simulation.	
Print	{ subcommand }	(Common)
PRT	{ ON OFF CLOSE ? }	
	Prints the displayed data.	
QUIT		(Common)
	Ends the session and returns to CMS.	
Regs		(Common)
	Displays the registers, clocks, PSWs, timers, CSW, and CAW.	
RIOblok	raddr	(CP dump only)
	Displays the RCHBLOK, RCUBLOK, and RDEVBLOK for the specified cuu.	

DUMPSCAN

DUMPSCAN (continued)

IPCS

Scroll U [HEX] (Common)

ScrollU [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. The HEX/FORMAT subcommands are only valid following trace entry displays.

SYMP (Common)

Formats and displays the summary of the symptom record.

Trace [[FOR]count][FROM fromloc] [HEX] (CP and TSAF dumps only)
[FORMat]
[Scroll [U]]
[ScrollU]

Displays trace table entries in hexadecimal or formatted display. After the first successful invocation of the TRACE subcommand, you can specify either SCROLL operand to move forward or back through the trace table.

USERMAP (CMS dump only)

Adds a user load map to the dump being viewed.

VIOblok cuu [userid] (CP dump only)
[OPERATOR]

Displays the VCHBLOK, VCUBLOK, and VDEVBLOK for the specified cuu and userid.

Vmblok [userid] (CP dump only)
[SYSSPOOL]

Scans the VMBLOK chain and displays userid, VMBLOK address and status of all logged-on users. Displays formatted information from a specified user's VMBLOK by specifying the userid operand.

ECHO

CP Class G

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

Echo [nn]
[1]

EDIT

CMS

Invokes the VM/SP System Product editor in CMS edit or (EDIT) migration mode.

Note: In all formats of the CMS EDIT subcommands and macros, use of the word "subcommand" means a CMS EDIT subcommand only.

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

options:

```
[LRECL nn] [NODISP]
```

The format of each EDIT **subcommand and macro** is followed by its description:

```
$DUP   [1|n]
        Duplicates the current line. This is a macro.
```

```
$MOVE  n   { Up m
             Down m
             TO label }
        Moves n lines up m or down m lines. This is a macro.
```

```
?
        Displays the last executed EDIT subcommand except for the REUSE(=)
        or ? (question mark) subcommands.
```

```
ALter char1 char2  [ n   [G]
                    *   [*]
                    1 ]
        Scans n records, altering the specified character.
```

EDIT

EDIT (continued)

CMS

AUTOsave [n|OFF]

Saves the file after the indicated number of changes or displays the current setting.

BACKward [1|n]

Repositions the current line pointer toward the top of the file.

Bottom

Moves the current line pointer to the last line of the file.

CASE [U|M]

Translates to uppercase.

Change [/string1[/string2[/ $\begin{bmatrix} n \\ * \\ 1 \end{bmatrix}$ $\begin{bmatrix} G \\ * \end{bmatrix}$]]]

Changes string1 to string2.

CMS

Enters CMS subset command mode.

DELeTe [n|1|*]

Deletes n lines or to EOF.

DOWn [n|1]

Moves the current line pointer to the nth line down from the current line.

DString /[string /]

Deletes lines from the current line down to (but not including) the line that contains the designated string.

EDIT (continued)

CMS

FILE {fn [ft [fm]]}

Saves the file edited on disk and returns to CMS mode.

Find [line]

Searches the file for the specified line.

FMode {fn}

Resets or displays the filemode.

FName {fn}

Resets or displays the filename.

FORMat {DISPLAY|LINE}

Changes the mode of displaying data on a 3270 terminal from typewriter (line) style to full-screen display style or vice versa.

FOrward [1|n]

moves the current line pointer forward n lines.

Getfile {fn} [ft [fm [firstrec] [numrec]]]

Inserts some or all of the specified file.

¹ Operands are positional.IMAGE [ON|OFF|CANON]²

Expands text (including backspace and tab) into line images or displays current settings.

² ON is the default for all filetypes except SCRIPT. CANON is the default for SCRIPT files.

EDIT

EDIT (continued)

CMS

Input [line]
Inserts a line into the file or enters input mode.

LINEmode $\left[\begin{array}{|c|c|c|} \hline \text{LEFT} & \text{RIGHT} & \text{OFF} \\ \hline \text{L} & \text{R} & \\ \hline \end{array} \right]^3$

Sets or cancels line-number editing, or displays current setting.

³ Line-number editing is the default for USBASIC and FREEFORT files.

[Locate] | / [string [/]]
Scans the file for the first occurrence of a string.

LONG
Enters LONG error message mode. LONG is the default mode.

Next [n | 1]
Advances the current line pointer n lines toward the end of the file.

Overlay [line]
Replaces all or part of the current line.

PREserve
Saves current mode settings.

PROMPT [n | 10]
Sets the line increment for line-number editing.

QUIT
Terminates the EDIT session leaving the previous copy of the file intact.

EDIT (continued)**CMS****RECFM** [F|V]

Sets or displays record format.

RENUM $\left[\begin{array}{l} \text{strtno}[\text{incrn0}] \\ \underline{10} \quad [\underline{\text{STRTNO}}] \end{array} \right]$

Recomputes line numbers.

REPEAT [n|1|*]Executes the following OVERLAY request *n* times or to EOF.**REPLACE** [line]

Replaces the current line with "line" or deletes the line and enters input mode.

RESTORE

Restores mode settings.

RETURN

Returns to EDIT environment.

REUSE [subcommand]

=

Stacks (LIFO) the last EDIT subcommand and executes the stacked subcommands.

SAVE [fn [ft [fm]]]

Saves the file on disk.

EDIT

EDIT (continued)

CMS

{Scroll
S[croll]U[p]} [*|n|1]

Displays a number of lines above or below the current line.

SERial { OFF
ON [incr]
ALL [10]
seq }

Turns serialization on or off in columns 73-80.

SHORT

Enters SHORT error message mode.

STACK [n|1|0|subcommand]

Stacks *n* lines in the terminal input buffer.

TABSet n1 [n2 ... nn]

Sets the specified tabs.

TOP

Moves the current line pointer to the beginning of the file.

TRUNC [n|*]

Sets or displays the column of truncation.

Type $\left[\begin{array}{l} m \\ * \\ \underline{1} \end{array} \right] \left[\begin{array}{l} [n] \\ [*] \end{array} \right]$

Displays the specified number of lines beginning with the current line.

EDIT (continued)

CMS

Up [n|1]
Points to the line *n* lines above the current line.

Verify $\left[\begin{array}{l} \text{ON} \\ \text{OFF} \end{array} \right] \left[\left[\begin{array}{l} \text{startcol} \\ \underline{1} \end{array} \right] \begin{array}{l} \text{endcol} \\ * \end{array} \right]$

Sets, displays, or resets verify mode.

$\left\{ \begin{array}{l} X \\ Y \end{array} \right\}$ [subcommand|n|1]
Assigns to X or Y the given EDIT subcommand.

Zone $\left[\begin{array}{l} \text{firstcol} \\ * \\ 1 \end{array} \left[\begin{array}{l} \text{lastcol} \\ * \end{array} \right] \right]$

Sets or displays the columns to be edited.

$\left\{ \begin{array}{l} \text{nnnnn} \\ \text{nnnnnnnn} \end{array} \right\}$ [text]
Locates the line when using line-number editing.

ENABLE

CP Class A or B

Activates communication lines.

ENable $\left\{ \begin{array}{l} \text{raddr...} \\ \text{Laddr} \\ \text{SNA [userid]} \\ \text{ALL} \end{array} \right\}$

ERASE

ERASE

CMS

Deletes files from a user's disk.

```
ERASE {fn}{ft}{fm} [(options...)]
```

options:

[Type|Notype]

ESERV

CMS

In CMS/DOS, copies edited DOS/VSE macros from system or private source statement E sublibraries to CMS disk files; or lists de-edited macros.

```
ESERV fn
```

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 SSS SVC GTrace } [Group]
```

[ALL]

[END]

EXEC

CMS

Executes one or more CMS commands or EXEC control statements contained in a specified System Product Interpreter, CMS EXEC, or EXEC 2 file.

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

The formats of the **EXEC control statements** are followed by their descriptions:

&variable = {ae|string|function|X'xxxxxx}
Assigns a variable.

&ARGS [arg1 [arg2 ... [arg30]]]
Defines or redefines arguments to special variables &1-&30.

&BEGEMSG [ALL]
Displays the following unscanned error message lines at the terminal. The list of lines to be displayed must end with an &END control statement.

Note: See the &EMSG statement for the format of message lines.

&BEGPUNCH [ALL]
Punches the following lines into cards. End the list of lines with an &END control statement.

&BEGSTACK [LIFO|FIFO] [ALL]
Stacks lines into the terminal input stack. The list is terminated by the &END control statement.

&BEGTYPE [ALL]
Displays lines at the terminal. The list is terminated by the &END control statement.

&CONTINUE
Provides a branching address for EXEC branch statements.

EXEC

EXEC (continued)

CMS

&CONTROL

[OFF]	[TIME]	[NOMSG]	
	ERROR			<u>NOTIME</u>			<u>PACK</u>		<u>MSG</u>
	ALL								
	<u>CMS</u>								

Specifies the data to be displayed in the execution summary of an EXEC.

&EMSG mmmnnns [tok1...[tokn]]

Displays a line of tokens to be edited as an error message.

&END

END statement for action started by &BEGPUNCH, &BEGSTACK &BEGEMSG, or &BEGTYPE.

&ERROR [executable-statement | &CONTINUE]

Provides error processing.

&EXIT [return-code | 0]

Exits from the EXEC file with a specified return code.

&GOTO {TOP | line-number | -label}

Transfers control to a specified location.

&HEX {ON | OFF}

Initiates or terminates hexadecimal conversion in an EXEC procedure.

&IF {tok1} operator {tok2} executable-statement
 {&\$} {&\$}
 {&*} {&*}

Allows statement execution if the comparison is satisfied. If the comparison is invalid, execution continues with the statement following the &IF statement.

&LOOP { n } { m }
 {-label} {condition}

Describes a loop in an EXEC, and conditions for exit from the loop.

EXEC (continued)

CMS

&PUNCH [tok1 [tok2 ... [tokn]]]

Punches a card with the specified tokens.

&READ $\left[\begin{array}{l} n \\ \text{ARGS} \\ \text{VARS}[\&\text{var1}[\&\text{var2} \dots [\&\text{varn}]]] \\ \underline{1} \end{array} \right]$

Reads the next line (or lines) from the terminal stack.

&SKIP [n|1]

Skips subsequent statements.

&SPACE [n|1]

Displays blank lines at the terminal.

&STACK $\left[\begin{array}{l} \text{FIFO} \\ \text{LIFO} \end{array} \right] \left[\begin{array}{l} \text{tok1}[\text{tok2} \dots [\text{tokn}]] \\ \text{HT} \\ \text{RT} \end{array} \right]$

Places a line of tokens in the console input stack.

&TIME [ON|OFF|RESET|TYPE]

Displays time information on the terminal after each CMS command is executed.

&TYPE [tok1 [tok2 ... [tokn]]]

Displays the specified tokens at the terminal.

EXEC

EXEC (continued)

CMS

The formats of **EXEC built-in functions**, followed by their descriptions, are:

`&variable = &CONCAT tok1[tok2 ... [tokn]]`
Concatenates two or more tokens and assigns the result to a variable symbol.

`&variable = &DATATYPE token`
Determines whether the specified token is alphabetic or numeric.

`&variable = &LENGTH token`
Indicates number of non-blank characters in the specified token.

`[...] &LITERAL token [...]`
Allows the use of the literal value of the token without symbolic substitution.

`&variable = &SUBSTR token i [j]`
Extracts a character string from the specified token.

The formats of **EXEC special variables**, followed by their descriptions, are:

`&n`
Represents the numeric variables &1 through &30.

`&*` and `&$`
Used to perform a collective test on all arguments passed to EXEC.

`&0`
Contains the filename of the EXEC file.

`&DISKx`
Used to determine whether a disk is an OS, DOS, or CMS disk.

`&DISK*`
Contains the mode letter of the first read/write disk in the CMS search order (or contains the word NONE).

`&DISK?`
Determines which of the user's accessed read/write disks has the most space.

`&DOS`
Indicates the current status of the CMS/DOS environment (ON or OFF).

`&EXEC`
Indicates the filename of the EXEC file.

EXEC (continued)

CMS

&GLOBAL

Indicates the recursion level of the EXEC procedure that is currently executing (maximum 19 levels).

&GLOBALn

Represents the numeric variables &GLOBAL0 through &GLOBAL9.

&INDEX

Indicates the number of arguments passed to the EXEC procedure (maximum - 30).

&LINENUM

Indicates the current line number in the EXEC file.

&READFLAG

Indicates the word STACK if there are lines stacked in the terminal input buffer, or the word CONSOLE if a read is to be issued to the terminal.

&RETCODE

Indicates the return code from the most recently executed command.

&TYPEFLAG

Indicates one of two literal values: RT (resume typing) or HT (halt typing).

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

EXEC 2

CMS

Invokes EXEC 2 files.

```
EXEC filename [arg1 [arg2 ... argn]]
```

The formats of the **EXEC 2 predefined variables**, followed by their descriptions, are:

- &**
Initializes to its own name. This variable is initialized or maintained automatically.
- &0**
Initializes to the first word of the command string passed to the EXEC 2 interpreter. This variable is initialized or maintained automatically.
- &1, &2, ...**
Initializes to the arguments arg1, arg2, ..., argn 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.
- &BLANK**
Assigns the value of a blank.
- &CMDSTRING**
Initialized to the untranslated command string that is 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.)

EXEC 2 (continued)**CMS****&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 invoked, or is zero.

&N, &INDEX

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

&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

EXEC 2 (continued)

CMS

The formats of the **EXEC 2 control statements**, followed by their formats are:

&ARGS [word1 [word2 ...]]

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

&BEGPRINT
&BEGTYPE

$$\left[\begin{array}{l} n \\ \text{label} \\ \underline{1} \\ * \end{array} \right] \left[\begin{array}{l} k \\ * \\ - \end{array} \right]$$

line1
line2

. . .

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

&BEGSTACK

$$\left[\begin{array}{l} n \\ * \\ \text{label} \\ \underline{1} \end{array} \left[\begin{array}{l} k \\ * \\ - \end{array} \left[\begin{array}{l} \text{FIFO} \\ \text{LIFO} \end{array} \right] \right] \right]$$

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.

EXEC 2 (continued)

CMS

```
&BUFFER  n  [comment]
        *
```

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

```
&CALL  line-number [arg1 [arg2...]]
        label
```

Invokes 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.

```
&COMMAND word1 [word2 ...]
        Issues the command comprising of word1, word2,..., each with one space between.
```

```
&CRASH [text]
        Aids debugging the EXEC 2 interpreter DMSEX. Intended to to be used by system support people only.
```

```
&DUMP  ARGS
        VAR[S] [var1 [var2 ... ]]
        Prints a line for each &1, &2, ..., &n argument or variables var1, var2, ..., varn.
```

EXEC 2

EXEC 2 (continued)

CMS

&ERROR action

Sets the action to be automatically invoked 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
 ¬=|NE
 <=|LT
 <=|¬>|LE|NG
 >|GT
 >=|¬<|GE|NL [word2 executable statement]

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

EXEC 2 (continued)

CMS

```
&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 ] ]
```

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.

EXEC 2

EXEC 2 (continued)

CMS

&RETURN [word] [comment]

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

&SKIP [1 [comment]]

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 $\left[\begin{array}{l} \text{FIFO} \\ \text{LIFO} \end{array} \right] \left[\text{word1} \left[\text{word2} \dots \right] \right]$

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 $\left[\begin{array}{l} \text{ON} \\ \text{ERR} \\ \text{ALL} \\ \text{OFF} \\ * \\ \text{—} \end{array} \right] \left[\text{output-action} \right]$

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.

EXEC 2 (continued)

CMS

&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.

The formats of the **EXEC 2 predefined functions**, followed by their descriptions, are:

&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.

EXEC 2

EXEC 2 (continued)

CMS

&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 non-blank 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 $\left[\begin{array}{c} j \\ * \end{array} \right]$
&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.

EXEC 2 (continued)

CMS

&RANGE OF stem i j

Yields a string comprising 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 non-blank character.

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

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]

If word has trailing blanks, removes them. 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 **EXEC 2 User-Defined function**, followed by its description, is:

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

Invokes 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

EXECDROP

CMS

Removes the specified EXEC(s) and Editor Macros(s) from storage or discontinues use of the specified EXEC(s) and Editor Macro(s) in an Installation Discontiguous Shared Segment (DCSS).

EXECDrop

EXDrop

$\left\{ \begin{array}{l} \text{execname} \\ * \end{array} \right\} \left[\begin{array}{l} \text{exectype} \\ * \\ - \end{array} \right] \left[(\text{options...}) \right]$

options:

[User]

[System]

[SHared]

EXECIO

CMS

Do I/O operations between a device and the program stack.

```
EXECIO {lines} {
  *
  DISKR fn ft [fm [linenum]] [([FINIS]
    options [a] [b] [])]
  CARD [( options [a] [b]) []]
  CP [( options [a] [b] [d]) []]
  DISKW fn ft fm [linenum
    [recfm [lrec1]]] [([FINIS]
    options [b] [c] [d][e]) []]
  PUNCH [( options [b] [c] [d]) []]
  PRINT [( [CC code] options [b] [c] [d])
    [DATA] ]
  MSG [( options [b] [c] [d]) []]
}
```

option formats:

(a)	(b)	(c)	(d)	(e)
[FInd /chars/]	[Margins {n1 n2}]	[Case {U}]	[String xxx...]	[BUffer length]
[LOcate /chars/]	[{ <u>1</u> *}]	[{ <u>M</u> }]		
[AvoId /chars/]	[STRIP]			
[Zone {n1 n2}]	[NOTYPE]			
[{ <u>1</u> *}]				
[LIFO]	[STEM xxxx]			
[FIFO]	[VAR xxxx]			
[SKip]				

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.

EXECLOAD

EXECLOAD

CMS

Loads the specified EXEC or Editor Macro into storage and prepares it for execution.

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

options:

```
[User
SYstem]
```

```
[Push]
```

EXECMAP

CMS

Provides a list of the EXEC(s) and Editor Macro(s) in storage and in an Installation Discontiguous Shared Segment (DCSS).

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

options:

```
[User]
[SYstem ]
[SHared]
```

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

EXECOS

CMS

Resets the OS, VSAM, and Vector environments under CMS without returning to the interactive environment. It can be invoked without specifying parameters or by *preceding* any CMS command with EXECOS.

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

EXECSTAT

CMS

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

- 0 - EXEC is in storage, and register 1 contains pointer to the fileblock.
- 4 - EXEC is not in storage but does exist on dasd, and register 1 contains pointer to FST.
- 8 - EXEC is not in storage and does not exist on dasd.

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

EXECUPDT

CMS

Applies updates to a System Product interpreter source program and creates an executable version of the program. It can only be used with System Product interpreter programs.

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

options:

```
[CTL fn1] [HISTory] [COMPress] [COMMENTS] [ETMODE]  
          [NOHISTory] [NOCOMPress] [NOCOMMENTS]  
[SID] [NOUPdate]  
[NOSID]
```

EXPAND

EXPAND

CMS

Lets you add space to a program in object deck form.

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

options:

```
[ INPUT filename ] [PRINT|NOPRINT]
[ CSECT csect SIZE size ]
```

Control Statements

```
EXPAND csect size [, csect size...]
```

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

Loads an executable phase into storage for execution.

```
FETCh phasename [(options...{})]
```

options:

[START]

[COMP]

[ORIGIN hexloc]

FILEDEF

FILEDEF

CMS

Simulates OS JCL (Job Control Language) data definition (DD) statements. Displays current definitions when entered without operands.

```
Filedef { ddname } {
          nn
          *
Terminal [(optionA optionD[])]

Printer [(optionA OPTCD J [])]
Punch  [(optionA [])]
Reader [(optionA [])]

DISK [ fn ft [ fm ] ] [(optionA optionB [])]
      [ FILE ddname [ A1 ] ]
or
[ DISK [ [ fn ft ] [ fm ] ]
  DISK [ [ FILE ddname ] [ A1 ] ]

      DSN ?
      DSN qual1 qual2...
      DSN qual1.qual2...
[ option A optionB() ]

or
DISK vaddr
DUMMY [(optionA[])]

TAPn [ LABOFF
      BLP[n]
      SL[n] [VOLID valid][DISP MOD] [(optionG[])]
      SUL[n] [VOLID valid]
      NSL filename
      NL[n]
      [(optionA optionC optionE optionF[])]

GRAF cuu [([PERM][CHANGE|NOCHANGE])]

CLEAR
```

FILEDEF (continued)

CMS

optionA:

```
[PERM] [CHANGE|NOCHANGE]
[RECFM F|FB|V|VB|U|VS|VBS|FS|FBS|A|M ]
[LRECL nnnnn] [ BLOCK|BLKSIZE nnnnn]
```

optionB:

```
[KEYLEN nnn] [XTENT nnnnn|50 ] [CONCAT]
[LIMCT nnn] [OPTCD a]
[DISP MOD] [MEMBER membername]
[DSORG {PS|PO|DA}]
```

optionC:

```
[7TRACK|9TRACK|18TRACK] [TRTCH 0|QC|OT|E|ET ]
[DEN {200|556|800|1600|6250|38K}]
```

optionD:

```
[UPCASE|LOWCASE]
```

optionE:

```
[LEAVE]
[NOEOV]
```

optionF:

```
[ ALT { TAPn }
  { cuu } ]
```

optionG:

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

FILEDEF

FILEDEF

GCS

Defines CMS format files and spool files.

```
FILEdef {
  { ddname } {
    Printer [optionA OPTCD ;()]
    PUnch [(optionA())]
    Reader
    DISK { fn ft [fm] [(optionA optionB())]
          [FILE ddname [A1]]
    }
    DUMMY [(optionA())]
    CLEAR
  }
}
```

optionA:

[PERM]

[CHANGE
NOCHANGE]

[RECFM a]

[LRECL nnnnn]

[BLOCK nnnnn
BLKSIZE nnnnn]

optionB:

[DISP MOD]

[DSORG PS]

Note: The above operands work in the same manner as 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 disk files, with the ability to edit and issue commands from the list.

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

options:

[Append]

[Filelist | Nofilelist]

[PROFile fn]

FINIS

CMS

Closes one or more files.

```
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

CP Class A

Forces logoff of the named user.

```
FORCE  userid
```

FORCE

FORCE

RSCS

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

FORCE linkid

FORMAT

CMS

Formats a disk for use by CMS, counts or resets the number of cylinders on a disk, or writes a label on a virtual disk.

FORMAT cuu mode $\left[\begin{array}{l} \text{nocyl} \\ \text{noblk} \end{array} \right] [(\text{options...})]$

options:

	512
<u>Blksize</u>	800
	1024
	2048
	4096
	1K
	2K
	4K

[Noerase]

[Label]

[Recomp]

FORMAT-ALLOCATE

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,startadr,endadr,volser

●Allocate Function

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

●Label Function

FORMAT,devadr,devtype,volser,LABEL

FREE

CP Class D

Releases previously held user spool files.

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

FREE

RSCS

Resumes transmission on a communication link previously in HOLD status.

FRee [linkid]

FWDSpace

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

$\left[\begin{array}{c} \text{hexloc1} \\ \underline{0} \end{array} \right] \left[\left\{ \begin{array}{c} - \\ : \end{array} \right\} \left[\begin{array}{c} \text{hexloc2} \\ \underline{END} \end{array} \right] \right] \left[\begin{array}{c} \text{TO *} \\ \text{TO userid} \end{array} \right] \left[\text{DSS} \right] \left[\begin{array}{c} \text{FORMAT type} \\ \text{GCS} \end{array} \right]$
.bytecount

GENDIRT

CMS

Creates auxiliary module directories.

GENDIRT directoryname [targetmode[source mode]]

GENIMAGE

CMS

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

GENIMAGE $\left[\begin{array}{c} \text{fn} \\ \underline{\text{SYSIN}} \end{array} \right] \left[\begin{array}{c} \text{ft} \\ \underline{\text{FILE}} \end{array} \right] \left[\begin{array}{c} \text{fm} \\ * \end{array} \right] \left[\begin{array}{c} \text{sfn} \\ \underline{\text{SYSPRINT}} \end{array} \right]$
 $\left[\begin{array}{c} \text{sft} \\ \underline{\text{LISTING}} \end{array} \right] \left[\begin{array}{c} \text{sfm} \\ \underline{\text{A1}} \end{array} \right]$

GENMOD

CMS

Generates a nonrelocatable (MODULE) file on a CMS disk.

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

options:

```
[MAP ] [STR ] [FROM entry1] [SYSTEM] [ OS ]  
[NOMAP] [NOSTR] [TO entry2] [DOS ]  
[ ALL ]
```

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]
```

GEN3705

CMS

Invokes 3705 Stage 2 service aid.

```
GEN3705 fname ftype [fmode] [(options...)]
```

options:

```
[RUN ] [SAVE ]  
[NORUN] [NOSAVE]
```


GET VSCREEN

GET VSCREEN

CMS

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

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

GLOBAL

CMS

Specifies CMS or CMS/DOS libraries to be searched for macros, copy files, subroutines, or DOS executable phases when processing CMS commands.

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

GLOBAL

GCS

Defines the CMS load libraries you want searched for modules.

```
GLOBAL      LOADLIB [libname1 ... libname63]
```

GLOBALV

CMS

Set, maintain, and retrieve a collection of named variables.

GLOBALV

```

INIT
SELECT {group
        UNNAMED}

        SET
        SETS {name1[value1][name2 value2]...}
        SETP

        SETL
        SETLS
        SETSL {name [value]}
        SETLP
        SETPL

        LIST [name1 [name2]...]

        STACK {name1 [name2]...}
        STACKR

[SELECT group
 UNNAMED] PUT
          PUTS {name1 [name2]...}
          PUTP

          GET [name1 [name2]...]

SELECT {group
        UNNAMED} PURGE

GRPLIST
GRPSTACK
PURGE

```

Note: Although this command may be used in CMS EXECs, it is designed for use with EXEC 2 or REXX EXECs. Restrictions/precautions are listed in "Usage Notes for the CMS EXECs."

GTRACE

GTRACE

RSCS

Provides additional tracing through VM/GCS.

GTrace $\left[\begin{array}{l} \text{ON} \\ \text{OFF} \end{array} \right]$

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 commands, instructions, and messages.

```

Help [ TASKs
      Help
      taskname TASKs
      menuname MENU
      component-name command-name ] [ ([optionA][optionB][optionC][ ])]

[ MESSAGE ] message-id
[ MSG ]

options:
optionA: [ BRIef
           DETail
           RELated ]

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

optionC: [ SCReen
           NOScreen ] [ TYPe
                       NOType ] [EXTend]

```

HELPCONV

CMS

Converts the specified file into a formatted HELP file.

```

HELPCONV filename filetype [filemode]
                          *

```

HI

CMS Immediate Command

Halt interpretation 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

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 a 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

Display or stack userid, nodeid, rscsid, date, time, time zone and day of week.

Identify [(options...)]

options:

STACK	<u>FIFO</u>
	LIFO
FIFO	
LIFO	
<u>TYPE</u>	

IMAGELIB

CMS

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

IMAGELIB namedsys

IMAGEMOD

IMAGEMOD

CMS

Allows changes to the 3800 named systems.

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

options:

```
{ TERM }  
{ PRINT }  
{ DISK }
```

IMMCMD

CMS

Establishes or cancels Immediate commands from within an EXEC. Determines whether a particular Immediate command has been established or if it has been issued by the terminal user.

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

INCLUDE

CMS

Brings additional TEXT files into virtual storage.

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

options:

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

```
[ TYPE ] [ MAP ] [ INV ] [ REP ] [ AUTO ] [ LIBE ]
[ NOTYPE ] [ NOMAP ] [ NOINV ] [ NOREP ] [ NOAUTO ] [ NOLIBE ]
```

```
[ DUP ] [ HIST ] [ SAME ] [ RLDSAVE ]
[ NODUP ] [ 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

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

Reads the dump from the virtual reader, creates a CMS file containing the symptom record and dump, and creates a problem report by extracting pertinent data from the dump.

IPCSDUMP

IPL

CP Class G

Simulates an initial program load function for a virtual machine.

Ipl { vaddr { cylno } { CLear } [STOP] [ATTN] [PARM p1[p2...]]
 { nnnnnn } { NOCLear }
 systemname }

optional CMS PARMs

AUTOCR

BATCH

INSTSEG [YES|NO|name]

NOSPROF

SAVESYS sysname (Class E only)

ITASK

CMS

Invokes other EXECs and commands to do most of the steps in the installation procedure.

ITASK	LOAD	{ CP CMS GCS HELP HPO IPCS TSAF CMSSRC CPSRC GCSSRC IPCSSRC TSAFSRC CMSFTSRC }	
	LANG	{ ALL ALLOBJ CP CMS TSAF HELP CMSSRC }	
	BUILD	{ CP CMS GCS }	{ [noassem] [groupname] }
	ASSEMBLE	{ ALLCP BMKBOX DMKFCB DMKRIO DMKSNT DMKSYS DMSNGP }	
	ALLOCATE		
	BASEIDS		

ITRACE

ITRACE

GCS

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

```
ITrace { [ [ GTrace ] ] }
        [ [ SUP ] ]
        [ [ ALL ] ] [OFF] [Group]
        [ [ 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.

```
LLabeldef { { * } }
           [ { fn } ]
           [ CLEAR
             [ [FID(?|fid)][VOLID (?|valid|SCRATCH)]
             [VOLSEQ volseq][FSEQ fseq]
             [GENN genn][GENV genv]
             [CRDTE yyddd][EXDTE yyddd]
             [SEQ {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 DCSS named NLSxy, 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 DCSS 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]
```

Note: 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.

LISTDS

CMS

Displays information about data sets, files, extents, or free space.

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

options:

```
[ FREE ] [ FORMAT ]  
[ EXTENT ] [ PDS ]
```

LISTFILE

LISTFILE

CMS

Lists information about CMS files.

```
Listfile [fn [ft [fm]]] [(options...{})]
```

options:

```
[Header ] [Exec [Trace] [ARGS] ] [FName ] [Blocks]
```

[NOHeader]	[Trace[ARGS]	[FType	
	APPend[ARGS]	FMode	[%x]
	STACK[<u>FIFO</u> LIFO]	FOrmat	
	FIFO	ALloc	
	LIFO	DaTe	
	XEDIT	LABEL	

LISTIO

CMS

Displays a list of current assignments for system and/or programmer logical units in a virtual machine.

```
LISTIO [SYS  
PROG  
SYSxxx  
A  
UA  
ALL ] [(options...{})]
```

options:

```
[EXEC ] [STAT]
```

[APPEND]

LKED

CMS

Used to create a CMS LOADLIB or LOADLIB member.

LKED fn [(options...)]

options:

[NCAL] [LET] [ALIGN2] [NE] [OL] [RENT]

[REUS] [REFR] [OVLY] [XCAL]

[NAME membername] [LIBE libraryname]

[XREF]	[PRINT]	[TERM]
[MAP]	[DISK]	[NOTERM]
[LIST]	[NOPRINT]	

[SIZE	{	value1 value2	}
		value1	
		value1,	
		,value2	
		,	

LOAD

CMS

Brings TEXT files into storage and establishes links.

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

options:

[CLEAR]	[RESET {entry}]	[MAP]	[INV]	[REP]	[START]
[NOCLEAR]	[*]	[NOMAP]	[NOINV]	[NOREP]	

[TYPE]	[AUTO]	[LIBE]	[ORIGIN {hexloc}]	[DUP]	[RLDsave]
[NOTYPE]	[NOAUTO]	[NOLIBE]	[TRANS]	[NODUP]	

[HIST]
[NOHIST]

LOADBUF

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

Lists, copies, or compresses a CMS LOADLIB.

```
LOADLIB [ LIST  
        COMPRESS { fileid1 } ] [fileid2[fileid3]] [(options...{})]  
        COPY
```

options:

```
[ TERM ] [ REPLACE ]  
[ PRINT ] [ MODIFY ]  
[ DISK ]
```

SYSIN control statements (for COPY only):

```
[SELECT|EXCLUDE]
```

LOADMOD

CMS

Brings a single MODULE file (in nonrelocatable format) into storage.

```
LOADMod fn [ MODULE [ fm ] ]  
          [ * ]
```

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 or 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 [HOLD]  
LOGout
```

LOGON

CP Class Any

Initiates all virtual machine operation.

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


M (CMS Border Command)

M

CMS Border Command

Changes the location of the window.

M

MACLIB

CMS

Creates and updates macro libraries.

```
MAClib { {GEN}
        {ADD} libname fn1[fn2...]
        {REP}
        DEL libname membername1[membername2...]
        COMP libname
        MAP libname [membername1[membername2...]][(options...)] }
```

options:

```
[ DISK
  PRINT
  TERM
  XEDIT
  STACK [FIFO]
         [LIFO]
  FIFO
  LIFO ]
```

MACLIST

CMS

Display a list of information about all members in the specified macro library with the ability to edit and issue commands against the members from the list.

```
MACLIST
Mlist      libname [(options ())]
```

options:

```
[Append] [Compact] [PROfile fn]
          [NOCompact]
```

MAKEBUF

CMS

Creates a new buffer within the program stack.

MAKEBUF

MAP

IPCS

Converts various types of load maps into the proper format for use by IPCS.

MAP type [Prompt]

MAXIMIZE WINDOW

CMS

Expands a window to the physical screen size.

MAXimize WINDOW { wname }
=

MESSAGE

CP Class A or B

Sends text messages to other users, system operator or self.

Message { ALL
Msg { userid
*
Operator } } msgtext

MESSAGE

CP Class Any

Sends text messages to other users, system operator or self.

Message { userid
MSG { *
Operator } } msgtext

MIGRATE

MIGRATE

CP Class A

Activates normal 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
= }

MODMAP

CMS

Displays a MODULE file load map.

MODmap fn

MONITOR

CP Class A or 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 } }

MONITOR (continued)

CP Class A or E

```

{
  ENable {
    PERForm
    RESPonse
    SChedule
    USER
    INSTsim
    DASTap
    SEEKS
    SYSprof
  } 1
}
  
```

¹ Select one or more of the classes, subject to the restrictions listed with the ENABLE operand.

```

{
  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
        1600
        6250
        38K
      } } [BUFFS n]
  }
}
  
```

MONITOR

MONITOR (continued)

CP Class A or E

}	STOP	}	} ²
}	TIME	}	[FROM h1:m1 TO h2:m2]
			[FOR hh:mm]
			[ALL]
			[NONE]

² See the operand description for the default values.

MOREHELP

CMS

Displays additional information, if available, for a previously-issued valid HELP command.

MOREhelp [(optionA optionB [])]

optionA: [BRIef
DETail
RELated]

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

MOVEFILE

CMS

Moves data from one device to another device.

MOVEfile [inddname [outddname]] [(PDS{})]
 [INMOVE [OUTMOVE]]

MSG

RSCS

Sends a message line to a local or remote operator or user.

```
Msg nodeid {userid|SYSTEM} [msgtext ]
```

MSGNOH

CP Class A or B

Allows a virtual machine to send messages without the standard header associated with the MESSAGE command.

```
MSGNOH {userid} msgtext
        {
        ALL
        *
        }
```

N

CMS Border Command

Minimizes the window.

```
N
```

NAMEFIND

CMS

Display/stack information from a NAMES file. (default 'userid NAMES').

```
NAMEFind :tag value [:tag [value]]... [options...()]
```

options:

STACK [n * <u>1</u>]	[FIFO LIFO]	[File fn]
FIFO [n * <u>1</u>]		[LINenum]
LIFO [n * <u>1</u>]		[START recnum]
TYPE [n * <u>1</u>]		[Size[n * <u>8</u>]]	
		[XEDIT]	

NAMES

NAMES

CMS

Display 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 3705 dumping operations.

NCPDUMP [DUMPxx] [[ERASE] [NOFORM] [NCPBUFF][]]]

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 hexlocl [{ - } [hexloc2]
 { : } [END]] }
 [.] [bytecount]
 [END] }

DUMP raddr [IMMED
 OFF
 AUTO]

ENable [ALL
resid [resid...]

LOAD raddr ncpname

POLLday 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] 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...] ]

VARY { ONline } resid [resid...]
     { OFFline }
```

NETWORK

RSCS

Starts or ends communications with ACF/VTAM. (For RSCS operator only)

```
NETwork { [APPLid name]
          [STArt [Pass password]
            [RETry nn]
          ]
          [HALT [QUICK]]
        }
```

NOTE

CMS

Prepare a 'note' for one or more computer users, to be sent via the SENDFILE command.

```
NOTE [name...[CC:name...]][(options...{})]
```

options:

```
{ [ACK] [NOAck] } [NOTebook fn] [NOTebook *] [NONotebook] { [LOG] [NOLog] } [LONG] [Short]
```

```
[Add] [Cancel] [Replace] [PROFile fn]
```

NOTREADY

CP Class G

Simulates loss of ready status on virtual device.

```
NOTReady vaddr
```

NUCXDROP

CMS

Deletes specified nucleus extensions.

```
NUCXDROP { name1 [name2...]
          *
        }
```

NUCXLOAD

NUCXLOAD

CMS

Loads a nucleus extension.

```
NUCXLOAD { name [fn] } [ ([System] [Service]
                        name member ddname) [ [ENdcmd] [IMmcmd] [Push] [] ] ]
```

NUCXMAP

CMS

Identifies existing nucleus extensions.

```
NUCXMAP [([STACK] [LIFO] [FIFO] [ ])]
```

O

CMS Border Command

Restores the window.

o

OPTION

CMS

In CMS/DOS, changes any or all of options in effect for the DOS/VS COBOL compiler. Only specified options are changed.

```
OPTION [options...]
```

options:

```
[ DUMP ] [ DECK ] [ LIST ] [ LISTX ] [ SYM ]
[ NODUMP ] [ NODECK ] [ NOLIST ] [ NOLISTX ] [ NOSYM ]

[ XREF ] [ ERRS ] [ 48C ] [ TERM ]
[ NOXREF ] [ NOERRS ] [ 60C ] [ NOTERM ]
```

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 {userid} {Reader|RDR} {Class c1 Class c2... }  
      {SYSTEM} {Printer|PRT} {spoolid1 spoolid2... }  
      { * } {PUnch|PCH} {FORM form1 FORM form2 ... }  
      { } {DEST dest1 DEST dest2 ... }
```

Note: 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... }  
      {Printer} {FORM form1 FORM form2... }  
      {PUnch} {spoolid1 spoolid2... }  
      { } {DEST dest1 DEST dest2... }
```

ORDER

RSCS

Reorders files enqueued on a specific link .

```
ORDER [linkid] spoolid1 [spoolid2 ...]
```

OSRUN

CMS

Executes a load module from a CMS LOADLIB or an OS module library.

```
OSRUN member [PARM=parameters]
```

OSRUN

OSRUN

GCS

Starts a GCS application program.

```
OSRUN      member [PARM=parameters]
```

OVERRIDE

CMS

Implements changes to the class structure.

```
OVERRIDE  fn  [ ft      ]  [ fm  ]  [ (EDIT ]  
           [ OVERRIDE ]  [ *  ]  [ (FREE ]
```

Control Statements:

```
DESTINATION cuu devtype volser altcuu
```

```
command [Type=c] Class = { classes }  
                        *
```

P

CMS Border Command

Pops the window.

```
P
```

PARSECMD

CMS

Use PARSECMD to invoke the parsing facility from an exec.

```
PARSECMD  uniqueid [ (options ... [])]
```

options:

```
[ TYPE ] [APPLID applid] [STRING cmdstring]  
[ NOTYPE ]
```

PEEK

CMS

Display a file that is in your virtual reader without reading it onto disk.

```
PEEK [spoolid] [(options...{})]
```

options:

```
[FROM recno]
[FOR numrec]
[PROFile fn]
```

PER

CP Classes 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.

event types:

```
PER [ Instruct [[DATA] hex-data] [options] ]
    [ BBranch [[INTO] into-addr-range] ]
    [ STore [INTO] storage-addr-range [INTO] addr [DATA] hex-data ]
    [ Mask [INTO] addr [DATA] mask-field ]
    [ G[reg1] { [-] [reg2] } [[DATA] hexword] ]
    [ [.] [regcount] ] ]
```

PER

PER (continued)

CP Classes A,B,C,D,E,F, AND G

options:

Range instruction-addr-range
FRom instruction-addr-range

PAss $\left[\begin{array}{c} 0 \\ n \end{array} \right]$

CMd {text}

$$\left[\begin{array}{l} \text{Printer} \quad \left[\text{RUN} \right] \\ \left[\begin{array}{l} \text{TERminal} \\ \text{BOth} \end{array} \right] \left[\begin{array}{l} \text{NORun} \\ \text{RUN} \end{array} \right] \\ \text{STEP} \left[\begin{array}{c} 1 \\ n \end{array} \right] \end{array} \right]$$

GUESTR
GUESTV
DATOFF
DAT

sub-commands:

COUNT
TABLE
SAVE traceset-name [APPEND]
GET traceset-name [APPEND]
END $\left\{ \begin{array}{l} \text{ALL} \\ \text{COUNT} \\ \text{CURRENT} \\ \text{ELEMENT-NUMBER} \\ \text{EVENT-TYPE} \\ \text{TRACESET NAME} \end{array} \right\}$

POP WINDOW

CMS

Moves a window up in the order of displayed windows.

```
POP WINDOW {wname} [ n ]  
           { WM } [ * ]
```

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 field in a symptom record for a specific problem number 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  
}
```

Note: The sequence of nnnnn and the other keywords can be reversed. However, nnnnn is a more recent problem than mmmmm.

PRELOAD

PRELOAD

CMS

Collects multiple text files and reformats them into a single text file.

```
PRELOAD loadlist [control]
```

PRINT

CMS

Directs a specified spool file to the virtual printer.

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

options:

```
[(OVerse)]  
[CC [HEADer] ] [UPCASE] [TRC  
[NOCC] ] [NOTRC]  
[LINECOUN {nn} ] [MEMBER { *  
[55} ] [membername} ] [HEX]
```

PROB

IPCS

Creates problem reports and adds information to an existing problem report.

```
PROB
```

PROGRAMMABLE OPERATOR

PROGRAMMABLE OPERATOR

CMS

Anyone, authorized by the active routing table, may 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 issued. Use the CMS EXEC, PROPST EXEC, to invoke 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  { nickname          } message
       { userid [AT node] }
```

The **formats of the programmable operator commands** are followed by their descriptions:

```
CMD  vmcmd
      Executes selected CP or CMS commands in the programmable operator's virtual machine.
```

PROGRAMMABLE OPERATOR

PROGRAMMABLE OPERATOR (continued)

CMS

FEEDBACK text...

FB

Places comments about the operation of the system and/or the programmable operator in the feedback file.

GET { FEEDBACK
FB
LOG [yyymmdd] }

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

Indicates node-checking status.

QUERY LGLOPR

Indicates the userid and nodeid of the currently assigned logical operator.

QUERY LOGGING

Indicates logging status.

QUERY PROPCHK

Indicates node-checking status.

QUERY RTABLE

Indicates the name of the programmable operator's active routing table.

PROGRAMMABLE OPERATOR

PROGRAMMABLE OPERATOR (continued)

CMS

SET DEBUG { ON }
 { OFF }

Enter and exit from programmable operator DEBUG mode.

SET HOSTCHK { ON }
 { OFF }

Restarts or halts checking of the host system by the distributed system.

SET LOGGING { ON }
 { OFF }
 { ALL }

Causes the programmable operator facility to stop writing any messages to the log file.

SET PROPCHK { ON } [nodeid]
 { OFF } [nodeid]

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.

PRTDUMP

IPCS

Formats and/or prints the symptom record on the first page with a disk dump file previously processed by IPCSDUMP.

PRTDUMP PRBnnnnn [option...]

CP DUMP options:

[NOFORM] [NOREAL] [NOVIRT] [NOHEX] [NOMAP]

CMS DUMP options:

not available; standard print routine is used.

partial DUMP options:

none; standard print routine is used.

PSERV

PSERV

CMS

In CMS/DOS, copies, displays, prints, or punches a procedure from the DOS/VSE procedure library.

PSERV procedure [ft|PROC] [(options...{})]

options:

[DISK] [PRINT] [PUNCH] [TERM]

PUNCH

CMS

Directs a specified spool file to the virtual punch.

PUnch fn ft [fm|*] [(options...{})]

options:

$\left[\begin{array}{l} \underline{\text{HEADER}} \\ \text{NOHEADER} \end{array} \right] \left[\begin{array}{l} \text{MEMBER} \{ * \\ \text{membername} \} \end{array} \right]$

PURGE

CP Class D

Deletes a closed spool file before reading, printing, or punching occurs. (A combination of CLASS and spoolid specifications may be entered.)

$\left. \begin{array}{l} \text{PURge} \quad [\text{FORCE}] \left[\begin{array}{l} \text{userid} \\ \text{SYSTEM} \\ * \end{array} \right] \left\{ \begin{array}{l} \text{Reader|RDR} \\ \text{Printer|PRT} \\ \text{PUnch|PCH} \\ \\ \text{ALL} \end{array} \right. \left. \left[\begin{array}{l} \underline{\text{ALL}} \\ \text{Class c1 CLassc2...1} \\ \text{spoolid1 spoolid2 ...} \\ \text{FORM form1 FORM form2 ...} \\ \text{DEST dest1 DEST dest2 ...} \end{array} \right] \right\} \right\}$

¹ Purging may be done using a combination of CLASS c, FORM and spoolid specifications.

PURGE

CP Class G

Deletes a closed file before reading, printing, or punching occurs. (A combination of CLASS, FORM, and spoolid specifications may be entered.)

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

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 scrollable data area of a virtual screen to a CMS file.

```
PUT VScreen vname fn ft [ fm [ fromlin [ numlin ] ] ]
                        [ * [ 1 [ * ] ] ]
                        [ A1 [ 1 [ * ] ] ]
```

QUERY

QUERY

CMS

In CMS or CMS/DOS mode, provides information about the virtual machine environment.

Query

```
ABBREV
APL
AUTOREAD
BLIP
BORDER wname [ALL]
CHARMODE
CMSLEVEL
CMSPF [ nn
      *
      - ]
CMSTYPE
CURSOR [vname]
DISK [ mode
      *
      R/W
      MAX ]

DISPLAY
DLBL
DOS
DOSLIB
DOSLNCNT
DOSPART
EXECTRAC
FILEDEF
FULLREAD
FULLSCREEN
HIDE [ wname
      *
      - ]

IMESCAPE
IMPCP
IMPEX
INPUT
INSTSEG
KEY
LABELDEF
LANGLIST
LANGUAGE [ALL]
LDRTBLS
```

[(options...)]

QUERY (continued)

CMS

```

LIBRARY
LINEND
LOADLIB
LOCATION wname
LOGFILE vname
MACLIB
NONDISP
OPTION
OUTPUT
PROTECT
RDYMSG
REDTYPE
RELPAGE
REMOTE
RESERVED wname
ROUTE [msgclass
      [*
      _]]
SEARCH
SHOW [wname
      [*
      _]]
SYNONYM [SYSTEM
        USER
        ALL]
SYSNAMES
TEXT
TRANslate [USER [TRANslate APPLID [applid]]]
          [SYSTEM [SYNonym [*_]]]
          [ALL [BOTH]]]
TXTLIB
UPSI
VSCREEN [vname [ALL]]
        [*
        _]
WINDOW [wname [ALL]]
        [=
        *
        _]
WMPF [nn]
      [*
      _]
options [STACK [FIFO]
        [LIFO]
        FIFO
        LIFO]

```

[(options...)]

QUERY

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]
        CPAssist1
        JOurnal2
        LOGmsg
        Names
        PAGIng
        PRIORity userid
        PROCessr
        QDROP
        SASsist1
        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 is used to determine 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
              FREe
              OFFline
              ALL
    GRaf
    LINES
    UR
    TApes
    ALL
  }
  DAsd valid
  DUMP
  Lnnn
  LOGmsg
  MITime
  Names
  PROCessr
  raddr1 [-raddr2]
  lprt
  STATUS raddr
  STORAge
  SYStem raddr
  TDsk
  Users [userid]
}
  
```

QUERY

CP Class C

Displays log message, number of logged-on users, the status of CPTRAP, CPLEVEL, specific userids, and the on-line processors in the system; lists logged-on users.

```

Query {
  CPTrap {
    STATUS
    SELECT {
      ALL
      typenum1 [...typenumn]
    }
  }
  LOGmsg
  Names
  PROCessr
  Users [userid]
}
  
```

QUERY

QUERY

CP Class D

Provides data on spooling operations.

Query

```
Files [Class c][FORM form][DEST dest][userid]2 [ * ]1 [ HOLD
NOHold
SYShold
USERhold ]

Hold

Printer [Class c][FORM form][DEST dest] userid2 [ HOLD
PUNCH * [ NOHold
Reader [ SYShold ] ] [ null ]1
[ USERhold ] ] [ ALL
[ PSF ] ]

spoolid

CPLEVEL
LOGmsg
Names

UR [ ACTIVE
ATTach
FREEe [PATHS]
OFFline
ALL ]

Users [userid]
```

¹ The options may be specified in any order.

² Using a 1-4 digit all numeric userid will cause unpredictable results for the QUERY command, which also has a 1-4 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]
  CPAssist1
  JOurnal2
  LOGmsg
  Names
  PAGing
  PRIORity userid
  PROCessr
  QDROP
  SASsist1
  SRM {
    APAGes
    DSPSlice
    IB
    MAXDrum
    MAXWss
    MHFULL
    PB
    PCI
    PGMStat
    PGMTlim
  }
  Users [userid]
}

```

- ¹ The collective use of both QUERY CPASSIST and QUERY SASSIST is used to determine 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, number of logged-on users; lists logged-on users.

```

Query {
  LOGmsg
  Names
  Users } [userid]

```

QUERY

QUERY

CP Class G

Provides system status and machine configuration information.

```
Query {
  Time
  Set
  TERMinal

  Files [Class c] [FORM form] [ NOHold
                                HOld
                                USERhold
                                SYShold ]

  LOGmsg
  {
    [Virtual] {
      CHANnells
      GRAF
      CONsole
      DAsd
      TAPes
      LINES
      UR
      STORAge
      ALL
      vaddr[-vaddr]
    }
  }

  Links vaddr
  CPLANG
  CPLEVEL
  CPUid
  SECuser
  Names
  S370E

  Reader [ [Class c ] [ NOHold ] ] [ ALL ]
  Printer [ [FORM form] HOld ] [ TBL ]
  PUnch [ [DEST dest] USERhold ] [ PSF ]
        [ spoolid [SYShold] ]

  PF [nn]
  SCREen
  PROCessr
  VMsave
  SPMODE
}
```

QUERY (continued)

CP Class G

```

{
  PER      [ CUrrent
            Names
            traceset-name
            ALL ]
  Users    [userid]
  USERID
}
```

QUERY

GCS

Requests information about your GCS virtual machine.

QUERY

```

{
  DISK      [ mode
            *
            R/W
            MAX ]
  DLBL      [mult]
  LOADLIB
  FILEDEF
  SEARCH
  SYSNAMES
  ETRACE
  GROUP
  ITRACE
  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

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
        LINK [vdev]
        RESOURCE [ALL] }
```

QUERY

RSCS

Requests system information for a link, a file, or for the system in general.

```
Query { nodeid
        linkid { Active
                Def
                Files
                Queue
                Sum
        }
        File { spoolid
               *spoolid
               *
        } [Status]
        System { NETwork
                 Active
                 Dest
                 EXits
                 LEVel
                 LOcal
                 Links
                 Ports
                 Queue
                 Routes
                 REroutes
        }
        Rscs
        Vm }
```

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

Generate a return code and either display or stack a message that identifies the characteristics of the next file in your virtual reader.

```
RDR [spool-class] [ (options...[])]  
    =
```

options:

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

RDRLIST

CMS

Display information about files in your virtual reader with the ability to issue commands from list.

```
RDRLIST [ (options...[])]  
    RList
```

options:

```
[PROFile fn]  
[Append ]
```


READCARD

READCARD

CMS

Reads data from the virtual card reader, and creates CMS disk files containing the data.

```
READcard { fn ft [fm] }
          { * [ _ [fm] ] } } [(options...[])]
          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

Read onto disk a file or note that is in your virtual reader.

```
RECEIVE [spoolid [fn [ft [fm]]]] [(options...[])]
```

options:

```
[NOTEbook fn] [Log] [Olddate] [Fullprompt] [Replace]
[NOTEbook *] [NOLog] [NEwdate] [Minprompt] [NOReplace]
               [NOPrompt]
```

[Purge]

[STack]

RECONN

RSCS

Reconnects the RSCS operator console after being disconnected and resets the userid, if any, that was used on the DISCONN command. (For RSCS operator only)

REConn

REFRESH

CMS

Updates virtual screen and their associated windows.

REFresh

RELEASE

CMS

Makes a disk and its directory inaccessible to a virtual machine.

RELease {cuu} [(DET {})]
 {mode}

RELEASE

GCS

Releases a disk.

RELEASE {cuu} [(DET{})]
 {mode}

RENAME

RENAME

CMS

Changes the name of one or more CMS disk files.

```
Rename fileid1 fileid2 [(options...)]
```

options:

```
[TYPE] [NOUPDIRT]  
[NOTYPE] [UPDIRT]
```

REPEAT

CP Class D

Holds or increases the copies of an output spool file.

```
REpeat { raddr } { [nnn] }  
        { lpvt } { [ 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

REROUTE

RSCS

Modifies the original routing (i.e. the destination system and userid) of files and messages for specific systems and userids. (For RSCS operator only)

```
REROUTE {
  { Files } [FOR] { nodeid } { userid }
  { Msgs }
  { ALL }
  NOTrcvg [FOR] { userid }
  { ANY }
  { ANY }
  { [TO] { nodeid } { userid }
  { = } { 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 the maximize or minimize.

```
REStore WInDow {wname}
  { = }
```

REWIND

REWIND

CP Class G

Rewinds a real tape drive.

```
REWind vaddr
```

REXX

CMS

The Restructured Extended Executor language (REXX) is a command programming language that allows you to combine useful sequences of commands to create new commands. The System Product Interpreter processes programs written in REXX. REXX is especially suitable for writing EXECs or editor macros, but is also a useful tool for algorithm development.

REXX 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 just a short form of the instruction PARSE UPPER ARG [template];.

```
CALL name [expression][,[expression]]... ;
```

Invokes an internal routine, an external routine or program, or a built-in function. The invoked routine may optionally return a result upon its completion.

REXX (continued)

CMS

```
DO [ name=expri [TO exprt] [BY exprb] [FOR exprf] ] [ WHILE exprw ]
  [ ;
    FOREVER
    exprr ] [ UNTIL expru ]
```

```
[ 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 terminated immediately.

REXX

REXX (continued)

CMS

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 plain DO).

LEAVE [name];
 Causes immediate exit from one or more repetitive DO loops (that is, any DO construct other than that with a plain 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 } }
 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]
 Specifies whether double byte character set (DBCS) strings can be manipulated.

 OPTIONS ETMODE DBCS strings can be manipulated
 OPTIONS NOETMODE DBCS strings cannot be manipulated

REXX (continued)

CMS

```

PARSE [UPPER] { ARG
                EXTERNAL
                NUMERIC
                PULL
                SOURCE
                VALUE [expression] WITH
                VAR name } [template];

```

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) to protect all the existing variables by
  making them unknown to following instructions.

```

```

PULL [template];
  Reads a string from the program stack (system-provided data 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 most recently created
  buffer of the program stack (system-provided data 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 displayed (or typed, etc.) on the user's console.
  The result of the expression may be of any length.

```


REXX

REXX (continued)

CMS

```
SELECT;  
  WHEN expression [;] THEN [;] instruction  
  [ WHEN expression [;] THEN [;] instruction  
    "      "      "      "  
    "      "      "      "  
    "      "      "      "
```

```
  [ OTHERWISE [;] [instruction]  
    "      "  
    "      "  
    "      " ] ]
```

END;
Conditionally executes one of several alternative instructions.

```
SIGNAL { labelname } ;  
      { [VALUE] expression  
        ON { ERROR  
          HALT  
          NOVALUE }  
        OFF { SYNTAX } }
```

Causes an **abnormal** change in the flow of control, or (if ON or OFF is specified) controls the trapping of exceptions.

REXX (continued)

CMS

```
TRACE  [ ? [ [?...] ] [ [!...] ] ] [ All ]
        [ [?...] ] [ [!...] ] ] [ Commands ]
        [ [number] ] [ Scan ]
```

Or, alternatively:

```
TRACE [ string ] [ [VALUE] expression ] [ symbol ] ;
```

Primarily used for debugging. It controls the tracing action taken (that is, how much will be displayed to the user) during execution of a REXX program.

UPPER variable [variable] [variable]...;

Used to translate the contents of one or more variables to uppercase. The variables are translated in sequence from left to right.

REXX BUILT-IN FUNCTIONS

REXX has many built-in functions and also various functions that are supplied externally.

The formats of the **built-in functions**, followed by their descriptions, are:

ABBREV(*information*, *info*[, *length*])

Tests whether *info* is a true abbreviation of *information*, with minimum *length*.

ABS(*number*)

Returns the absolute value of *number*.

ADDRESS()

Returns the current environment for commands.

ARG([*n*[, *option*]])

Returns the number of arguments, the *n*th argument, or test if the *n*th argument exists or is omitted.

BITAND(*string1*[, [*string2*][, *pad*]])

Returns a string composed of the two input strings logically AND'ed together, bit by bit.

BITOR(*string1*[, [*string2*][, *pad*]])

Returns a string composed of the two input strings logically OR'ed together bit by bit.

BITXOR(*string1*[, [*string2*][, *pad*]])

Returns a string composed of the two input strings logically exclusive OR'ed together, bit by bit.

CENTER(*string*, *length*[, *pad*])

CENTRE(*string*, *length*[, *pad*])

Returns a string of length *length* with *string* centered in it, with *pad* characters added as necessary to make up *length*.

COMPARE(*string1*, *string2*[, *pad*])

Returns 0 if the strings are identical. If they are not, the returned number is non-zero and is the position of the first character that does not match.

COPIES(*string*, *n*)

Returns *n* concatenated copies of *string*.

REXX (continued)

CMS

C2D(string[,n])

Character to Decimal. Returns the decimal value of the binary representation of *string*.

C2X(string)

Character to Hexadecimal. Converts a character string to its hexadecimal representation (unpicks).

DATATYPE(string[,type])

If only *string* is specified, the returned result is **NUM** if *string* is a valid REXX number (any format), or **CHAR** otherwise. If *type* is specified, returns 1, *string* matches the type, otherwise 0.

DATE([option])

Returns the local date in the format: dd Mmm 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*.

DELWORD(string,n[,length])

Deletes the substring of *string* that starts at the *n*th word, and is of length *length* blank-delimited words.

D2C(whole-number[,n])

Decimal to Character. Returns a character string of length as needed, or of length *n*, which is the binary representation of the decimal number.

D2X(whole-number[,n])

Decimal to Hexadecimal. Returns a string of hexadecimal characters of length as needed or of length *n*, which is the hexadecimal (unpacked) representation of the decimal number.

ERRORTXT(n)

Returns the error message associated with error number *n*.

EXTERNALS()

Returns the number of lines in the terminal input buffer (system external event queue).

FIND(string,phrase)

Searches *string* for the first occurrence of the sequence of blank-delimited words *phrase*, and returns the word number of the first word of *phrase* in *string*.

FORMAT(number[, [before] [, [after]]])

Rounds and formats number to specified integer (before) and (after) decimal places.

REXX

REXX (continued)

CMS

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 to length *length*, into the string *target* after the *n*th character.

JUSTIFY(*string*, *length*[, *pad*])

Formats blank-delimited words in *string*, by adding *pad* characters between words to justify to both margins.

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* containing the left-most *length* characters of *string*.

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 or truncated to length *length*, onto the string *target* starting at the *n*th character.

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 non-negative whole number in the range of 0-999 or *min* to *max* inclusive. The generator *seed* may be specified.

REVERSE(*string*)

Returns *string*, swapped end for end.

REXX (continued)

CMS

- RIGHT**(string, length[, pad])
Returns a string of length *length* right justified.
- SIGN**(number)
Number is rounded and returns the sign of number (-1, 0, or 1).
- 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 it is 0, all blanks are removed.
- STRIP**(string[, [option][, char]])
Removes Leading, Trailing, or Both leading and trailing characters from *string* when the first character of *option* is L, T, or B respectively. The default is B.
- SUBSTR**(string, n[, [length][, pad]])
Returns the substring of *string* that begins at the *n*th character, and is of length *length*.
- 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, and 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.
- 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*.

REXX

REXX (continued)

CMS

VERIFY(*string*,*reference*[, ['Match'] [, *start*]])

Verifies that the *string* is composed only of characters from *reference*, by returning the position of the first character in *string* that is not also in *reference*. If all the characters were found in *reference*, 0 is returned.

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*.

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(*hex-string*)

Converts *hex-string* (a string of hexadecimal characters) to Character (packs).

X2D(*hex-string* [, *n*])

Converts *hex-string* (a string of hexadecimal characters) to decimal.

REXX (continued)**CMS****RXSYSFN PACKAGE OF CP/CMS FUNCTIONS**

These all provide useful CP or CMS functions. The package is loaded automatically when needed. The formats are followed by their descriptions.

CMSFLAG(flag)

Returns the setting of one of the specified CMS flags:

ABBREV	AUTOREAD	CMSTYPE	DOS	EXECTRAC
IMPCP	IMPEX	PROTECT	RELPAGE	SUBSET

DIAG(n[?] [,data][,data]...)

Communicates with CP via a dummy DIAGNOSE instruction and returns data as a character string.

DIAGRC(n[?] [,data][,data]...)

Is identical to the DIAG function, except that CP return code and condition code are prefixed to the result.

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...

RO**CMS Immediate Command**

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 NOALARM]	[NOTify NONotify]
--------------------	----------------------

ROUTE

ROUTE

RSCS

Temporarily adds, deletes, or alters an RSCS routing table entry. (For RSCS operator only)

```
ROUTE nodeid {OFF|TO linkid}
```

RSERV

CMS

In CMS/DOS, copies, displays, prints, or punches a DOS/VS relocatable module from a private or system library.

```
RSERV modname [ft|TEXT] [(options...{})]
```

options:

```
[DISK] [PUNCH] [PRINT] [TERM]
```

RT

CMS Immediate Command

Resumes terminal displaying.

```
RT
```

RUN

CMS

Initiates a series of functions for a file depending on the file type. Selects or combines the procedures to compile, load, or start execution of the specified 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]
 [1]

S

CMS Border Command

Changes the size of the window.

s

SAVENCP

CMS

Reads/Loads 3705 control program load module.

Note: CP command privilege class A, B, or C is required to use SAVENCP.

SAVENCP fn [(options...{})]

options:

[ENTRY symbol] [NAME ncpname] [LIBE libname] [CAMOD {0
CXFINIT [fn [fn {1}}]

SAVESYS

CP Class E

Creates a copy of virtual machine storage, registers, and PSW.

SAVESYS systemname

SCREEN

SCREEN

CP Class G

Alters or changes any color and/or 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 } } } }
{ { DEfault } { NOne } } }

Note: 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
		=	*
			<u>1</u>
	Bottom	[wname	
		=	
	Down	[wname	[n
		=	*
			<u>1</u>
	Forward	[wname	[n
	=	*	
		<u>1</u>	
Left	[wname	[n	
	=	<u>1</u>	
Next	[wname	[n	
	=	*	
		<u>1</u>	
Right	[wname	[n	
	=	<u>1</u>	
Top	[wname		
	=		
Up	[wname	[n	
	=	*	
		<u>1</u>	

SEND

CP Class G

Passes commands and message replies to the designated disconnected virtual machines for execution.

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

SENDFILE

SENDFILE

CMS

Send files or notes to one or more computer users, attached locally or remotely, by issuing the command or by using a menu (display terminal only).

```
SENDFile [fn ft[fm][[TO] name ... ][options...{ }]]
```

options:

```
[Ack] [Filelist] [Log] [NEw] [Type] [NOTE]  
[NOAck] [NOFilelist] [NOLog] [Old] [NOType]
```

SENTRIES

CMS

Determines the number of lines currently in the program stack.

```
SENTRIES
```

SET

CMS

Controls various functions within your virtual machine. (Only one function may be specified per SET command.)

The options available with SET are listed below. A complete description of each option follows.

ABBREV	DOSPART	LDRTBLS	RELPAGE
APL	EXECTRAC	LINEND	REMOTE
AUTOREAD	FULLREAD	LOCATION	RESERVED
BLIP	FULLSCREEN	LOGFILE	SYSNAME
BORDER	IMESCAPE	NONDISP	TEXT
CHARMODE	IMPCP	NONSHARE	TRANSLATE
CMSPF	IMPEX	OUTPUT	UPSI
CMSTYPE	INPUT	PROTECT	VSCREEN
DOS	INSTSEG	RDYMSG	WINDOW
DOSLNCNT	LANGUAGE	REDTYPE	WMPF

SET ABBREV

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 console read is to be issued immediately after command execution or pressing the ENTER key is its equivalent.

```
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

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
 NOHigh]

[color]
[exthi]
[PSset]

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

SET CMSTYPE

CMS

Specifies suppression of a CMS terminal display within an EXEC.

```
SET CMSTYPE {HT}
             {RT}
```

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 [mode[(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 FULLREAD

SET FULLREAD

CMS

Allows 3270 null characters to 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 }  
               {OFF }  
               {SUSPEND }  
               {RESUME }
```

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 that are unrecognized by CMS are considered CP commands and are passed on to CP.

```
SET IMPCP {ON }  
          {OFF }
```

SET IMPEX

CMS

Controls whether EXEC files are treated as commands.

```
SET IMPEX { ON  
          { OFF }
```

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.

```
SET INPUT [ a  xx  
          { xx  yy }
```

SET INSTSEG

CMS

Specifies whether the system should search the Installation Discontiguous Shared Segment (DCSS) to locate an EXEC or Editor Macro.

```
SET INSTSEG { ON [mode|LAST]  
            { 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

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/or 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 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} [fn [LOGFILE [*
                        -
                        A1]]]
```

SET NONDISP

CMS

Defines a character to be used in place of nondisplayable characters.

```
SET NONDISP [char]
```

SET NONSHARE

SET NONSHARE

CMS

Specifies a non-shared copy of a normally 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 { LMSG  
           { SMSG
```

SET REDTYPE

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 {ON }  
            {OFF }
```

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 {ON }  
            {OFF }
```

SET REMOTE

CMS

Controls the display of data transmissions.

```
SET REMOTE {ON }  
           {OFF }
```

SET RESERVED

CMS

Specifies whether the number of lines in a window are to be used to display virtual screen reserved lines.

```
SET RESERVED wname {rtop } {rbot }  
                  { * } { * }
```

SET SYSNAME

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 } entryname
             { CMSVSAM }
             { CMSAMS  }
             { CMSBAM  }
```

SET TEXT

CMS

Activates character code conversion of TEXT characters for windows.

```
SET TEXT { ON }
         { OFF }
```

SET TRANSLATE

CMS

Suppresses translations and translation synonyms of command names for a language.

```
SET TRANSLATE { ON } [ USER [ TRANslate [ APPLID applid ] ] ]
               { OFF } [ SYStem [ 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

SET VSCREEN

CMS

Indicates what action should take place when the virtual screen is updated with data.

```
SET VSCREEN    vname { [TYPE] [Protect] [High]
                    { [NOTYPE] [NOPProtect] [NOHigh]
                    { [color] [exthi] [psset] }
```

SET WINDOW

CMS

Specifies whether the window is to be variable or fixed size.

```
SET WINDOW    wname { [VARIABLE] [POP] [TOP]
                    { [FIXed] [NOPop] [NOTop] }
```

SET WMPF

CMS

Defines a WMPF key to execute a windowing command.

```
SET WMPF     nn { [pseudonym] [keyword] string
                { [NOWRITE] [DELAYED] }
```

SET

CP Class A

Sets special CP preferred options.

```

SET {
  AFFinity [userid] {ON|OFF|nn}
  CPAssist {ON|OFF} [PROC[nn]]

  FAVORed userid [nnn]
  OFF

  JOurnall {LOgon} {ON}
  {LInk} {OFF}

  PRIORity userid nn
  QDROP userid {ON|OFF}[USERS][NOQ3]
  REServe userid {nnn|OFF}
  SASsist {ON|OFF} [PROC[nn]]
  S370E {ON|OFF} [[PROC]addr]
}

```

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} {OFF} [class {mm:ss}...]}
  {OFF}
}

```


SET

SET

CP Class E

Sets SRM function and the number to be 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, or enables/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

The SET command controls various functions within your virtual machine.

```

Set {
  ACNT
  AUTOPoll
  RUN      {ON }
  LINEdit  {OFF}
  NOTRans
  ECmode
  CONCeal
  ISAM
  PAGEX
  EMSG {ON|OFF|CODE|TEXT|IUCV}
  TIMER {ON|OFF|REAL}
  CPUid bbbbbb

  IMSG {ON }
  MSG  {OFF}
  SMsg {IUCV}
  WNG

  CPCONIO {OFF }
  VMCONIO {IUCV}
  ASSist  {[ON] [SVC|NOSVC] [TMR|NOTMR]}
          {OFF}

  AFFinity {ON|OFF}

  PFnn [IMMed ] [pfdatal#pfdata2#...pfdatan]
       [DElAyed]

  PFnn [TAB n1 n2...]

  PFnn [COPY (resid) ]
       [COPY (cuu) ]
       [COPY (luname) ]
       [COPY (Laddr) ]

  PFnn RETrieve

  VMSAVE1 {ON|OFF|name}

  STBypass { [nnnnK [NOVERIFY]] ]
           [nnM ]
           [VR ]
           [OFF] }
}

```

SET

SET (continued)

CP Class G

```

| { STMult { [ n [[ [USEG xx] CSEG yyy] ] ] } }
| { 370E [ON|OFF]
|   MIH { ON
|         OFF }
| { SVCAcc [ON|OFF]
| }

```

¹ When specifying this operand, virtual machine size cannot exceed eight megabytes.

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

Sets the storage key for a discontinuous saved segment.

```
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[({mmm[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

SHUTDOWN

RSCS

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 it is 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 which 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 that are issued 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 rcsid" characters in front of the RSCS command expression).

```
SMsg rcsid { command-text  
           { Msg nodeidm userid message-text  
           { CMD nodeidc CMD-command-text  
           }
```

SNTMAP

CMS

Processes DMKSNT macro definitions and produces two CMS files; a saved segment DASD map and a virtual memory map.

```
SNTMAP (fn (ft (fm)))
```

SO

CMS Immediate Command

Suspends the recording of trace information during the execution command or program.

```
SO
```

SORT

CMS

Sorts records within a file and creates a new file containing the sorted records.

```
SORT fileid1 fileid2
```

SPACE

SPACE

CP Class D

Forces single spacing on the printer.

```
SPAcE{raddr }
      {lprt }
```

SPGEN

CMS

Does various system generation and maintenance functions, using the parameters contained in SPGEN PROFILE. These functions include:

- Creating, verifying, and displaying system profile parameters.
- Assembling system files.
- Generating CP, CMS, and GCS nuclei.
- Receiving and verifying load maps.

```
SPGEN { CREATE                [(optionA{})]
      { VERIFY                [(optionA{})]
      { DISPLAY                [(optionA{})]
      {   [ ALL                [(optionA{})]
      {     [ compid          ]
      { ASSEMBLE fn [ compid ] [(optionA{})]
      {   [ CP                ]
      { SETUP                 compid [(optionA{})]
      { NUCLEUS               compid [(optionA optionB optionC{})]
      { MAP                   compid [(optionA{})]
      { DTYPE                 vdev }
```

```
optionA:
        [PROFILE profname]
```

```
optionB:
        [NOIPL]
```

```
optionC:
        [NOSETUP]
```

SPLOAD

CMS

Loads contents of the VM/SP product tapes to appropriate minidisks during initial VM/SP installation.

SPLOAD group element [fn [ft]]
 [* [*]]

SPMODE

CP Class A

Establishes or resets the single processor mode.

SPMode { ON }
 { OFF }

SPOOL

SPOOL

CP Class G

Changes spooling control options.

```

SPool {Reader} {
  {Class { * } [CONT] [EOF] [HOLD] }1
  {vaddr} { [NOCont] [NOEOF] [NOHold] }
}

{Printer} {
  {PUnch} {
    {CONsole} {
      {vaddr} {
        { [To] {userid} }1 [Hold] [CONT]
        { [For] { * } } [NOHold] [NOCont]
          {SYSTEM}
        [OFF] [Class c] [COpy[*]nnn]1
        {CLOSE} [Flash name nnn]3 [FORM {form}] [DEST {dest}]
        {PURGE} [MODify name[n] ] [ {OFF} ]
          Chars name1
          [name2]
          [name3]
          [name4]]]
          CHars name1
          [CHars name2
          [CHars name3
          [CHars
            name4]]]
          FCB name
        {Dist {distcode} } ] [STArt]2 [TErm]2
        {OFF} ] [STop] ] [NOTerm]
      }
    }
  }
}

```

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 only apply 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 option2
  LOAD raddr SADump option2
  LOAD raddr {
    Printer
    PUnch
    Reader
  } {
    spoolid1 [spoolid2]
    [END]
  } {
    option2
    option3
  }
  DUMP raddr {
    Printer
    PUnch
    Reader
  } {
    spoolid1 spoolid2
    [END]
  } {
    Class c1 [c2[c3[c4]]][FORM form][DEST dest]
    FORM form [DEST dest]
    DEST dest
    ALL
  } {
    option1
    option2
    option3
    option4
  }
}

```

options:

```

option1 [
  MODE [
    800
    1600
    6250
    38K
  ]
]
option2 [
  LEAVE
  REWIND
  RUN
]
option3 [
  SYSHOLD
  USERHOLD
  NOHOLD
]
option4 [PURGE]

```

SSERV

SSERV

CMS

In CMS/DOS, copies, displays, prints, or punches a book from a DOS/VSE source statement library.

```
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 program execution.

```
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 ]
                [Queue      {Priority|Fifo|Size} ]
                [TRace      {ALL|LOG} ]
                TYPE      {
                           NJE
                           SNANJE
                           RJE
                           MRJE
                           3270P
                           SNA3270P
                           }
                           [OParm operation parameters... ]
                           [Parm [operation parameters... ] ]

```

Note: Any combination of keywords with associated options may be entered in any order, except that Parm keyword must be the rightmost keyword.

STAT

STAT

IPCS

Lists current status of a problem, a specific subset of problems, or all problems.

STAT	nnnnn	}	1	}	2
	ALL				
	{OPENUSER}				
	{OPNUSR}				
	{OPENIBM}				
	{OPNIIBM}				
	{OPEN}				
	{OPN}				
	APARED				
	{NEEDINFO}				
	{NDINFO}				
	{PTFRCDV}				
	{PTFRCV}				
	PTFON				
CLOSED					
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 file on any accessed disk. STATEW verifies the existence of a file on a read/write disk.

```
STATE {fn}{ft}{fm}
STATEW {*}{*}{*}
```

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 { { hexloc
        Mhexloc
        Nhexloc } hexword1 [hexword2...]
      { Lhexloc
        MLhexloc
        NLhexloc }
      { Shexloc
        MShexloc
        NShexloc } hexdata
    }
  
```

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 issue this command.

```
STOP TSAF
```

STORE

STORE

CP Class G

Alters virtual machine storage, PSW, and registers.

```
STore {
    hexloc hexword1[hexword2...]
    Lhexloc

    Shexloc hexdata

    {Greg} hexword1[hexword2...]
    {Xreg}

    {Yreg} hexdword1[hexdword2...]

    Psw [hexword1] hexword2

    STATUS
}
```

SVCTRACE

CMS

Records information about supervisor call instructions.

```
SVCTrace { ON }
          { OFF }
```

SYNONYM

CMS

Specifies alternate names for invoking CMS commands.

```
SYNONym [ fn [ SYNONYM [ fm ] ] ] ] [(options...{ })]
```

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 DEv { Printer } [tagtext]  
      { PUnch  
        CONsole  
        vaddr }
```

```
File spoolid [tagtext]
```

```
QUery { { Printer  
         { PUnch  
           CONSOLE  
           vaddr  
         }  
       }  
       File spoolid }
```


TAPE

TAPE

CMS

Performs tape to disk or disk to tape operations for CMS files.

```
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())]
```

Note: The tapcmd operand can be one of the following:

[BSF|BSR|ERG|FSF|FSR|REW|RUN|WTM]

optionA:

[WTM] [BLKSIZE 4096]
[NOWTM] [BLKSIZE 800]

optionB:

[NOPrint]
[Print]
[DISK]
[Term]

optionC:

[EOF n]
[EOT]
[EOF 1]

optionD:

[TAPn] [cuu] [TRTCH a] [7TRACK] [DEN den]
[TAP1] [181] [9TRACK]
[18TRACK]

TAPE (continued)

CMS

optionE:

[LEAVE
REWIND]

optionF:

[TRANsfer BUFFered
TRANsfer IMMEDIATE]

TAPEMAC

CMS

Creates a CMS MACLIB from an unloaded partitioned data set (PDS) from a tape created by the IEHMOVE utility program under OS.

TAPEMAC fn [SL [labeldefid]
NSL filename [ID=identifier]] [(options...{})]

options:

[TAPn] [ITEMCT yyyyyy]
[TAP1] [ITEMCT 50000]

TAPPDS

CMS

Creates CMS disk files from tapes which are input to or output from the IEBTPCH, IEBUPDTE, and IEHMOVE OS utility programs.

TAPPDS [fn [ft [fm]
* * *]
* * [A1]] [SL [labeldefid]
NSL filename [ID=identifier]] [(options...{})]

options:

[PDS
NOPDS
UPDATE] [COL1
NOCOL1] [TAPn
TAP1] [END
NOEND] [MAXTEN
NOMAXTEN]

TE

TE

CMS Immediate Command

Trace end command stops all tracing of your System Product interpreter or EXEC 2 program or macro.

TE

TELL

CMS

Send a message to one or more computer users who are logged on to your computer or to one attached to yours via RSCS.

TELL name message

TERMINAL

CP Class G

Controls virtual console functions.

TERMinal	<pre> CHardel LINEDel {ON OFF} LINEnd {OFF char} EScape TABchar 2 </pre>	1
	<pre> APL TEXT {ON OFF} ATtn Hilight SCRNsave MODE {CP VM} </pre>	3
	<pre> LINESize {nnn OFF} CONmode {3215 3270} BREakin {IMmed GUESTctl} BRKkey {PA1 PF1 ... PF24 NONE} TYpe {3101 TTY} PROMpt {VM TTY} SCROLL {nnn CONT} ACSiitbl {VM2 VM1} CNTL {USR SYS} </pre>	

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

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

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

³ The SCRNSAVE operand is not provided for VM/VTAM and remote terminals.

TRACE

TRACE

CP Class G

Traces and records program execution.

```
Trace { [ SVC  
        I/O  
        PROgram  
        EXTERNAL  
        PRIV  
        SIO  
        CCW  
        BRanch  
        INSTRUCT  
        ALL  
        CSW  
        END ] } 1 [ PRINTER  
                  [ TERminal ] [ NORun ]  
                  [ BOTH ] [ RUN ]  
                  OFF ]
```

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

```
TRACE SVC PROGRAM SIO PRINTER.
```

TRACE

RSCS

Monitors line activity on a specified link.

```
Trace [linkid] [ ALL  
                LOG  
                NOLog  
                OFF ] [TO userid [nodeid ]
```

TRANSFER

CP Class D

Transfers closed reader spool files.

```
TRANSFER [userid] [Printer  
                SYSTEM  
                * ] [Punch  
                Reader ] [ spoolid  
                Class c  
                FORM form  
                DEST dest  
                ALL ] [ [TO] { *  
                          {userid} } ] [Printer  
                PUNCH  
                Reader ]  
                [ FROM { ALL  
                        {userid} } ]
```

TRANSFER

CP Class G

Transfers closed reader spool files.

```
TRANSFER [Printer] { spoolid } { [TO] { userid } } [Printer]
          [PUnch]   { Class c } { From { * } } [PUnch]
          [Reader]  { FORM form } {      { ALL } } [Reader]
                   { DEST dest }
                   { ALL }
```

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]
```

TRAPRED

TRAPRED

CMS

Accesses the CPTRAP reader file and reviews the entries contained in that file. Once TRAPRED is invoked, you may execute TRAPRED subcommands.

TRAPRED filename

TRAPRED subcommands:

typenum	[Vmblok	address]
		DEVaddr	cuu	
		COde	code-value	
		MACHtype	{ nn	}
			{ machname }	
		OFF		

ALL [ON]
 [OFF]

Hex
FOrmat
TOP
BOTtom

Up [1] Down [1]
 [n] [n]

Type [1] TYPEBack [1]
 [n] [n]

Printer [1]
 [n]

QUIT

Note: The TRAPRED subcommands must be entered on separate command lines.

TS

CMS Immediate Command

Trace start command starts tracing your System Product Interpreter or EXEC 2 program or macro.

TS

TXTLIB

CMS

Updates a library of TEXT files (object modules).

```
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

Types or displays all or part of a file at a terminal.

```
Type fn ft [fm] [recl[recn]] [(options...{})]
          [*] [* [*]]
          [ ] [ ]
```

options:

```
[COL {xxxxx} -[yyyyy]] [HEX] [MEMBER{*}
 [ ] [ ] [MEM {name}]
```

UNLOCK

CP Class A

Releases pages of storage.

```
UNLOCK { {userid} firstpage lastpage
         {SYSTEM}
         {VIRT=REAL}
         {V=R} }
```


UPDATE

UPDATE

CMS

Makes changes in a program source file as defined by control statements in a record file.

```
Update {fn1 ft1      {fn1 [fn2 [ft2 [fm2]]]} } [(options...{})]  
      {ASSEMBLE    {A1}}
```

options:

```
{REP   } {SEQ8 } {INC } {CTL } {STK } {TERM } {PRINT} {STOR  
{NOREP} {NOSEQ8} {NOINC} {NOCTL} {NOSTK} {NOTERM} {DISK} {NOSTOR}
```

[OUTMODE fm]

Control Statements:

```
./ S [seqstrt [seqincr [label]]]  
./ I seqno [$ [seqstrt [seqincr]]]  
./ D seqno1 [seqno2] [$]  
./ R seqno1 [seqno2] [$ [seqstrt [seqincr]]]  
./ * [comment]
```

UTILITY

CMS

Allows you to do some occasionally used utility functions (i.e. print off system definition files, create a stand-alone service utility tape, and write back-up copy of the CP nucleus to tape.

```
UTILITY { PRSAMPLE  
        { UTILITAPE { ALL  
                    { DSF  
                    { FMT  
                    { DDR  
                    { DIR  
        { IPLDECK { ALL  
                  { DIR [ctlfile]  
                  { DDR  
                  { FMT  
        { NUCTAPE
```

VALIDATE

CMS

Verifies the syntax of a file identifier (filename filetype filemode). If the filemode is specified, VALIDATE verifies whether or not the disk is accessed.

```
VALIDATE {fn} {ft} {fm}
          { * } { * } { _ }
```

VARY

CP Class B

Varies the availability of a device.

```
VARY {ONLine} {raddr...}
     {OFFline} {raddr-raddr}
              {lprrt}
              {PROCCessr}

Offline PROCCessr nn {VPHY} [FORCE]
                    {VLOG}
```

VMDUMP

CP Glass G

Dumps storage for virtual machine. It enables sending dumps to other users. Used in conjunction with VM/IPCS.

```
VMDUMP [hexloc1] [ { - } [hexloc2] ] [ { TO * } [FORMAT vmtypel]
        [ Q ] [ : ] [ END ] [ TO userid ]
               [ SYSTEM ]
               [ { . } [bytecount] ] [DSS] [*dumpid]
               [ END ] ]
```

VMFASM

VMFASM

CMS

Updates a specified source file according to entries in a control file and assembles the updated source file.

```
VMFASM fn ctlfile [(options...)]
```

options:

```
[DISK] [TERM] [LIST] [DECK] [RENT] EXP XREF [MAX]
[PRINT] [NOTERM] [NOLIST] [NODECK] [NORENT] [MIN]
[STD]
```

VMFDOS

CMS

Creates CMS files containing VSE modules for specific installation purposes only. VMFDOS uses either a VSE distribution library tape or VSE SYSIN tape to install only DOS/VS RPG II and VSE/VSAM Program Products.

```
VMFDOS [LOAD] [SCAN] [181] [182] [PRIVate] [fn] [(options...)]
[SYSTEM]
[SYSIN]
[TAP1]
[TAP2]
```

PRIV or SYST options:

```
CSL { *
      xxx*
      module name }
```

```
RL { _
     yyy*
     module name }
```

```
SL { *
     zzz*
     module name }
```

SYSIN options:

```
{ ALL }
{ SELECT }
```

VMFLKED

CMS

Invokes the CMS LKED command to link-edit modules into a loadlib.

```
VMFLKED  fn [ft [fm]]  [(options [])]
```

options:

PRINT

MODULE module-name

VMFLOAD

CMS

Generates a new CP nucleus, or stand-alone dump (DDR) program. The VMFLOAD program uses two files, a loadlist EXEC file and a control file, to produce a punch file that has several object modules.

```
VMFLOAD  loadlist  ctlfile  [langid]
```

VMFMAC

CMS

Updates Macro libraries. If you specify a control file, the EXEC invokes the CMS UPDATE command to update specified copy or macro files, according to entries in a control file, and then builds a new macro library from the resulting new versions of those files.

```
VMFMAC  libname  [ctlfile]
```

VMFMERGE

CMS

Applies PTFs from the DELTA disk to the Merge disk.

```
VMFMERGE  prodid  {PTF {ptfnum | *} }  [EXCLUDE exclist]
              {PTFLIST applist }
```

VMFNLS

VMFNLS

CMS

Automatically applies updates to national language-related source files, generates text files, and renames the text files so they can be loaded into the system.

```
VMFNLS fn ft ctlfile [(options...)]
```

VMFPLC2

CMS

Loads source code from the Product Tape, loads the service installation VMSEV EXEC from the Program Update Tape, dump CMS-formatted files from disk to tape, and loads previously dumped files from tape to disk. The VMFPLC2 command does not process multi-volume files. Files processed by the VMFPLC2 command must be CMS-formatted.

```
VMFPLC2 { DUMP {fn} {ft} {fm} [(optionA optionB optionD)]
        LOAD [{fn} {ft} {fm}] [(optionB optionC optionD optionE optionF)]
        SCAN [{fn} {ft}] [(optionB optionC optionD optionF)]
        SKIP [{fn} {ft}] [(optionB optionC optionD)]
        MODESET [(optionD)]
        tapcmd [n] [(optionD)] }
```

optionA:

```
[ WTM
  NOWTM ]
```

optionB:

```
[ NOPrint
  Print
  Term
  DISK
  APPend ]
```

VMFPLC2 (continued)

CMS

optionC:

```
[EOT
 EOF n
 EOF 1]
```

optionD:

```
[TAPn
 TAP1]
[vdev
 181]
```

```
[ 7TRACK
 9TRACK
 18TRACK]
```

[DEN den]

[TRTCH a]

optionE:

[SElect] [STOP]

optionF:

[DATE]

VMFREMOV

CMS

Removes PTFs applied by the VMFMERGE EXEC procedure.

```
VMFREMOV  prodid  [PTF {ptfnum | *}
                   PTFLIST remlist
                   CONVERT [lastfilemode]]
```

VMFTXT

CMS

Creates text libraries. VMFTXT rebuilds a named TXTLIB file using a member list in an EXEC file with the same name.

```
VMFTXT  libname  [ctlfile]
```

VMFZAP

CMS

Applies ZAPs and maintains a record if them in the ZAP Log. This exec uses the Base disk, Merge disk and the ZAP disk as inputs and produces an updated ZAP disk as output.

```
VMFZAP  prodid
```

WAITREAD VSCREEN

WAITREAD VSCREEN

CMS

Updates the virtual screen with data, rebuilds the screen image (updates the 3270 data buffers), and waits for the next attention interrupt.

WAITREAD VScreen vname

WAITT VSCREEN

CMS

Updates the virtual screen with data.

WAITT VScreen [vname]
 *
 _

WARNING

CP Class A or B

Sends high priority messages.

Warning { userid } msgtext
Wng { OPERator }
 ALL

WRITE VSCREEN

WRITE VSCREEN

CMS

Enters information into 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 optionD is used, a right parentheses cannot be used to mark the end of the options.

X

CMS Border Command

Maximizes the window.

x

XEDIT

XEDIT

CMS

Creates, modifies, and manipulates CMS disk files. This is the command used to invoke the VM/SP System Product editor. Once XEDIT is invoked you may execute XEDIT subcommands and use the System Product interpreter or the EXEC 2 macro facility.

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:

```
[Width nn] [NOScreen] [PROFile macroname] [WINDow wname]
[NOCLear] [NOPROFil] [NOMsg] [MEMber membername]
```

options valid only in Update mode:

```
[Update|NOUpdate] [Seq8|NOSeq8] [Ctl fnl|NOctl]
[Merge] [UNtil filetype]
[Incr nn] [SIDcode string]
```

The formats of the **XEDIT subcommands and macros**, followed by their descriptions, are:

& [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.

XEDIT (continued)

CMS

ALter char1 char2 $\left[\begin{array}{c} \text{target} \\ \underline{1} \end{array} \left[\begin{array}{c} n \\ * \\ \underline{1} \\ G \end{array} \left[\begin{array}{c} p \\ \underline{1} \end{array} \right] \right] \right]$

Changes a single character to another character unavailable on 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).

Change /string1 $\left[\begin{array}{c} /string2/ \\ \underline{1} \end{array} \left[\begin{array}{c} \text{target} \\ * \\ \underline{1} \end{array} \left[\begin{array}{c} p \\ \underline{1} \\ q \\ \underline{1} \end{array} \right] \right] \right]$

Changes a specified group of characters on one or more lines at one time.

XEDIT

XEDIT (continued)

CMS

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|_]

Prepares one or more lines for automatic repositioning of data (see SET TABS).

COPY target1 target2

Copies one or more lines at a specified location in the file.

COunt /string [/target|_]

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.

XEDIT (continued)

CMS

CURsor CMdline [colno|1] [Priority n]
 Column [Priority n]
 File lineno [colno] [Priority n]
 Screen lineno [colno] [Priority n]
 Home [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 $\left[\begin{array}{l} n \\ \underline{1} \end{array} \left[\begin{array}{l} \text{target} \\ \underline{1} \end{array} \right] \right]$

Duplicates one or more lines beginning with the current line.

EMSG [text]

or

[mmmnns text]

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').

XEDIT

XEDIT (continued)

CMS

EXTRACT /operand [/operand [/operand]]...

Used **within a macro** to get information about internal XEDIT variables or about file data.

operand

may be any one of the keywords listed below.

ACTION	HEX	RESERVED [*]	UNTIL
ALT	IMAGE	RING	UPDATE
APL	IMPCMSCP	SCALE	VARBLANK
ARBCHAR	INPMODE	SCOPE	VERIFY
AUTOSAVE	LASTLORC	SCREEN	VERSHIFT
BASEFT	LASTMSG	SELECT	WINDOW
BRKKEY	LENGTH	SEQ8	WIDTH
CASE	LIBNAME	SERIAL	WRAP
CMDLINE	LIBTYPE	SHADOW	ZONE
COLOR field *	LINE	SIDCODE	=
COLPTR	LINEND	SIZE	
COLUMN	LRECL	SPAN	
CTLCHAR [char]	LSCREEN	SPILL	
CURLINE	MACRO	STAY	
CURSOR	MASK	STREAM	
DISPLAY	MEMBER	SYNONYM [name *]	
EFMODE	MSGLINE	SPILL	
EFNAME	MSGMODE	STAY	
EFTYPE	NBFILE	STREAM	
EOF	NONDISP	SYNONYM [name *]	
ESCAPE	NULLS	TABLINE	
ETARBCH	NUMBER	TABS	
ETMODE	PA n _*	TARGET	
FILLER	PACK	TERMINAL	
FILLER	PENDING (see below)	TEXT	
FLSCREEN	PF [n _*	TOF	
FMODE	POINT [*]	TOFEOF	
FNAME	PREFIX (see below)	TOL	
FTYPE	RANGE	TRANSLAT	
FULLREAD	RECFM	TRUNC	
	REMOTE	UNIQUEID	

PENDING [BLOCK] [OLDNAME] name|* [target1[target2]]

PREFIX [Synonym name|*]

XEDIT (continued)

CMS

FILE $\left[\begin{array}{c} \text{fn} \\ = \end{array} \left[\begin{array}{c} \text{ft} \\ = \end{array} \left[\begin{array}{c} \text{fm} \\ = \end{array} \right] \right] \right]$

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|*|1]

Scrolls (toward the end of the file) the operand-specified number of screen displays.

GET $\left[\begin{array}{c} \text{fn} \\ = \end{array} \left[\begin{array}{c} \text{ft} \\ = \end{array} \left[\begin{array}{c} \text{fm} \\ = \\ * \\ - \end{array} \right] \left[\begin{array}{c} \text{firstrec} \\ \underline{1} \end{array} \left[\begin{array}{c} \text{numrec} \\ * \\ - \end{array} \right] \right] \right] \right]$

Inserts all or part of a specified CMS file following the current line of the edit file.

Help [MENU|HELP|name]

Displays a list of all XEDIT subcommands and macros and their descriptions, formats, and parameters, or invokes the CMS HELP command.

HEXType [target|1]

Displays a specified number of lines in both hexadecimal and EBCDIC. This is a macro.

Input [line]

Inserts a single line into a file. Also used to leave edit mode for entry into input mode.

XEDIT

XEDIT (continued)

CMS

Join [ALigned] [Column
CURSOR]

or

[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...[...])]

Reads a copy of the file being edited into virtual storage. This subcommand **can only be issued from the XEDIT profile**. Allows the macro to 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.

options:

[Width nn] [NOScreen] [PROFile macroname] [WINdow wname]

[NOClear] [NOPROFil] [NOMsg] [MEMber membername]

options valid only in Update mode:

[Update|NOUpdate] [Seq8|NOSeq8] [Ctl fnl|NOctl]

[Merge] [UNtil filetype]

[Incr nn] [SIDcode string]

[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.

XEDIT (continued)

CMS

MErge target1 target2 [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	IMPcmscp	SElect
APL	LASTLorc	SERial
ARBchar	LINeNd	SHADow
AUtosave	LRecl	SIDcode
BRKkey	MACRO	SPAN
CASE	MASK	SPILL
CMDline	MSGLine	STAY
COLOR field	MSGmode	STReam
COLPtr	NONDisp	SYNonym
COLUMN	NULls	TABLine
CURLine	PAn	TABS
DISPlay	PACK	TERMinal
ENTer	PFn	TEXT
ESCApe	PREfix [Synonym name]	TOFEOF
ETARBCH	RANge	TRunc
ETMODE	RECFm	VARblank
FILLer	REMOte	VeriFy
FMode	SCALE	VERShift
FName	SCOPE	WRap
FType	SCREen	Zone
FULLread		
HEX		
IMage		

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 target1 target2

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.

XEDIT

XEDIT (continued)

CMS

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 non-blank 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.

XEDIT (continued)

CMS

PRESeRve

settings saved include:

ARBCHAR	IMPCMSCP	SHADOW
AUTOSAVE	LASTLORC	SPAN
CASE	LINEND	SPILL
CMDLINE	LRECL	STAY
COLOR	MACRO	STREAM
COLPTR	MASK	SYNONYM
CURLINE	MSGMODE	TABLINE
DISPLAY	NULLS	TABS
ESCAPE	NUMBER	TOEOF
ETARBCH	PACK	TRUNC
FILLER	PREFIX	VARBLANK
FMODE	RECFM	VERIFY
FNAME	SCALE	WRAP
FTYPE	SCOPE	ZONE
HEX	SERIAL	=
IMAGE		

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 $\left[\begin{array}{l} \text{target} \\ \underline{\text{1}} \end{array} \left[\begin{array}{l} \text{fn} \\ = \end{array} \left[\begin{array}{l} \text{ft} \\ = \end{array} \left[\begin{array}{l} \text{fm} \\ = \end{array} \right] \right] \right] \right]$

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 $\left[\begin{array}{l} \text{target} \\ \underline{\text{1}} \end{array} \left[\begin{array}{l} \text{fn} \\ = \end{array} \left[\begin{array}{l} \text{ft} \\ = \end{array} \left[\begin{array}{l} \text{fm} \\ = \end{array} \right] \right] \right] \right]$

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.

XEDIT

XEDIT (continued)

CMS

Query [option]

options (specify only one each time):

ACTION	LASTLorc	Seq8
ALT	LASTmsg	SERial
APL	LENGth	SHADow
ARBchar	LIBName	SIDcode
AUTosave	LIBType	SIZE
BASEft	Line	SPAN
BRKkey	LINENd	SPILL
CASE	LRecl	STAY
CMDline	LScreen	STReam
COLOR * field	MACRO	SYNonym [* name]
COLPtr	MASK	TABLine
COLumn	MEMBER	TABS
CTLchar[char]	MSGLine	TARGet
CURLine	MSGMode	TERMinal
CURSor	NBfile	TEXT
DISPlay	NONDISP	TOF
EFMode	NULls	TOPEOF
EFName	NUMber	TOL
EFType	PA [n *]	TRANSLat
ENTER	PACK	TRunc
EOF	PENDing (see below)	UNIQueid
EOL	PF [n *]	UNTil
ESCAPE	Point [*]	UPDate
ETARBCH	PREFix (see below)	VARblank
ETMODE	RANge	Verify
FILLer	RECFm	VERShift
FMode	REMote	Width
FName	RESERved	WINDow
FType	RING	Zone
FULLread	SCALE	=
HEX	SCOPE	
IMage	SCREen	
IMPcmscp	SELEct	

PENDing [BLOCK] [OLDNAME]name|*

PREFix [Synonym *|name]

Displays the current setting of various editing options.

XEDIT (continued)

CMS

QUIT [n]

Terminates the editing session and leaves the previous copy intact.

READ [CMDLINE] [Tag]
All [Number] [NOTAG]
Nochange [Number]

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/or 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.

XEDIT

XEDIT (continued)

CMS

REStore

Restores the settings of the XEDIT variables to the values in effect when last the PRESERVE subcommand was issued.

RGTLEFT [n]

View columns of data not currently visible on the screen.

RIght [n|_]

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 $\left[\begin{array}{l} fn \\ = \end{array} \left[\begin{array}{l} ft \\ = \end{array} \left[\begin{array}{l} fm \\ = \end{array} \right] \right] \right]$

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

Changes the settings of various editing options while editing is in progress.

options (must specify one):

SET ALT n [p]

Changes the number of alterations that have been made to the file since the last AUTOSAVE and/or since the last SAVE.

[SET] APL ON|OFF

Shows whether APL keys are available.

[SET] ARBchar ON|OFF [char]

Defines an arbitrary character used in a target definition. Note that the initial setting is OFF.

XEDIT (continued)

CMS

```
[SET] AUtosave    n    mode
                OFF   A
```

Sets or resets the automatic save function of the editor. Note that the initial setting is OFF.

```
[SET] BRKkey ON|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.

XEDIT

XEDIT (continued)

CMS

- [SET] CURLine ON M[+n|-n] | [\pm 1|-]n
Defines the nth 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|OFF [char]
Allows entry of subcommand (on typewriter terminals) when in input mode without leaving input mode.
- [SET] ETARBCH ON|OFF [char]
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 filemode of the edited file.
- [SET] FName fn
Changes the filename of the edited file.
- [SET] FType ft
Changes the filetype of the edited file.
- [SET] FULLread ON|OFF
Allows 3270 null characters to be recognized in the middle of screen lines.
- [SET] HEX ON|OFF
Allows subcommand operands and targets to 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.

XEDIT (continued)

CMS

[SET] IMPcmscp ON|OFF

Determines whether or not non-XEDIT recognized subcommands are transmitted implicitly 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|l] [Overlay]
OFF

Defines the location of the message line on the screen, and the maximum number of lines that a message may occupy.

[SET] MSGMode ON [Short|Long]
OFF

Controls message display. Note that the initial setting is ON LONG.

XEDIT

XEDIT (continued)

CMS

[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 prefix area. Note that the initial setting is OFF.

[SET] PAn

[BEFORE]	[string]
	[NULLKEY]
	[COPYKEY]
	[TABKEY]
[AFTER]	
[ONLY]	
[IGNORE]	

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|BLOCK|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]
	[NULLKEY]
	[COPYKEY]
	[TABKEY]
[AFTER]	
[ONLY]	
[IGNORE]	

Defines or removes a meaning for a specified program function (PF) key. Note that TABKEY is the initial setting of the PF4 key.

[SET] Point .symbol [OFF]

Defines or redefines the symbolic name for the current line.

XEDIT (continued)

CMS

```
[SET] PREFIX ON      [LEFT|Right]
          OFF
          Nulls [LEFT|Right]
```

or

PREFIX Synonym newname oldname
Controls display of the prefix area. Also defines a synonym for a prefix subcommand.

```
[SET] RANGE target1 target2
      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] | [+|- n][color] [exthi] [PSs] High
      [text]
                                     Off                               Nohigh
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 s11 sw1 sh1 sv1 [s12 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.
```

XEDIT

XEDIT (continued)

CMS

[SET] SElect [+|-] n [target]

Designates a "selection level" for specified lines. A selection level is a positive value assigned to a line in a file.

[SET] SERIAL ON [incrno [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]]
 [Nonblank [*]]

 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.

XEDIT (continued)

CMS

[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|OFF

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

or

SYNonym [LINEND char] newname [n] oldname

or

SYNonym [LINEND char] newname [n[format1...formatn]]
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 n1[n2 ... n28]

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.

XEDIT

XEDIT (continued)

CMS

[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 OFF [[Hex] startcol endcol] ...

Controls whether or not subcommand(s)-changed lines are to be displayed. Also defines columns to be displayed on screen. Data may also be displayed in hexadecimal.

[SET] WRap ON|OFF

Controls use of wraparound. Note that the initial setting is OFF.

[SET] Zone zone1 zone2
*

Defines starting and ending column of each record for target search scanning.

[SET] = string

Inserts specified string into the equal buffer (see = subcommand).

XEDIT (continued)

CMS

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 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.

SORT target [A] col1 col2 [col1 col2] ...
 [D]

Arranges a specified number of file lines in ascending or descending EBCDIC sequence according to specified sort columns.

SOS option

options:

ALarm	POP
CLEAR	PUSH
LINEAdd	TABB [n <u>1</u>]
LINEDel	TABCMD
NULLs	TABCMDB [n <u>1</u>]
NULLs ON	TABCMDF [n <u>1</u>]
NULLs OFF	TABF [n <u>1</u>]
PFn	

Provides a set of functions used mainly in XEDIT macros or assigned to PF keys.

SPlit [Aligned] [COLUMN]
 [CURSOR]

or

[Aligned] [colno
 [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.

XEDIT

XEDIT (continued)

CMS

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 non-blank character, the line is split (at the cursor position). If the cursor is positioned after the last non-blank 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 $\left[\begin{array}{l} \text{target} \\ \underline{1} \end{array} \left[\begin{array}{l} \text{startcol} \\ \underline{1} \end{array} \left[\begin{array}{l} \text{length} \\ \underline{*} \end{array} \right] \right] \right]$

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).

XEDIT (continued)

CMS

TRAnSfer keyword ...

keywords: (more than one can be specified)

APL	IMage	PFn	TABLine
ARBchar	IMPcmscp	Point	TABS
AUTosave	LASTmsg	PREFix	TARGet
CASE	LENgth	RANge	TERMinal
CMDline	LIne	RECFm	TEXT
COLPtr	LINENd	RESERved	TOF
COLumn	LRecl	SCALE	TOFFEO
CTLchar[char]	LScreen	SCReen	Trunc
CURLine	MACRO	Seq8	UPDate
CURSor	MASK	SERial	VARblank
EOF	MSGMode	SIDcode	Verify
ESCAPE	NBFile	SIZE	VERShift
FILLer	NONDisp	SPAN	Width
FMode	NULLs	STAY	WRap
FName	NUMBER	STReam	Zone
FType	PACK	SYNonym[name]	=
HEX			

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...[...])]

Edits multiple files in virtual storage.

Options: are the same as the command options (see XEDIT command).

XEDIT

XEDIT (continued)

CMS

The **XEDIT Prefix subcommands** (line subcommands) are used as follows:

Prefix	Meaning and/or Action
A	Add one line immediately after the line in which subcommand is entered
nA	Add n lines
An	Add n lines
C	Copy one line; must have F or P prefix subcommand to indicate destination of line
Cn	Copy n lines
nC	Copy n lines
CC	Copy block of lines
D	Deletes one line
Dn	Delete n lines
nD	Delete n lines
DD	Delete block of lines
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 immediately following the line in which prefix subcommand is entered
nl	Insert n lines
In	Insert n lines
M	Move one line from one location to another in the file
Mn	Move n lines
nM	Move n lines
MM	Move block of lines
P	Data is entered preceding this point (using the C or M prefix subcommands)

XEDIT (continued)

CMS

S	Show all lines
S*	Show all lines
Sn	Show the first n lines
S + n	Show the first n lines
nS	Show the first n lines
S-n	Show the last n lines
SCALE	Display 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
Xn	Exclude n lines
nX	Exclude n lines
XX	Exclude a block of lines
.xxxx	Assigns xxxx as symbolic name to this line
<	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 n columns 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
/[n]	Make this line current and move the column pointer under the nth column.
[n]/	
>	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 n columns 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

XEDIT

XEDIT (continued)

CMS

" Duplicates one line (must be a double quote symbol)
"n Duplicate line n times
n" Duplicate line n times
"" Duplicate block of lines
""n Duplicate block n times
n"" Duplicate block n 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 } [SYSLANG]
{NOCOMPress } {NOHEADer } {NONDISPlay
ERRMSG }

ZAP

CMS

Modifies or dumps MODULE, LOADLIB, or TXTLIB files.

```
ZAP { MODULE
    LOADLIB } [libname1 ... libname3][(options...[])]
    TXTLIB }
```

options:

```
[TERM|INPUT filename] [PRINT|NOPRINT]
```

Control Statements:

```
BASE address
```

```
DUMP { membername } csectname [startaddress [endaddress]]
     { modulename } ALL
```

```
LOG [fixnum] [ZAPLOG|filetype [userdata]]
```

```
NAME {membername|modulename} [csectname]
```

```
REP disp data
```

```
{VERIFY|VER} disp data
```

```
* comment
```

```
END
```

Summary of Changes

The HELP Facility

The HELP Facility has been enhanced to include National Language Support and to improve performance and usability. These enhancements include:

- BRIEF layer of HELP for a subset of frequently-used commands
- RELATED information
- New HELP command options
- Toggling ability
- Windowing of the BRIEF layer
- MOREHELP command
- Simplified screens and syntax notation
- New control section keywords
- Enhancement of the DEFAULTS command
- National Language Support

NLS Parser Summary

The CMS parsing facility parses and translates command name arguments. It is important to National Language Support (NLS) because it lets users enter commands in their own national language.

Enhancements for EXECs in Storage

An optional Installation Discontiguous Shared Segment (DCSS) is added to contain frequently used EXECs and Editor Macros that your installation provides. All users can access the DCSS and share the same executing copy of the EXECs.

Advanced Printer Subsystem Support

The SPOOL System Service allows authorized users an interface for communication between CP and a printer subsystem.

The DESTination option allows you to select a specific printer or punch to process your print, punch, or console file.

The CMS PRINT Command has been enhanced to support an Oversize option to print files that have records larger than the virtual printer's character size.

Transparent Services Access Facility

Transparent Services Access Facility is a facility that lets users connect to and communicate with local or remote virtual machines within a group of systems.

Enhanced 3270 Usability

Usability features for 3270 display stations have been enhanced. The VM logo message at the top of the screen is changed from "VM/370" to "VIRTUAL MACHINE/SYSTEM PRODUCT" and users can now log on directly from the logo screen.

New Commands for Release 5

ALARM VSCREEN	QUERY CURSOR	SET LANGUAGE
CLEAR VSCREEN	QUERY DISPLAY	SET LINEND
CLEAR WINDOW	QUERY FULLREAD	SET LOCATION
CONVERT COMMANDS	QUERY FULLSCREEN	SET LOGFILE
CMSSERV	QUERY HIDE	SET NONDISP
CURSOR VSCREEN	QUERY KEY	SET REMOTE
DEFINE VSCREEN	QUERY LANGLIST	SET RESERVED
DEFINE WINDOW	QUERY LANGUAGE	SET VSCREEN
DELETE VSCREEN	QUERY LINEND	SET WINDOW
DELETE WINDOW	QUERY LOCATION	SET WMPF
DISKMAP	QUERY LOGFILE	SHOW WINDOW
DROP WINDOW	QUERY NONDISP	SIZE WINDOW
GET VSCREEN	QUERY REMOTE	SNTMAP
HIDE WINDOW	QUERY RESERVED	SPGEN
ITASK	QUERY SHOW	SPLOAD
LANGGEN	QUERY TEXT	UTILITY
LANGMERG	QUERY VSCREEN	VALIDATE
MAXIMIZE WINDOW	QUERY WINDOW	VMFASM
MINIMIZE WINDOW	QUERY WMPF	VMFDOS
MOREHELP	REFRESH	VMFLKED
PARSECMD	RESTORE WINDOW	VMFLOAD
POP WINDOW	ROUTE	VMFMAC
POSITION WINDOW	SCROLL	VMFMERGE
PRELOAD	SET APL	VMFREMOV
PUT SCREEN	SET BORDER	VMFTEXT
PUT VSCREEN	SET CHARMODE	VMFZAP
QUERY APL	SET CMSPF	WAITREAD VSCREEN
QUERY BORDER	SET FULLREAD	WAITT VSCREEN
QUERY CHARMODE	SET FULLSCREEN	WRITE VSCREEN
QUERY CMSPF		XMITMSG

TSAF Commands

- ADD LINK
- DELETE LINK
- QUERY
- RUNTSAF
- STOP TSAF

Changed Commands for Release 5

ACCESS	IPL
BACKSPAC	LOAD
CHANGE	OVERRIDE
CLOSE	PRINT
DEFAULTS	PURGE
DETACH	QUERY
DRAIN	REPEAT
DUMPSCAN	SAVENCP
EXECDROP	SET
EXECIO	SPOOL
EXECMAP	SPTAPE
EXECUPDT	START
FILEDEF	TERMINAL
FLUSH	TRANSFER
FORMAT	TXTLIB
GLOBAL	UPDATE
HELP	VARY
INCLUDE	XEDIT

Bibliography

Prerequisite Publications

Virtual Machine/System Product:

CMS Command Reference, SC19-6209

CMS Primer, SC24-5236

CMS Primer for Line-Oriented Terminals, SC24-5242

CMS User's Guide, SC19-6210

CP Command Reference, SC19-6211

Introduction, GC19-6200

Corequisite Publications

Virtual Machine/System Product:

Application Development Guide, SC24-5247

VM/SP CMS for System Programming, SC24-5286

VM/SP CP for System Programming, SC24-5285

EXEC 2 Reference, SC24-5219

VM/SP System Facilities for Programming, SC24-5288

*VM/SP Transparent Services Access Facility Reference,
SC24-5287*

*IBM Cooperative Processing Programmer's Guide for VM/SP,
SC24-5291*

Installation Guide, SC24-5237

Macros and Functions Reference, SC24-5284

Operator's Guide, SC19-6202

Planning Guide and Reference, SC19-6201

VM Distributed Data Processing Guide, LY24-2451

*Remote Spooling Communications Subsystem Networking
Program Reference and Operations Manual, SH24-5005*

System Messages and Codes, SC19-6204

System Messages Cross-Reference, SC19-5264

***System Product Editor Command and Macro Reference,
SC24-5221***

System Product Editor User's Guide, SC24-5220

System Product Interpreter Reference, SC24-5239

System Product Interpreter User's Guide, SC24-5238

Terminal Reference, GC19-6206

Index

Special Characters

* command (CMS) 53
* command (CP class any) 53
* command (GCS) 53
* command (RSCS) 53
#CP command (CP class any) 53

A

ACCESS command (CMS) 54
ACCESS command (GCS) 54
ACNT command (CP class A) 54
ADD LINK command (TSAF) 55
ADSTOP command (CP class G) 55
ALARM VSCREEN command (CMS) 55
AMSERV command (CMS) 55
APAR command (IPCS) 56
ASMGEND command (CMS) 56
ASM3705 command (CMS) 56
ASSEMBLE command (CMS) 57
assembling a source file 24
ASSGN command (CMS) 58
ATTACH command (CP class B) 58
ATTN command (CP class G) 58
AUTOLOG command (CP class A,B) 59

B

B (CMS Border Command) 59
BACKSPAC command (CP class D) 59
BACKSPAC command (RSCS) 59
BEGIN command (CP class G) 59

C

C (CMS Border Command) 60
CATCHCHECK command (CMS) 60
CHANGE command (CP class D) 61
CHANGE command (CP class G) 62
CHANGE command (RSCS) 63
CLEAR VSCREEN command (CMS) 63
CLEAR WINDOW command (CMS) 63
CLOSE command (CP class G) 64
CMD command (RSCS) 64
CMDCALL command (CMS) 64
CMS (Conversational Monitor Commands)
commands
* 53
ACCESS 54
ALARM VSCREEN 55
AMSERV 55
ASMGEND 56
ASM3705 56
ASSEMBLE 57
ASSGN 58
B (CMS Border Command) 59
C (CMS Border Command) 60
CATCHCHECK 60
CLEAR VSCREEN 63
CLEAR WINDOW 63
CMDCALL 64

CMSBATCH	64	FINIS	127
CMSGEND	65	FORMAT	128
CMSSERV	65	GENDIRT	130
COMPARE	65	GENIMAGE	130
CONVERT COMMANDS	66	GENMOD	131
CONWAIT	66	GENMSG	131
COPYFILE	67	GEN3705	131
CP	67	GET VSCREEN	132
CURSOR VSCREEN	69	GLOBAL	132
D (CMS Border Command)	69	GLOBALV	133
DDR	70	H (CMS Border Command)	134
DEBUG	72	HB	134
DEFAULTS	73	HELP	135
DEFINE VSCREEN	77	HELPCONV	135
DEFINE WINDOW	78	HI	135
DELETE VSCREEN	78	HIDE WINDOW	136
DELETE WINDOW	79	HO	136
DESBUF	79	HT	136
DIRECT	80	HX	137
DISK	82	IDENTIFY	137
DISKMAP	82	IMAGELIB	137
DLBL	84	IMAGEMOD	138
DOSLIB	86	IMMCMD	138
DOSLKED	86	INCLUDE	139
DROP WINDOW	87	ITASK (CMS)	141
DROPBUF	87	L (CMS Border Command)	142
DSERV	88	LABELDEF	142
EDIT	95	LANGGEN	143
ERASE	102	LANGMERG	143
ESERV	102	LISTDS	143
EXEC	103	LISTFILE	144
EXEC 2	108	LISTIO	144
EXECDROP	118	LKED	145
EXECIO	119	LOAD	145
EXECLOAD	120	LOADLIB	146
EXECMAP	120	LOADMOD	146
EXECOS	121	M (CMS Border Command)	148
EXECSTAT	121	MACLIB	148
EXECUPDT	121	MAKEBUF	149
EXPAND	122	MAXIMIZE WINDOW	149
F (CMS Border Command)	122	MINIMIZE WINDOW	150
FETCH	123	MODMAP	150
FILEDEF	124	MOREHELP	152
FILELIST	126		

MOVEFILE	152	SENTRIES	200
N (CMS Border Command)	153	SET	200
NAMEFIND	153	SET ABBREV	201
NAMES	154	SET APL	201
NCPDUMP	154	SET AUTOREAD	201
NOTE	157	SET BLIP	201
NUCXDROP	157	SET BORDER	202
NUCXLOAD	158	SET CHARMODE	202
NUCXMAP	158	SET CMSPF	202
O (CMS Border Command)	158	SET CMSTYPE	203
OPTION	158	SET DOS	203
OSRUN	159	SET DOSLNCNT	203
P (CMS Border Command)	160	SET DOSPART	203
PARSECMD	160	SET EXECTRAC	203
PEEK	161	SET FULLREAD	204
POP WINDOW	163	SET FULLSCREEN	204
POSITION WINDOW	163	SET IMESCAPE	204
PRELOAD	164	SET IMPCP	204
PRINT	164	SET IMPEX	205
PSERV	168	SET INPUT	205
PUNCH	168	SET INSTSEG	205
PUT SCREEN	169	SET LANGUAGE	205
PUT VSCREEN	169	SET LDRTBLS	206
QUERY	170	SET LINEND	206
R (CMS Border Command)	179	SET LOCATION	206
RDR	179	SET LOGFILE	206
RDRLIST	179	SET NONDISP	206
READCARD	180	SET NONSHARE	207
RECEIVE	180	SET OUTPUT	207
REFRESH	181	SET PROTECT	207
RELEASE	181	SET RDYMSG	207
RENAME	182	SET REDTYPE	208
RESERVE	183	SET RELPAGE	208
RESTORE WINDOW	183	SET REMOTE	208
REXX	184	SET RESERVED	208
RO	195	SET SYSNAME	209
ROUTE	195	SET TEXT	209
RSERV	196	SET TRANSLATE	209
RT	196	SET UPSI	209
RUN	196	SET VSCREEN	210
S (CMS Border Command)	197	SET WINDOW	210
SAVENCP	197	SET WMPF	210
SCROLL	199	SETKEY	215
SENDFILE	200	SETPRT	215

SHOW WINDOW 215
 SIZE WINDOW 216
 SNTMAP 217
 SO 217
 SORT 217
 SPGEN 218
 SPLOAD 219
 SSERV 222
 START 222
 STATE/STATEW 224
 SVCTRACE 226
 SYNONYM 226
 TAPE 228
 TAPEMAC 229
 TAPPDS 229
 TE 230
 TELL 230
 TRAPRED 234
 TS 234
 TXTLIB 235
 TYPE 235
 UPDATE 236
 UTILITY 236
 VALIDATE 237
 VMFASM 238
 VMFDOS 238
 VMFLKED 239
 VMFLOAD 239
 VMFMAC 239
 VMFMERGE 239
 VMFNLS 240
 VMFPLC2 240
 VMFREMOVE 241
 VMFTXT 241
 VMFZAP 241
 WAITREAD VSCREEN 242
 WAITT VSCREEN 242
 WRITE VSCREEN 243
 X (CMS Border Command) 243
 XEDIT 244
 XMITMSG 270
 ZAP 271
 CMS Border Commands
 B 59
 C 60
 D 69
 F 122
 H 134
 L 142
 M 148
 N 153
 P 160
 R 179
 S 197
 X 243
 CMS DDR command return codes 42
 CMS fileid 32
 CMS immediate commands
 HB 134
 HI 135
 HO 136
 HT 136
 HX 137
 RO 195
 RT 196
 SO 217
 TE 230
 TS 234
 CMS program development facilities 22
 CMS return codes 39
 CMSBATCH command (CMS) 64
 MSGEND command (CMS) 65
 CMSSERV (CMS command) 65
 COMPARE command (CMS) 65
 Control Program (CP)
 See CP (Control Program) commands
 conventions for notations 48
 Conversational Monitor System (CMS)
 See CMS (Conversational Monitor
 Commands) commands
 CONVERT command (IPCS) 66
 CONVERT COMMANDS (CMS) 66
 CONVIPCS EXEC (IPCS) 66
 CONWAIT command (CMS) 66
 COPYFILE command (CMS) 67
 correcting errors 25
 COUPLE command (CP class G) 67

CP (Control Program) commands

* (class any) 53
#CP (class any) 53
ACNT (class A) 54
ADSTOP (class G) 55
ATTACH (class B) 58
ATTN (class G) 58
AUTOLOG (class A,B) 59
BACKSPAC (class D) 59
BEGIN (class G) 59
CHANGE (class D) 61
CHANGE (class G) 62
CLOSE (class G) 64
COUPLE (class G) 67
CP (class any) 67
CPTRAP (class C) 69
DCP (class C,E) 70
DEFINE (class A,B) 74
DEFINE (class G) 75
DETACH (class B) 79
DETACH (class G) 79
DIAL (class any) 80
DISABLE (class A,B) 81
DISCONN (class any) 81
DISPLAY (class G) 83
DMCP (class C,E) 85
DRAIN (class D) 87
DUMP (class G) 88
ECHO (class G) 94
ENABLE (class A,B) 101
EXTERNAL (class G) 122
FLUSH (class D) 127
FORCE (class A) 127
FREE (class D) 129
HALT (class A) 134
HOLD (class D) 136
INDICATE (class A) 139
INDICATE (class E) 139
INDICATE (class G) 140
IPL (class G) 140
LINK (class G) 143
LOADBUF (class D) 146
LOADVFCB (class G) 147
LOCATE (class E) 147

LOCK (class A) 147
LOGOFF (class any) 147
LOGON (class any) 147
MESSAGE (class A,B) 149
MESSAGE (class any) 149
MIGRATE (class A) 150
MONITOR (class A,E) 150
MSGNOH (class A,B) 153
NETWORK (class A) 155
NETWORK (class B) 156
NOTREADY (class G) 157
ORDER (class D) 159
ORDER (class G) 159
PER (class A,B,C,D,E,F,G) 161
PURGE (class D) 168
PURGE (class G) 169
QUERY (class A) 172
QUERY (class B) 173
QUERY (class C) 173
QUERY (class D) 174
QUERY (class E) 175
QUERY (class F) 175
QUERY (class G) 176
QVM (class A) 179
READY (class G) 180
REPEAT (class D) 182
REQUEST (class G) 182
RESET (class G) 183
REWIND (class G) 184
SAVESYS (class E) 197
SCREEN (class G) 198
SEND (class G) 199
SET (class A) 211
SET (class B) 211
SET (class E) 212
SET (class F) 212
SET (class G) 213
SHUTDOWN (class A) 215
SLEEP (class any) 216
MSG (class G) 216
MSG (RSCS) 217
SPACE (class D) 218
SPMODE (class A) 219
SPOOL (class G) 220

SPTAPE (class D) 221
 START (class D) 222
 STCP (class C) 225
 STORE (class G) 226
 SYSTEM (class G) 227
 TAG (class G) 227
 TERMINAL (class G) 231
 TRACE (class G) 232
 TRANSFER (class D) 232
 TRANSFER (class G) 233
 UNLOCK (class A) 235
 VARY (class B) 237
 VMDUMP (class G) 237
 WARNING (class A,B) 242
 CP command (CMS) 67
 CP command (CP class any) 67
 CP command (RSCS) 68
 CP DIRECT return codes 41
 CP privilege classes 50-53
 CPQUERY command (RSCS) 68
 CPTRAP command (CP class C) 69
 creating an assembler language source
 file 22
 assembling a source file 24
 correcting errors 25
 erasing unwanted files 28
 testing and correcting 26
 CURSOR VSCREEN command (CMS) 69

D

D (CMS Border Command) 69
 DCP command (CP class C,E) 70
 DDR command (CMS) 70
 DEBUG command (CMS) 72
 subcommands 72
 BReak 72
 CAW 72
 CSW 72
 DEFine 72
 DUmp 72

GO 72
 GPR 73
 HX 73
 ORigin 73
 PSW 73
 RETurn 73
 SET 73
 STore 73
 X 73
 DEFAULTS command (CMS) 73
 DEFINE command (CP class A,B) 74
 DEFINE command (CP class G) 75
 DEFINE command (RSCS) 77
 DEFINE VSCREEN command (CMS) 77
 DEFINE WINDOW command (CMS) 78
 DELETE command (RSCS) 78
 DELETE LINK command (TSAF) 78
 DELETE VSCREEN command (CMS) 78
 DELETE WINDOW command (CMS) 79
 DESBUF command (CMS) 79
 DETACH command (CP class B) 79
 DETACH command (CP class G) 79
 developing program facilities 24
 developing programs 22
 DIAL command (CP class any) 80
 DIRECT command (CMS) 80
 DISABLE command (CP class A,B) 81
 DISCONN command (CP class any) 81
 DISCONN command (RSCS) 82
 disconnect function 9
 DISCONNECT status 9
 disconnection of terminal 9
 DISK command (CMS) 82
 DISKMAP command (CMS) 82
 DISPLAY command (CP class G) 83
 display terminals, status action 11
 DLBL command (CMS) 84
 DLBL command (GCS) 85
 DMCP command (CP class C,E) 85
 DOSLIB command (CMS) 86
 DOSLKED command (CMS) 86
 DRAIN command (CP class D) 87
 DRAIN command (RSCS) 87

DROP WINDOW command (CMS) 87
 DROPBUF command (CMS) 87
 DSERV command (CMS) 88
 DUMP command (CP class G) 88
 DUMPSCAN command (GCS) 89
 DUMPSCAN command (IPCS) 90
 subcommands 90-93
 + (increment) 90
 (null line) 90
 & 90
 &name 90
 - (decrement) 90
 ? 90
 Aregs 90
 ARIOBLOK 90
 C 90
 CHain 91
 CMS 91
 CMSPoint 91
 CORtable 91
 Display 91
 DOSPoint 91
 DUMPID 91
 END 91
 FORMat 94
 G 92
 HELP 92
 HEX 94
 HX 92
 IPCSMAP 92
 Locate 92
 Locate Up 92
 MAPA 93
 MAPN 93
 Mregs 93
 MRIOBLOK 93
 OSPoint 93
 Print 93
 PRT 93
 QUIT 93
 Regs 93
 RIOblok 93
 Scroll 94
 Scrollu 94

SYMP 94
 USERMAP 94
 VIOblok 94
 Vmblok 94

E

ECHO command (CP class G) 94
 EDIT command (CMS) 95
 subcommands and macros 95-101
 &DUP 95
 &MOVE 95
 ? 95
 ALter 95
 AUTOsave 96
 Backward 96
 Bottom 96
 CASE 96
 Change 96
 CMS 96
 DELeTe 96
 Down 96
 DString 96
 FILE 97
 Find 97
 FMode 97
 FName 97
 FORMat 97
 FORward 97
 Getfile 97
 IMAGE 97
 Input 98
 LINEmode 98
 Locate 98
 LONG 98
 Next 98
 nnnnnnnn 101
 Overlay 98
 PREserve 98
 PROMPT 98
 QUIT 98

RECfm 99
 REnum 99
 REPEAT 99
 Replace 99
 REStore 99
 RETURN 99
 REUSE 99
 SAVE 99
 Scroll 100
 SERial 100
 SHORT 100
 STACK 100
 TABSet 100
 TOP 100
 TRUNC 100
 Type 100
 Up 101
 Verify 101
 X 101
 Y 101
 Zone 101
 ENABLE command (CP class A,B) 101
 ERASE command (CMS) 102
 erasing unwanted files 28
 ESERV command (CMS) 102
 ETRACE command (GCS) 102
 EXEC command (CMS) 103
 built-in functions 106
 &LITERAL 106
 &variable = &CONCAT 106
 &variable = &DATATYPE 106
 &variable = &LENGTH 106
 &variable = &SUBSTR 106
 control statements 103
 &ARGS 103
 &BEGEMSG 103
 &BEGPUNCH 103
 &BEGSTACK 103
 &BEGTYPE 103
 &CONTINUE 103
 &CONTROL 104
 &EMSG 104
 &END 104
 &ERROR 104
 &EXIT 104
 &GOTO 104
 &HEX 104
 &IF 104
 &LOOP 104
 &PUNCH 105
 &READ 105
 &SKIP 105
 &SPACE 105
 &STACK 105
 &TIME 105
 &TYPE 105
 &variable 103
 special variables 106
 &* and &\$ 106
 &0 106
 &DISK* 106
 &DISK? 106
 &DISKx 106
 &DOS 106
 &EXEC 106
 &GLOBAL 107
 &GLOBALn 107
 &INDEX 107
 &LINENUM 107
 &n 106
 &READFLAG 107
 EXEC command (RSCS) 107
 EXEC 2 command (CMS) 108-117
 control statements 110
 &ARGS 110
 &BEGPRINT 110
 &BEGSTACK 110
 &BEGTYPE 110
 &BUFFER 111
 &CALL 111
 &CASE 111
 &COMMAND 111
 &CRASH 111
 &DUMP 111
 &ERROR 112
 &EXIT 112
 &GOTO 112

- &IF 112
- &LOOP 113
- &PRESUME 113
- &PRINT 113
- &READ 113
- &RTURN 114
- &SKIP 114
- &STACK 114
- &SUBCOMMAND 114
- &TRACE 114
- &TRUNC 115
- &TYPE 113
- &UPPER 115
- predefined functions 115
 - &CONCAT 115
 - &CONCATENATION 115
 - &DATATYPE 115
 - &DIV 115
 - &DIVISION 115
 - &LEFT OF 116
 - &LENGTH OF 116
 - &LITERAL OF 116
 - &LOCATION OF 116
 - &MULT 116
 - &MULTIPLICATION OF 116
 - &PIECE 116
 - &POSITION OF 116
 - &RANGE OF 117
 - &RIGHT OF 117
 - &STRING OF 117
 - &SUBSTR 116
 - &TRANS 117
 - &TRANSLATION 117
 - &TRIM OF 117
 - &TYPE 115
 - &WORD OF 117
- predefined variables 108
 - & 108
 - &0 108
 - &ARGSTRING 108
 - &BLANK 108
 - &CMDSTRING 108
 - &COMLINE 108
 - &DATE 108

- &DEPTH 109
- &FILEMODE 109
- &FILENAME 109
- &FILETYPE 109
- &FROM 109
- &LINE,&ILINENUM 109
- &LINK 109
- &N.&INDEX 109
- &RC,&RETCODE 109
- &TIME 109
- &1,&2,&n 108
- user-defined functions 117
 - label 117
 - line-number OF 117
- EXEC 2 Interpreter 17
- EXECDROP command (CMS) 118
- EXECIO command (CMS) 119
- EXECLOAD command (CMS) 120
- EXECMAP command (CMS) 120
- EXECOS command (CMS) 121
- EXECUPDT command (CMS) 121
- EXESTAT command (CMS) 121
- EXPAND command (CMS) 122
- EXTERNAL command (CP class G) 122

F

- F (CMS Border Command) 122
- FETCH command (CMS) 123
- FILEDEF command (CMS) 124
- FILEDEF command (GCS) 126
- FILELIST command (CMS) 126
- FINIS command (CMS) 127
- FLUSH command (CP class D) 127
- FLUSH command (RSCS) 127
- FORCE command (CP class A) 127
- FORCE command (RSCS) 128
- FORMAT command (CMS) 128
- FORMAT/ALLOCATE command (service aid) 129
- FREE command (CP class D) 129

FREE command (RSCS) 129
FWDSpace command (RSCS) 130

G

GCS (Group Control System) Ccommands

* 53
ACCESS 54
DLBL 85
DUMPSCAN 89
ETRACE 102
FILEDEF 126
GDUMP 130
GLOBAL 132
HX 137
ITRACE (GCS) 142
LOADCMD 146
OSRUN 160
QUERY 177
RELEASE 181
REPLY 182
SET 214
GDUMP command (GCS) 130
GENDIRT command (CMS) 130
GENIMAGE command (CMS) 130
GENMOD command (CMS) 131
GENMSG command (CMS) 131
GEN3705 command (CMS) 131
GET VSCREEN command (CMS) 132
GLOBAL command (CMS) 132
GLOBAL command (GCS) 132
GLOBALV command (CMS) 133
Group Control System (GCS)
 See GCS (Group Control System)
 Ccommands
GTRACE command (RSCS) 134

H

H (CMS Border Command) 134
HALT command (CP class A) 134
HB command (CMS) 134
HELP command (CMS) 135
HELPCONV command (CMS) 135
HI command (CMS) 135
HIDE WINDOW command (CMS) 136
HO command (CMS) 136
HOLD command (CP class D) 136
HOLD command (RSCS) 136
how to IPL CMS 16
HT command (CMS) 136
HX command (CMS) 137
HX command (GCS) 137

I

IDENTIFY command (CMS) 137
IMAGELIB command (CMS) 137
IMAGEMOD command (CMS) 138
IMMCMD command (CMS) 138
INCLUDE command (CMS) 139
INDICATE command (CP class A) 139
INDICATE command (CP class E) 139
INDICATE command (CP class G) 140
INIT command (RSCS) 140
input conventions 10
Interactive Problem Control System
 See IPCS commands
IPCS commands
 APAR 56
 CONVERT 66
 CONVIPCS EXEC 66
 DUMPSCAN 90
 + (increment) 90
 (null line) 90
 & 90

&name 90
 - (decrement) 90
 ? 90
 Aregs 90
 ARIOBLOK 90
 C 90
 CHain 91
 CMS 91
 CMSPoint 91
 CORtable 91
 Display 91
 DOSPoint 91
 DUMPID 91
 END 91
 FORMat 94
 G 92
 HELP 92
 HEX 94
 HX 92
 IPCSMAP 92
 Locate 92
 Locate Up 92
 MAPA 93
 Mregs 93
 MRIOBLOK 93
 OSPoint 93
 Print 93
 PRT 93
 QUIT 93
 Regs 93
 RIOblok 93
 Scroll 94
 Scrollu 94
 SYMP 94
 USERMAP 94
 VIOblok 94
 Vmblok 94
 IPCSDUMP 140
 MAP 149

PRB 163
 PROB 164
 PRTDUMP 167
 STAT 224
 IPCS commands return codes 45
 IPCSDUMP command (IPCS) 140
 IPL command (CP class G) 140
 ITASK command (CMS) 141
 ITRACE command (GCS) 142



L (CMS Border Command) 142
 LABELDEF command (CMS) 142
 LANGGEN command (CMS) 143
 LANGMERG command (CMS) 143
 line length 11
 line termination 11
 LINK command (CP class G) 143
 LISTDS command (CMS) 143
 LISTFILE command (CMS) 144
 LISTIO command (CMS) 144
 LKED command (CMS) 145
 LOAD command (CMS) 145
 LOADBUF command (CP class D) 146
 LOADCMD command (GCS) 146
 LOADLIB command (CMS) 146
 LOADMOD command (CMS) 146
 LOADVFCB command (CP class G) 147
 LOCATE command (CP class C,E) 147
 LOCK command (CP class A) 147
 logging off 8
 logging on 3
 logical line-editing characters 10
 logoff and security 9
 LOGOFF command (CP class any) 147
 LOGON command (CP class any) 147

M

M (CMS Border Command) 148
 MACLIB command (CMS) 148
 MAKEBUF command (CMS) 149
 MAP command (IPCS) 149
 MAXIMIZE WINDOW command (CMS) 149
 MESSAGE command (CP class A,B) 149
 MESSAGE command (CP class Any) 149
 MIGRATE command (CP class A) 150
 MINIMIZE WINDOW command (CMS) 150
 MODMAP command (CMS) 150
 MONITOR command (CP class A,E) 150
 MOREHELP command (CMS) 152
 MOVEFILE command (CMS) 152
 MSG command (RSCS) 153
 MSGNOH command (CP class A,B) 153

N

N (CMS Border Command) 153
 NAMEFIND command (CMS) 153
 NAMES command (CMS) 154
 NCPDUMP command (CMS) 154
 NETWORK command (CP class A) 155
 NETWORK command (CP class B) 156
 NETWORK command (RSCS) 157
 notational conventions 48
 notational usage 48
 NOTE command (CMS) 157
 NOTREADY command (CP class G) 157
 NUCXDROP command (CMS) 157
 NUCXLOAD command (CMS) 158
 NUCXMAP command (CMS) 158

O

O (CMS Border Command) 158
 OPTION command (CMS) 158
 ORDER command (CP class D) 159
 ORDER command (CP class G) 159
 ORDER command (RSCS) 159
 OSRUN command (CMS) 159
 OSRUN command (GCS) 160

P

P (CMS Border Command) 160
 PARSECMD command (CMS) 160
 PEEK command (CMS) 161
 PER command (CP class A,B,C,D,E,F,G) 161
 POP WINDOW command (CMS) 163
 POSITION WINDOW command (CMS) 163
 PRB command (IPCS) 163
 PRELOAD command (CMS) 164
 PRINT command (CMS) 164
 printing files 29
 printing, punching and reading files 29
 privilege classes 50
 PROB command (IPCS) 164
 program development facilities 22
 PROGRAMMABLE OPERATOR (CMS) 165
 commands 165-166
 CMD 165
 FB 166
 FEEDBACK 166
 GET 166
 LGLOPR 166
 LOADTBL 166
 LOG 166
 QUERY HOSTCHK 166
 QUERY LGLOPR 166
 QUERY LOGGING 166
 QUERY PROPCHK 166

QUERY RTABLE 166
 SET DEBUG 167
 SET HOSTCHK 167
 SET LOGGING 167
 SET PROPCHK 167
 PRTDUMP command (IPCS) 167
 PSERV command (CMS) 168
 PUNCH command (CMS) 168
 punching files 29
 PURGE command (CP class D) 168
 PURGE command (CP class G) 169
 PURGE command (RSCS) 169
 PUT SCREEN command (CMS) 169
 PUT VSCREEN command (CMS) 169

Q

QUERY command (CMS) 170
 QUERY command (CP class A) 172
 QUERY command (CP class B) 173
 QUERY command (CP class C) 173
 QUERY command (CP class D) 174
 QUERY command (CP class E) 175
 QUERY command (CP class F) 175
 QUERY command (CP class G) 176
 QUERY command (GCS) 177
 QUERY command (RSCS) 178
 QUERY command (TSAF) 178
 QVM command (CP class A) 179

R

R (CMS Border Command) 179
 RDR command (CMS) 179
 RDRLIST command (CMS) 179
 READCARD command (CMS) 180
 reading files 29
 READY command (CP class G) 180
 READY command (RSCS) 180

RECEIVE command (CMS) 180
 RECONN command (RSCS) 181
 REFRESH command (CMS) 181
 RELEASE command (CMS) 181
 RELEASE command (GCS) 181
 Remote Spooling Communications Subsystem (RSCS)
 See RSCS (Remote Spooling Communications Subsystem) commands
 RENAME command (CMS) 182
 REPEAT command (CP class D) 182
 REPLY command (GCS) 182
 REQUEST command (CP class G) 182
 REROUTE command (RSCS) 183
 RESERVE command (CMS) 183
 reserved filetype descriptions 34
 RESET command (CP class G) 183
 RESTORE WINDOW command (CMS) 183
 return codes 39
 CMS DDR command return codes 42
 CMS return codes 39
 CP DIRECT return codes 41
 IPCS commands return codes 45
 System Product Editor command return codes 42
 XEDIT command return codes 42
 REWIND command (CP class G) 184
 REXX language (CMS) 184
 REXX built-in functions 190-193
 ABBREV 190
 ABS 190
 Address 190
 Arg 190
 BITAND 190
 BITOR 190
 BITXOR 190
 CENTER 190
 CENTRE 190
 COMPARE 190
 COPIES 190
 C2D 191
 C2X 191
 DATATYPE 191
 DATE 191

DELSTR 191
DELWORD 191
D2C 191
D2X 191
ERRORTXT 191
EXTERNALS 191
FIND 191
FORMAT 191
INDEX 192
INSERT 192
JUSTIFY 192
LASTPOS 192
LEFT 192
LENGTH 192
LINESIZE 192
MAX 192
MIN 192
OVERLAY 192
POS 192
QUEUED 192
RANDOM 192
REVERSE 192
RIGHT 193
SIGN 193
SOURCELINE 193
SPACE 193
STRIP 193
SUBSTR 193
SUBWORD 193
SYMBOL 193
TIME 193
TRACE 193
TRANSLATE 193
TRUNC 193
USERID 193
VALUE 193
VERIFY 194
WORD 194
WORDINDEX 194
WORDLENGTH 194
WORDS 194
XRANGE 194
X2C 194
X2D 194

REXX instructions 184-188
ADDRESS 184
ARG 184
CALL 184
DO 185
DROP 185
EXIT 185
IF-THEN-ELSE 186
INTERPRET 186
ITERATE 186
LEAVE 186
NOP 186
NUMERIC DIGITS 186
NUMERIC FORM 186
NUMERIC FUZZ 186
PARSE 187
PROCEDURE 187
PULL 187
PUSH 187
QUEUE 187
RETURN 187
SAY 187
SELECT 188
SIGNAL 188
TRACE 189
UPPER 189
RSXYSFN package 195
CMSFLAG 195
DIAG 195
DIAGRC 195
STORAGE 195
RO command (CMS) 195
ROUTE command (CMS) 195
ROUTE command (RSCS) 196
RSCS (Remote Spooling Communications
Subsystem) commands
* 53
BACKSPAC 59
CHANGE 63
CMD 64
CP 68
CPQUERY 68
DEFINE 77

DELETE 78
 DISCONN 82
 DRAIN 87
 EXEC 107
 FLUSH 127
 FORCE 128
 FREE 129
 FWDSPACE 130
 GTRACE 134
 HOLD 136
 INIT 140
 MSG 153
 NETWORK 157
 ORDER 159
 PURGE 169
 QUERY 178
 READY 180
 RECONN 181
 REROUTE 183
 ROUTE 196
 SET 214
 SHUTDOWN 216
 START 223
 STOP 225
 TRACE 232
 TRANSFER 233
 RSERV command (CMS) 196
 RT command (CMS) 196
 RUN command (CMS) 196
 RUNTSAF command (TSAF) 197

S

S (CMS Border Command) 197
 SAVENCP command (CMS) 197
 SAVESYS command (CP class E) 197
 SCREEN command (CP class G) 198
 SCROLL command (CMS) 199
 security
 security and logoff 9
 SEND command (CP class G) 199
 SENDFILE command (CMS) 200

sending files 30
 SENTRIES command (CMS) 200
 service aid
 FORMAT/ALLOCATE 129
 SET ABBREV command (CMS) 201
 SET APL command (CMS) 201
 SET AUTOREAD command (CMS) 201
 SET BLIP command (CMS) 201
 SET BORDER command (CMS) 202
 SET CHARMODE command (CMS) 202
 SET CMSPF command (CMS) 202
 SET CMSTYPE command (CMS) 203
 SET command (CMS) 200
 SET command (CP class A) 211
 SET command (CP class B) 211
 SET command (CP class E) 212
 SET command (CP class F) 212
 SET command (CP class G) 213
 SET command (RSCS) 214
 SET command (SET) 214
 SET DOS command (CMS) 203
 SET DOSLNCNT command (CMS) 203
 SET DOSPART command (CMS) 203
 SET ETRACE command (TSAF) 214
 SET EXECTRAC command (CMS) 203
 SET FULLREAD command (CMS) 204
 SET FULLSCREEN command (CMS) 204
 SET IMESCAPE command (CMS) 204
 SET IMPCP command (CMS) 204
 SET IMPEX command (CMS) 205
 SET INPUT command (CMS) 205
 SET INSTSEG command (CMS) 205
 SET LANGUAGE command (CMS) 205
 SET LDRTBLS command (CMS) 206
 SET LINEND command (CMS) 206
 SET LOCATION command (CMS) 206
 SET LOGFILE command (CMS) 206
 SET NONDISP command (CMS) 206
 SET NONSHARE command (CMS) 207
 SET OUTPUT command (CMS) 207
 SET PROTECT command (CMS) 207
 SET RDYMSG command (CMS) 207
 SET REDTYPE command (CMS) 208
 SET RELPAGE command (CMS) 208

SET REMOTE command (CMS) 208
 SET RESERVED command (CMS) 208
 SET SYSNAME command (CMS) 209
 SET TEXT command (CMS) 209
 SET TRANSLATE command (CMS) 209
 SET UPSI command (CMS) 209
 SET VSCREEN command (CMS) 210
 SET WINDOW command (CMS) 210
 SET WMPF command (CMS) 210
 SETKEY command (CMS) 215
 SETPRT command (CMS) 215
 SHOW WINDOW command (CMS) 215
 SHUTDOWN command (CP class A) 215
 SHUTDOWN command (RSCS) 216
 SIZE WINDOW command (CMS) 216
 SLEEP command (CP class any) 216
 SMSG command (CP class G) 216
 SMSG command (RSCS) 217
 SNTMAP command (CMS) 217
 SO command (CMS) 217
 SORT command (CMS) 217
 SPACE command (CP class D) 218
 SPGEN command (CMS) 218
 SPLOAD command (CMS) 219
 SPMODE command (CP class A) 219
 SPOOL command (CP class G) 220
 SPTAPE command (CP class D) 221
 SSERV command (CMS) 222
 START command (CMS) 222
 START command (CP class D) 222
 START command (RSCS) 223
 STAT command (IPCS) 224
 STATE/STATEW command (CMS) 224
 status action on display terminals 11
 STCP command (CP class C) 225
 STOP command (RSCS) 225
 STOP TSAF command (TSAF) 225
 STORE command (CP class G) 226
 summary of status action on display terminals 11
 SVCTRACE command (CMS) 226
 SYNONYM command (CMS) 226
 SYSTEM command (CP class G) 227

System Product Editor 16
 System Product Editor command return codes 42
 System Product Interpreter 18



TAG command (CP class G) 227
 TAPE command (CMS) 228
 TAPEMAC command (CMS) 229
 TAPPDS command (CMS) 229
 TE command (CMS) 230
 TELL command (CMS) 230
 TERMINAL command (CP class G) 231
 terminal operating procedures

- disconnecting your terminal 9
- holding line 8
- logging off 8
- logging on 3

 testing and correcting a program 26
 TRACE command (CP class G) 232
 TRACE command (RSCS) 232
 TRANSFER command (CP class D) 232
 TRANSFER command (CP class G) 233
 TRANSFER command (RSCS) 233
 Transparent Services Access Facility (TSAF)

- See TSAF (Transparent Services Access Facility) commands

 TRAPRED command (CMS) 234
 TS command (CMS) 234
 TSAF (Transparent Services Access Facility) commands

- ADD LINK 55
- DELETE LINK 78
- QUERY 178
- RUNTSAF 197
- SET ETRACE 214
- STOP TSAF 225

 TXTLIB command (CMS) 235
 TYPE command (CMS) 235

U

- UNLOCK command (CP class A) 235
- UPDATE command (CMS) 236
 - using CMS 38
 - CMS file system 33
 - CMS fileids 32
 - example of CMS program development facilities 22-31
 - assembling a source file 24
 - correcting errors 25
 - creating a load module 26
 - creating an assembler language source file 22
 - erasing unwanted files 28
 - printing, punching and reading files 29
 - testing and correcting a program 26
- EXEC 2 Interpreter 17
 - how to IPL CMS 16
 - notational conventions 48
 - reserved filetype descriptions 34
 - return codes 39-46
 - System Product Editor 16
 - System Product Interpreter 18
- using the VM/SP system
 - input conventions 10
 - line length 11
 - line termination 11
 - logical line-editing characters 10
 - virtual machine configuration 1
- UTILITY command (CMS) 236

V

- VALIDATE command (CMS) 237
- VARY command (CP class B) 237
- virtual machine configuration 1
- VMDUMP command (CP class G) 237
- VMFASM command (CMS) 238

- VMFDOS command (CMS) 238
- VMFLKED command (CMS) 239
- VMFLOAD command (CMS) 239
- VMFMAC command (CMS) 239
- VMFMERGE command (CMS) 239
- VMFNLS command (CMS) 240
- VMFPLC2 command (CMS) 240
- VMFREMOV command (CMS) 241
- VMFTXT command (CMS) 241
- VMFZAP command (CMS) 241

W

- WAITREAD VSCREEN command (CMS) 242
- WAITT VSCREEN command (CMS) 242
- WARNING command (CP class A,B) 242
- WRITE VSCREEN command (CMS) 243

X

- X (CMS Border Command) 243
- XEDIT command (CMS) 244-270
 - prefix subcommands 268
 - .xxxx 269
 - < 269
 - /[n] 269
 - > 269
 - " 270
 - A 268
 - C 268
 - D 268
 - E 268
 - F 268
 - I 268
 - M 268
 - P 268
 - S 269
 - SCALE 269
 - SI 269

TABL	269	Join	250
X	269	LEft	250
subcommands and macros	244	LOAD	250
&	244	Locate	250
?	244	LOWercas	250
=	244	LPrefix	250
Add	244	MACRO	250
ALL	244	MErge	251
ALter	245	MOdify	251
Backward	245	MOve	251
Bottom	245	MSG	251
CANcel	245	Next	252
CAppend	245	NFind	252
CDelete	245	NFindUp	252
CFirst	245	NFU	252
Change	245	OverlAy	252
CInserT	246	PARSE	252
CLAst	246	POWERinp	252
CLocate	246	PREServe	253
CMS	246	PURge	253
CMSG	246	PUT	253
COMMAND	246	PUTD	253
COMpress	246	Query	254
COPy	246	QUIT	255
COUnt	246	READ	255
COVerlay	246	RECOVer	255
CP	246	REFRESH	255
CREplace	246	RENum	255
CURsor	247	REPEat	255
DElete	247	ReplAcE	255
Down	247	RESet	255
DUPLICat	247	RESTore	256
EMSG	247	RGTLLEFT	256
EXPand	247	Right	256
EXTRACT	248	SAVE	256
FILE	249	SCHANGE	256
Find	249	SET	256-264
FINDUp	249	SHift	265
Forward	249	SI	265
FUp	249	SORT	265
GET	249	SOS	265
Help	249	SPIit	265
HEXType	249	SPLTJOIN	266
Input	249		

STAck 266
STATus 266
TOP 266
TRAnsfer 267
Type 267
Up 267
UPPerCas 267
Xedit 267

XEDIT command return codes 42
XMITMSG command (CMS) 270



ZAP command (CMS) 271

**International Business
Machines Corporation
P.O. Box 6
Endicott, New York 13760**

**File No. S370/4300-36
Printed in U.S.A.**

SX20-4400-4



Quick Reference
Order No. SX20-4400-4

**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.

Fold and tape

Please Do Not Staple

Fold and tape

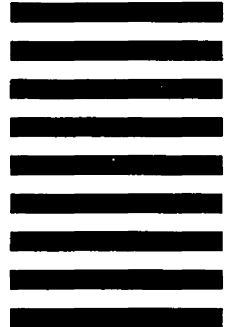


BUSINESS REPLY MAIL

FIRST-CLASS MAIL PERMIT NO. 40 ARMONK, NY

POSTAGE WILL BE PAID BY ADDRESSEE:

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES



INTERNATIONAL BUSINESS MACHINES CORPORATION
DEPARTMENT G60
PO BOX 6
ENDICOTT NY 13760-9987



Fold and tape

Please Do Not Staple

Fold and tape



International Business
Machines Corporation
P.O. Box 6
Endicott, New York 13760

File No. S370/4300-36
Printed in U.S.A.

SX20-4400-4



SX20-4400-04

