SX20-4400-1

Reference Summary

IBM Virtual Machine/ System Product Quick Guide for Users

IBM Programming Publications
Department G60, P.O. Box 6
Endicott, New York U.S.A. 13760

SX20-4400-1

IBM **Reference Summary**

IBM Virtual Machine/ System Product Quick Guide for Users

Second Edition (September 1982)

This edition, SX20-4400-1, applies to Release 2 of the IBM Virtual/Machine System Product, and to all subsequent releases unless otherwise indicated in new editions or Technical Newsletters. This major revision obsoletes SX20-4400-0. Changes or additions to the text and illustrations are indicated by a vertical line to the left of the change.

Changes are periodically made to the information herein; before using this publication in connection with the operation of IBM systems, consult the latest IBM System/370 and 4300 Processors Bibliography, Order No. GC20-0001, for the editions that are applicable and current.

Summary of Changes

For a list of changes see page iii.

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 program product in this publication is not intended to state or imply that only IBM's program product may be used. Any functionally equivalent program may be used instead.

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

A form for readers' comments is provided at the back of this publication. If the form has been removed, address comments to IBM Programming Publications, 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.

Summary of Changes
for SX20-4400-1
VM/SP Release 2

NEW:

This revision documents the changes that have been made to the VM/SP publications for Release 2.

The "Summary of VM/SP Commands and Service Aids" has been updated with new command formats and descriptions that document new CMS and CP Release 2 functions for the end-user.

The "Reserved Filetypes" section has been updated with several new reserved filetypes.

Descriptions of valid filenames, filetypes, and filemodes have been included.

CHANGED:

Minor technical and editorial changes have been made throughout this publication.

PREFACE

This publication provides a quick reference to VM/SP.

The section "Using VM/SP System" describes 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 are obtainable as a group under Order No. SB0F3820.

BINDER AVAILABLE

A handbook-sized binder, FE part number S229-4124, may be purchased from IBM. Customers may order it through their IBM marketing representative. IBM personnel should order it as an FE part from Mechanicsburg.

USING THIS PUBLICATION

This publication includes commands and operands that pertain to the VM/SP System Control Program (5664-67).

The following VM/SP publications should be used in conjunction with the first sections of this publication:

PREREQUISITE PUBLICATIONS

IBM Virtual Machine/System Product:

- CMS Command and Macro Reference, SC19-6209
- CMS User's Guide, SC19-6210
- CP Command Reference for General Users, SC19-6211
- Introduction, GC19-6200

COREQUISITE PUBLICATIONS

IBM Virtual Machine/System Product:

- CMS Primer, SC24-5236
- EXEC 2 Reference, SC24-5219
- Operator's Guide, SC19-6202
- Planning and System Generation Guide, SC19-6201
- System Messages and Codes, SC19-6204
- System Product Editor Command and Macro Reference, SC24-5221
- System Product Editor User's Guide,

| SC24-5220

System Programmer's Guide, SC19-6203

Terminal User's Guide, GC19-6206

CONTENTS

USING THE VM/SP SYSTEM 1
Getting Started 2
Terminal Operating Procedures 2
  Logging On 3
  Logging Off 5
Input Conventions 5
  Attention Handling 7

USING CMS 14
How To IPL CMS 14
  CMS System Product Editor 15
  EXEC 2 Interpreter 16
  System Product Editor (XEDIT) Features 16
Example of CMS Program Development Facilities 17
  Creating an Assembler Language Source File 17
  Assembling a Source File 20
  Creating a Load Module 22
  Erasing Unwanted Files 24
  Printing, Punching, and Reading Files 26

Reserved Filetype Descriptions 29
Return Codes 38
  CMS Return Codes 38
  CP DIRECT Command Return Codes 41
  CMS DDR Command Return Codes 42
  System Product Editor (XEDIT) Command Return Codes 43
  IPCS Commands Return Codes 44
  Example of a Return Code from a CP Command 45

SUMMARY OF VM/SP COMMANDS AND SERVICE AIDS 46
  Notational Conventions 46
  CP Privilege Classes 48
  CP, CMS, RSCS, and IPCS Commands . . . 51

FIGURES

Figure 1. Attention Handling in VM Mode (2 Parts)....................9

Figure 2. Attention Handling in CP Mode.......................11

Figure 3. Summary of Screen Status Action while Executing CMS and CP (2 Parts)............12

Figure 4. Sample Assembler Language Program Used for Creating a Source File (2 Parts).......18

Figure 5. Reserved Filetypes (8 Parts)....................30

Figure 6. Return Codes Produced by CMS (3 Parts)................38

Figure 7. Return Codes Produced by the CP DIRECT Command...........41

Figure 8. Return Codes Produced by the CMS DDR Command.........42

Figure 9. Return Codes Produced by the System Product Editor (XEDIT)Command (2 Parts)...43

Figure 10. Return Codes Produced by the IPCS Commands................44

Figure 11. Syntax Abbreviations Used..........................47

Figure 12. CP Privilege Class Descriptions (3 Parts)......48

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 which 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: a virtual operator's console, virtual storage, a virtual processor, and 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 physical machine in the computer room 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 sets of information called files.

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, and 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 _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 User's Guide_.

TERMINAL OPERATING PROCEDURES

For a more detailed description of the various terminal operating procedures, see the _VM/SP Terminal User's Guide_.

LOGGING ON

First, establish communication with VM/SP by turning on your terminal's power. The system sends a VM/370 ONLINE message or logo to the terminal.

Press the attention (ATTN) key, PA1 key (or equivalent), type logon, and identify yourself by entering your user identification (userid) as follows:

```
   l userid   (short for logon userid)
```

Then press the Enter (or Return) key.

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

```
   ENTER PASSWORD:
```

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

Note: In many systems you will not see your password as you type it. This is a security measure which prevents other people from learning your password and using your system without authorization. For more information on this masking technique or using the Print Inhibit feature, see the _VM/SP Terminal User's Guide_.

If you make a typing error you will not be able to see your mistake. VM/SP will give you the error message,

```
   DMKLOG050E PASSWORD INCORRECT REINITIATE
   LOGON PROCEDURE
```

You must start the logon procedure from the beginning by entering your userid again. If you do not do this, you are prompted by the message:

```
   RESTART
```

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

```
   DMKLOG054E ALREADY LOGGED ON LINE nnn
```

where nnn indicates the real line address of the terminal on which the user is logged. 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 log on with another userid (if another userid is reserved for your use) or try again later.

Once you have successfully logged on and entered the system, VM/SP replies with a LOGON AT message, such as:

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

A logon message from the VM/SP operator (if any) also displays at this time.

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= 00:11:43 VIRTCPU= 000:05.21
      TOTCPU= 000:21.03
   LOGOFF AT 11:34:44 EST THURSDAY 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. Only when the logoff procedure is completed should you turn off terminal power.

Note: If you logged on over a dialed line, you could 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 does not have to re-dial the line before logging on again.

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.

The logical line edit characters shown below are line edit default values. Your installation or your terminal may require other characters to fulfill the line edit functions.

Character Delete symbol (@): The character delete symbol deletes the preceding character in the input line. A string of "n" character delete symbols deletes the preceding "n" characters in the input line and itself. For example:

```
ABC@@ results in A
ABC#D@@@ results in AB
```

Line End symbol (#): The line end symbol indicates the end of a logical input line. Use of this character permits you to enter more than one logical input line in the same real input string. You can use this symbol to type several commands on the same line. CMS stacks the commands and executes them in sequence. For example:

```
down 1#type1#top
```

executes the commands as though entered:

```
down 1
type 1
top
```

Line Delete symbol (¢): The line delete symbol deletes all characters in the current logical line and itself. If you enter a line delete symbol erroneously, you can cancel its effect by entering a character delete symbol. For example:

```
ABC#DEF¢ results in ABC
ABC#¢ results in ABC
ABC¢ deletes the whole line
```

Logical Escape symbol ("): The logical escape symbol causes VM/SP to interpret the character following it as a data character (that is, to ignore it as a logical line editing character). This allows the system to interpret any of the line editing characters literally. For example, consider how you might enter the following line into the system:

```
   1 gross #2 pencils @ 92¢ per dozen
```

Under the VM/SP input conventions, you could not enter this line as shown, since it would be affected by the #, @, and ¢ line editing symbols. For example, the # symbol would end the line. However, VM/SP interprets the line correctly if entered as follows:

```
   1 gross "#2 pencils "@ 92"¢ per dozen
```

The logical escape symbol (") is not put into the file.

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.

ATTENTION HANDLING

Figures 1 and 2 indicate the various effects of an attention interruption for your virtual console. Figure 3 shows the screen control functions of the Enter and Cancel keys for display terminals.

Figure 1. Attention Handling in VM Mode (Part 1 of 2)

State	Press ATTN Key	Action
Terminal idle; keyboard entry blocked; virtual machine is running	1	Attention interruption pending; virtual machine is running
	>1	Keyboard activated for CP input
Terminal receiving output from virtual machine	1	Attention interruption pending; virtual machine running
	>1	Keyboard activated for CP input at completion of console I/O
Keyboard activated for input to virtual machine; no data entered or all data deleted	1	Device end (DE) and attention status pending; virtual machine is running[1]
	>1	Unit exception (UE) status pending; keyboard activated for CP input
Keyboard activated for input to virtual machine; some data entered	1	Unit exception (UE) status pending; virtual machine is running
	>1	Device end (DE) status pending; keyboard activated for CP input
Keyboard entry blocked; executing CP command	1 or >1	Attention ignored

[1] To perform this function on a 3270, position the cursor one position to the left of the user input area (bottom line, position 0) and press the ENTER key.

State	Press ATTN Key	Action
Keyboard entry blocked; in SLEEP mode entered via command	1 or >1	Keyboard activated for CP input
Keyboard entry blocked; in SLEEP mode entered via DIAGNOSE instruction	1 or >1	Virtual machine resumes execution
Terminal receiving output from CP but not from user command	1	Attention interruption pending; virtual machine running
	>1	Keyboard activated for CP input activated for CP input
Terminal receiving output in response to CP command	1 or >1	Output line cancelled and, in some cases, command output is cancelled
Keyboard activated for CP input; no data entered or all data cancelled	1 or >1	Attention interruption made pending; virtual machine is running
Keyboard activated for CP input; some data entered	1 or >1	Input line cancelled; keyboard activated for CP input

Figure 1. Attention Handling in VM Mode (Part 2 of 2)

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

Figure 2. Attention Handling in CP Mode

Figure 3. Summary of Screen Status Action while Executing CMS and CP (Part 1 of 2)

Initial Status	Mode	Key Pressed	Data	Action	Resulting Status
RUNNING	CP	ENTER	NONE	Enters console function mode	CPREAD
			DATA	Executes console function	RUNNING
		CNCL	N/A	Clears output area	RUNNING
	VM	ENTER	NONE	"Attn" interruption pending, VM run	RUNNING[1]
			DATA	"Attn" interruption pending, stack data, VM run	RUNNING[2]
		CNCL	N/A	Clears output area	RUNNING
		PA2(APL on)	N/A	Clears output area	RUNNING
MORE	CP/VM	ENTER	NONE	Holds screen output	HOLDING
			DATA	"Attn" interruption pending, stack data	MORE
		CNCL	N/A	Clears output area, continues output	RUNNING
HOLDING	CP/VM	ENTER	NONE	Allows screen output to continue	MORE
			DATA	"Attn" interruption pending, stack data	HOLDING
		CNCL	N/A	Clears output area, continues output	RUNNING[3]

CPREAD	CP/VM	ENTER	NONE	"Null" line return	RUNNING[4]
			DATA	Data return for function	RUNNING
		CNCL	N/A	Clears output area	CPREAD
VMREAD	CP/VM	ENTER	NONE	"Null" line return, VM run	RUNNING
			DATA	Data return for read, VM run	RUNNING
		CNCL	N/A	Clears output area	VMREAD
		PA2 (if APL on)	N/A	Clears output area. Present external interruption to virtual machine.	RUNNING
NOT ACCEPTED	A previously stacked input buffer is still pending				Returns to former status

[1] The status shown is RUNNING; however, the virtual machine should respond to the Attn with a read, whereupon the status goes to VMREAD.

[2] If a data buffer is already stacked for a virtual machine, the terminal displays NOT ACCEPTED status before returning to the RUNNING status.

[3] If you are running with TERMINAL MODE CP (the default for the primary system operator) then an attention return is also made, causing cancellation of the function. Operators at the 158 and 3066 display console as well as the 3036 display console for the 3031, 3032, and 3033 processors use this function to allow termination of certain QUERY or DISPLAY functions because consoles do not have a PA1 key.

[4] Unless you are the VM/370 primary system operator or are using the SET RUN ON option, the status returns to CP READ for another console function if the previous read was for a console function.

Figure 3. Summary of Screen Status Action while Executing CMS and CP (Part 2 of 2)

USING CMS

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

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

- Log on 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 the "Terminal Operating Procedures" section of this book. The IPL procedure is described in this section. Disk formatting is detailed in VM/SP CMS User's Guide.

HOW TO 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:

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

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

CMS SYSTEM PRODUCT EDITOR

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

- Full screen editing
- Multiple views of the same file
- Multiple files viewed on the same screen
- Selective column viewing
- Automatic wraparound of lines that are longer than the screen
- Ability to issue selected commands directly from the displayed line
- Ability to define screen formats as preferred

For a thorough understanding of the System Product editor (XEDIT) 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 for 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 EDITOR (XEDIT) FEATURES

The system product editor (XEDIT) provides for expansion and flexibility in language with the following:

- Extended string search to facilitate text processing
- Automatic update generation functions
- Packed file handling functions
- CMS HELP Facility support
- Column pointer for intraline editing

- Line splitting and/or joining functions

- Import and/or export of data facilities

- Macros for use in migrating from EDIT or DES (display editing system)

- Macro writing capabilities to expand the basic subcommand language and to tailor the language to your own application.

EXAMPLE OF CMS PROGRAM DEVELOPMENT FACILITIES

This section illustrates several CMS functions that are useful in creating and manipulating CMS files.

CREATING AN ASSEMBLER LANGUAGE SOURCE FILE

The program shown in Figure 4 in this section 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
NEW FILE:
XEDIT:
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  *
```

Figure 4. Sample Assembler Language Program Used for Creating a Source File (Part 1 of 2)

```
          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',edit=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
buffl     ds    cl80
          end   manip
(Press the RETURN key to leave Input mode.)

XEDIT:
file
R;
```

Figure 4. Sample Assembler Language Program Used for Creating a Source File (Part 2 of 2)

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 in order 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 cmslib osmacro
```

The Load Address (LA) instruction following EOF (end-of-file) is inserted only for testing; it is deleted after the function is tested.

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
 IEU024 NEAR OPERAND COLUMN 1-UNDEF SYMBOL

     1 STATEMENT FLAGGED IN THIS ASSEMBLY
     8 WAS HIGHEST SEVERITY CODE
R(00008);
```

Each asterisk (*) on the second line indicates two seconds of virtual processor time. The message IEU024 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.

First, 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.

The MANIP TEXT file contains the object module.

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
R;
```

Now that the error has been corrected,

you can assemble the file again:

```
assemble manip
*
*
*
 ASSEMBLER (F) DONE

 NO STATEMENTS FLAGGED IN THIS ASSEMBLY
R;
```

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
R;

genmod manip
R;
```

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.
R(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
R;
```

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

```
assemble manip
*
*
*
 ASSEMBLER (F) DONE

 NO STATEMENTS FLAGGED IN THIS ASSEMBLY
R;

load manip
R;

genmod manip
R;
```

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
R(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.
R(00001);
```

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

```
manip manip assemble a1 manip1 assemble a1
R;
```

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

R;

erase manip1 assemble
R;
```

Or type DISCARD in the "Cmd" space next to the file you no longer want from the FILELIST screen. Then press PF10 (not 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
R;
```

PRINTING, PUNCHING, AND READING FILES

Printing

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        COPY 01
R;
```

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

```
print manip listing
R;
```

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
R;

erase manip text
R;
```

Punching

If other users want to use your MANIP program, send it to them by changing the destination of your virtual punch, then punch the MANIP TEXT file. Use the CMS COPYFILE or MOVEFILE commands to transfer the MANIP MODULE file to another user. For example, suppose the user PAYROLL wanted to use the MANIP program. You could send PAYROLL a copy of the TEXT file by entering:

```
spool 00d to payroll
R;
```

```
punch manip text
PUN FILE 029 TO  PAYROLL
R;
```

Or simply use the SENDFILE EXEC procedure.

Reading

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  TEXT 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:

```
read *
:READ  MANIP TEXT A1 PUBS mm/dd/yy 13:29:03
R;
```

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

```
purge reader      (or purge reader 029)
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 the new VM/SP user 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.

RESERVED FILETYPE DESCRIPTIONS

Figure 5 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 documentation.

A fileid is a CMS file identifier that consists of a filename, filetype, and 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 alphameric 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 alphameric 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 5). 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.

Figure 5. Reserved Filetypes (Part 1 of 8)

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 F	80 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	U	≤3440	User input and output files
BASIC	BASIC	Input	fn	F	≤256	BASIC language source statements
CMSUT1	READCARD COPYFILE DISK LOAD	Intermediate work	READCARD COPYFILE DISK	F	80	

				Format		
Filetype	Command	Usage	Filename	RECFM	LRECL	Contents
	TAPE LOAD	files	TAPE			
	UPDATE		fn			
	INCLUDE		DMSLDR			
	LOAD		DMSLDR			
	MACLIB		DMSLBM			
	EDIT		EDIT			
CNTRL	UPDATE	Input	fn	F	80	Control file update
	XEDIT					
COBOL	COBOL	Input	fn	F	80	COBOL source statements
COPY	MACLIB	Input	fn	F	80	COPY control cards and macro definitions
	SSERV	Output	fn			A book from a DOS/VS source library.
DIRECT	DIRECT	Input	fn	F	80	User Directory entries
DOSLIB	DOSLIB	Input	fn	V	1024	CMS/DOS phase library
	DOSLKED	Output	fn			
	FETCH	Input	fn			
	GLOBAL	Input	fn			
DOSLNK	DOSLKED	Input	fn	F	80	Linkage editor control statements for input to CMS/DOS linkage editor

Figure 5. Reserved Filetypes (Part 2 of 8)

Filetype	Command	Usage	Filename	Format		Contents
				RECFM	LRECL	
ESERV	ESERV	Input	fn	F	80	Input control statements for ESERV program
EXEC	EXEC LISTFILE GEN3705 LISTIO	Input Output Output Output	fn CMS fn $LISTIO	V	130	EXEC statements
FREEFORT	GOFORT	Input	fn	V	≤81	FREEFORM FORTRAN source statements
FORTRAN	FORTGI FORTHX GOFORT TESTFORT	Input	fn	V	80	FORTRAN source statements
FTnnF001	FORTRAN execution	Input/ Output	fn			User input and output files
GLOBALV	GLOBALV DEFAULTS	Input/ Output	Initial Session Lasting	F/V V V	≤5201 ≤520 ≤520	Collection of named variables
LISTING	ASSEMBLE ASM3705 ESERV GOFORT FORTGI FORTHX COBOL	Output Output Output Output	fn fn fn fn	F	121	Processor printed output

Figure 5. Reserved Filetypes (Part 3 of 8)

Filetype	Command	Usage	Filename	Format RECFM	 LRECL	Contents
	PLIC					
	PLICR					
	PLIOPT	Output	fn			
	TESTCOB	Input	fn	F	121	COBOL processor output used as input to SOURCE subcommand of TESTCOB
LKEDIT	LKED	Output	fn	F	121	Listing
LOADLIB	LKED	Output	fn	F	≤260	3704/3705 control program load modules
	ZAP	Input	fn			
MACLIB	GLOBAL	Library	fn			Macro definitions (dictionary and members)
	MACLIB	Input/ Output	fn			
MACRO	ESERV	Output	fn	F	80	Macro definitions
	MACLIB	Input	fn			
MAP	DOSLIB	Output	libname	F	80	Library map
	DOSLKED	Output	fn			DOS/VS linkage editor map
	DSERV	Output	DSERV			Directory information from DOS/VS private or system libraries.
	INCLUDE	Output	LOAD			Module map

Figure 5. Reserved Filetypes (Part 4 of 8)

Filetype	Command	Usage	Filename	Format RECFM	 LRECL	Contents
	LOAD	Output	LOAD			Module map
	MACLIB	Output	fn			Library map
	TXTLIB	Output	fn			Library map
	TAPE	Output	fn			Library map
MEMO				F	80	
MODULE	GENMOD	Output	fn	V		Nonrelocatable object file
	LOADMOD	Input	fn			
	MODMAP	Input	fn			
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
NOTEBOOK	RECEIVE SENDFILE	Input	userid	V	132	Notes sent to or received by you
PLI or PLIOPT	PLIOPT	Input	fn	F		PL/I source statements
	PLIC	Input	fn			
	PLICR	Input	fn			
PROC	PSERV	Output	fn	F	80	A procedure from the DOS/VS procedure library
SCRIPT	SCRIPT[1]	Input	fn	V	132	Input to SCRIPT processor

[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, Order No. SH20-9161.

Figure 5. Reserved Filetypes (Part 5 of 8)

				Format		
Filetype	Command	Usage	Filename	RECFM	LRECL	Contents
SYMDMP	FCOBOL	Output	fn	V	512	DOS/VS COBOL DEBUG file for SYMDMP option
SYNONYM	SYNONYM	Reference	fn	F	80	Command name synonyms
SYSUT1 SYSUT2 SYSUT3	ASM3705 ASSEMBLE COBOL LKED PLIOPT	Work Work Work Work Work	fn fn fn fn fn			
SYSUT4	COBOL LKED PLIC PLICR	Work	fn	F	80	
	TESTCOB	Input			512	Used as input to TESTCOB
TESTFORT	TESTFORT	Output	fn	VB	125	Processor printed output
TEXT	ASSEMBLE	Output	fn	F	80	Object code

Figure 5. Reserved Filetypes (Part 6 of 8)

				Format		
Filetype	Command	Usage	Filename	RECFM	LRECL	Contents
	ASM3705	Output	fn			3704/3704 source code and job control language statements
	COBOL	Output	fn			Object code
	DOSLKED	Input	fn			Object code
	GEN3705	Output	fn(Ln)			Linkage editor control statements for 3704/3704 control programs
	INCLUDE	Input	fn			Object code
	LKED	Input	fn			Object code and LKED control cards
	LOAD	Input	fn			Object code
	PLIOPT	Output	fn			Object code
	TXTLIB	Input	fn			Object code
	GOFORT	Output	fn			Object file
	FORTGI	Output				
	FORTHX	Output				
	RSERV	Output	fn			
	TEXTFORT	Input				
TXTLIB	GLOBAL	Library	fn			Object decks
	TXTLIB	Output	fn			(dictionary and members)

Figure 5. Reserved Filetypes (Part 7 of 8)

Figure 5. Reserved Filetypes (Part 8 of 8)

Filetype	Command	Usage	Filename	Format		Contents
				RECFM	LRECL	
UPDATE	UPDATE	Input	fn	F	80	UPDATE control cards
UPDLOG	UPDATE	Output	fn	F		UPDATE log
VSBASIC	VSBASIC	Input	fn	F	≤256	VSBASIC language source statements
VSBDATA	VSBDATA	Execution time files	fn	V	≤140	VSBASIC user input/ output files
UPDTxxxx	UPDATE	Input	fn	F	80	Contains UPDATE control statements
ZAP	ZAP	Input	fn	F	80	Contains control records that modify or dump files

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 to register 15.

If no Warning, Error, Severe Error, or Terminal Error messages are generated during execution of the command, the return code passed to register 15 is zero.

Commands that invoke Program Products pass the return code set by the program in register 15 to the user. 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

Figure 6 shows the return codes passed by CMS commands.

RC	Meaning
-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).

Figure 6. Return Codes Produced by CMS (Part 1 of 3)

RC	Meaning
-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).
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.
20	An invalid character is in the fileid. Valid characters are: 0-9, A-Z, $, @, #, +, -(hyphen), :(colon), _(underscore).
24	The user did not specify the command line correctly.
28	An error occurred while trying to access, or manipulate, a user's files; for example, file not found.

Figure 6. Return Codes Produced by CMS (Part 2 of 3)

RC	Meaning
32	-The user's file is not in the expected format, or -The user's file does not contain the expected information.
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	-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	Input/output device errors.
104	A functional error for which the system is responsible occurred during execution of the command.
256	All unexpected errors for which the system is responsible; that is, Terminal Error messages.

Figure 6. Return Codes Produced by CMS (Part 3 of 3)

CP DIRECT COMMAND RETURN CODES

Figure 7 shows the return codes passed by the CP DIRECT command.

RC	Meaning
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.

[1] xx is the CMS routine return code.

Figure 7. Return Codes Produced by the CP DIRECT Command

CMS DDR COMMAND RETURN CODES

Figure 8 shows the return codes passed by the CMS DDR command.

RC	Meaning
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.

[1] xx is the CMS routine return code.

Figure 8. Return Codes Produced by the CMS DDR Command

SYSTEM PRODUCT EDITOR (XEDIT) COMMAND RETURN CODES

Figure 9 shows the return codes passed by the XEDIT command.

RC	Meaning
nn	Command's RC specified as operand
-1	LOAD command has already been issued
0	Normal
1	TOP or EOF reached during execution
2	Target line not found
3	Terminal is not a display terminal
4	Insufficient storage
5	Invalid operand
6	Subcommand rejected in the profile because of LOAD error
7	Error when building the update file
8	Prefix area contains subcommand that is not yet executed
12	Disk is read-only

Figure 9. Return Codes Produced by the System Product Editor (XEDIT) Command (Part 1 of 2)

RC	Meaning
13	Disk is full
20	Invalid character in filename or filetype
24	Invalid filemode
28	Source file not found (UPDATE MODE) or specified PROFILE macro does not exist, or file XEDTEMP CMSUT1 already exists
32	Error during updating process
36	Disk has not been accessed as yet
88	File is too large; cannot fit into storage
100	Error occurred while reading the file into storage

Figure 9. Return Codes Produced by the System Product Editor (XEDIT) Command (Part 2 of 2)

IPCS COMMANDS RETURN CODES

Figure 10 shows the return codes passed by the IPCS commands.

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

Figure 10. Return Codes Produced by the IPCS Commands

EXAMPLE OF A RETURN CODE FROM A CP COMMAND

Commands or functions 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

R;   which indicates successful execution.

-- or --

R(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 and Macro Reference or EXEC 2 Reference for a description of the &RETCODE special variable.

SUMMARY OF VM/SP COMMANDS AND SERVICE AIDS

The following list contains CP, CMS, IPCS, and RSCS commands and the VM/SP service aids; a brief description precedes a syntactic representation of each command.

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 Figure 11.

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 11. Syntax Abbreviations Used

For a detailed and thorough explanation of command formats and notational conventions, see the _VM/SP CMS Command and Macro Reference_.

CP PRIVILEGE CLASSES

VM/SP users are divided into seven classes (A through G) according to the functions they must perform. Each user class may use a specific subset of CP commands. In addition, there are commands (class Any) available to all users.

Figure 12 summarizes the CP privilege classes and the functions performed by each class of user. Figure 12 also indicates the publications in which the commands for each class are detailed.

Class	User and Function
A[1]	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.

[1] Described in VM/SP Operator's Guide.

Figure 12. CP Privilege Class Descriptions (Part 1 of 3)

Class	User and Function
	Note: The class A system operator who is automatically logged on during CP initialization is designated as the primary.system operator.
B[1]	System Resource Operator: The class B user controls allocation and deallocation of all the real resources of the VM/SP system, except those controlled by the primary system operator and spooling operator.
C[1]	System Programmer: The class C user updates certain functions of the VM/SP system. The system programmer can modify real storage in the real machine.
D[1]	Spooling Operator: The class D user controls spool data files and specific functions of the system's unit record equipment.
E[1]	System Analyst: The class E user displays the contents of real storage, performs the functions required to generate saved systems and discontiguous saved segments, and controls the collection and recording of performance measurement data. This class of user can display the contents of specified real storage areas on the virtual operator's console or on a spooled virtual printer, but cannot modify real storage.

[1] Described in VM/SP Operator's Guide.

Figure 12. CP Privilege Class Descriptions (Part 2 of 3)

Class	User and Function
F[1]	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 repressable machine check errors to be initiated or resumed.
G[2]	General User: The class G user controls functions associated with the execution of his virtual machine. A general user cannot display or modify real storage.
Any[2]	The Any classification is given to certain CP commands that are available to any user. These are primarily for the purpose of gaining and relinquishing access to the VM/SP system.
H	Reserved for IBM use.

[1] Described in VM/SP Operator's Guide.

[2] Described in VM/SP CP Command Reference for General Users.

Figure 12. CP Privilege Class Descriptions (Part 3 of 3)

CP, CMS, RSCS, AND IPCS COMMANDS

The following table details command formats and descriptions. Also included are macro formats and service aids.

Description	Format
* CP Class Any Permits comments	* anycomment
* CMS Permits comments	* anycomment
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]
ACNT CP Class A Creates accounting records.	ACNT { userid1 [userid2...] } ALL [CLOSE] CLOSE1
ADSTOP CP Class G Halts the virtual machine's execution.	ADSTOP { hexloc|OFF }
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}]

```
|ASM3705                         CMS |ASM3705 fn [(options...[)]]
|  Invokes the 3705 assembler.       |
|  Note: All of the options of the   | options:
|        3705 XF Assembler are       |[LIST|NOLIST]
|        supported and may be used   |[LINECOUN 55|LINECOUN nn]
|        with the ASM3705 command,   |[DISK|PRINT|NOPRINT]
|        with the exception of       |[XREF |NOXREF]
|        ALIGN|NOALIGN and TEST|     |[DECK|NODECK]
|        NOTEST.                     |[RENT|NORENT]
|                                    |[LOAD|NOLOAD]
|                                    |
|ASMGEND                         CMS |ASMGEND
|  Builds the system assembler and   |
|  creates the associated auxiliary  |
|  directory.  (System programmers   |
|  only)                             |
|                                    |
|ASSEMBLE                        CMS |Assemble  fn  [(options...[)]]
|  Invokes the system assembler.     |
|                                    |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]
```

<table>
<tr><th>Description</th><th>Format</th></tr>
<tr><td>ASSEMBLE (cont.)</td><td>output control options:
[DECK|NODECK] [OBJECT|NOOBJECT]
[TEST|NOTEST]

SYSTERM 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]</td></tr>
<tr><td>ASSGN CMS
Assigns or unassigns a system or programmer logical unit for a virtual I/O device.</td><td>ASSGN SYSxxx { Reader
PUnch
PRinter
Terminal
TAP [n|1]
mode
IGN
UA } [(options...[)]]

options:
[UPCASE|LOWCASE] [7TRACK|9TRACK] [TRTCH a] [DEN den]</td></tr>
</table>

ATTACH	CP Class B	Attaches a real device to a specified user or to the system.	ATTach raddr [To] {userid [As] vaddr [R|/O]}
			{SYSTEM [As] volid}
			{raddr...}
			{raddr-raddr} [TO] userid [R[/O]] [3330V]

ATTACH CHANNEL	CP Class B	Attaches a channel to a designated user.	ATTach CHANnel c [PROC nn] [To] [userid|*]
ATTN	CP Class G	Makes attention interruption pending.	ATTN
AUTOLOG	CP Class A,B	Logs on any virtual machine that is defined in the VM/370 directory.	AUTOLOG userid password [variable data]
BACKSPAC	CP Class D	Restarts or repositions a current spool file.	Printer Format: BAckspac raddr [File pages [EOF] 1] Punch Format: BAckspac raddr [File]

Description	Format
BACKSPAC RSCS Restarts or repositions spool file processing.	BAckspac [linkid][FILE|nnn]
BEGIN CP Class G Starts the execution of a virtual machine.	Begin [hexloc]
CHANGE CP Class D Alters the attributes of a closed spool file.	CHange [userid|SYSTEM|*] {Reader|Printer|PUnch} {CLass c1|spoolid|FORM form1|ALL} {CLass c2 COpy[*]nnn [HOLD|NOHOLD] [SYS|NOSYS] DIst distcode FLash name nnn MOdify name [n] CHars name1 [name2[-3[-4]]] CHars name1[CHars name2[-3[-4]]] FCB name FORM form2}[1] [NAme {fn[ft]|dsname}]

CHANGE CP Class G Alters the attributes of a closed spool file.	CHange Reader | Printer | Punch CLass c1 | spoolid | FORM form1 | ALL CLass c2 }[2] COpy[*]nnn [HOld | NOHold] DIst dist COpy [*]nnn FLash name nnn MOdify name [n] CHars name1 [name2 [name3 [name4]]] CHars name1 [CHars name2[CHars name3[CHars name4]]] FCB name FORM form2 [NAME {fn[ft] | dsname}]
CHANGE RSCS Alters the attributes of a closed spool file.	CHange [linkid] spoolid PRIority nn CLass c COpy nn HOld|NOHOLD DIst distcode 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] 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.

Description	Format
CLOSE CP Class G Terminates spooling operations on a virtual reader, printer, or punch.	Close Reader vaddr HOld NOHold CONsole PUrge Printer FORM form PUnch vaddr HOld [DIst distcode] NAme fn[ft] NOHold dsname
CMD Executes selected CP or CMS commands in the Programmable Operator's virtual machine.	CMD vmcmd
CMD RSCS Controls the logging of I/O activity on a specified RSCS link or passes control information to remote batch processor. Available to RSCS operator only.	CMD linkid {text|LOG|NOLOG}
CMDCALL CMS Converts EXEC2 extended plist function calls to CMS extended 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 A-disk.	CMSGEND fn CTLCMS mode 'x' CTLALL A NOCLEAR MAP NOINV
COMPARE CMS Compares two existing files on a record for record basis and displays dissimilar records.	COMpare fileid1 fileid2 [(option[)]] options: COL mm[-]nn 1 lrecl
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 fileidi1 [fileidi2...] [fileido] [(options...[)]] options: Type NEWDate NEWFile PRompt NOType OLDDate REPlace NOPRompt

Description	Format
COPYFILE (cont.)	copy extent options: FRom recno FOR numrec SPecs OVly FRLabel xxxxxxxx TOLabel xxxxxxxx NOSPecs APpend data modification options: RECfm {F} LRecl nnnnn TRUnc PAck FIll c {V} NOTRunc UNPack Fill hh FIll 40 character translation options: EBcdic UPcase TRAns SIngle LOWcase
COUPLE CP Class G Connects virtual channel to channel adapters.	COUPLE vaddr1 [TO] userid vaddr2
CP CP Class Any Permits execution of CP commands within your privilege class.	CP [commandline1 [#commandline2 #...]]

__CP__ CMS

Permits entry of CP commands without leaving the CMS environment.

```
CP [commandline]
```

__CPEREP__ CMS

Invokes CPEREP service aid to process VM/SP error recordings. Requires CP Class C, E, or F.

```
CPEREP  [filename filetype [filemode|*]]

         ┌              ┐   ┌      ┐1
         │{[ACC[=Y]]}│   │CLEAR │
         │{ACC=N     }│   │CLEARF│
         └              ┘   └      ┘

         ┌                         ┐ ┌                     ┐     ┐
         │                {cua    } │ │           {cua    } │     │
         │CONTROLLER=(cpuser.{cuX    } │ │,cpuser.{cuX    } │...)│
         │                {cua-cua} │ │           {cua-cua} │     │
         └                         ┘ └                     ┘     ┘

         [CPU=serialno.modelno[,serialno.modelno,...]]
         [CPUCUA=(serial.addr,serial.addr[,serial.addr,...])]

         ┌                                                   ┐2
         │CTLCRD {date1[date2[interval[title...]]]}│
         │       {date1,[date2],[interval],[title]}│
         └                                                   ┘
```

__Notes__

1. The operands for the CPEREP command cannot be entered on the command line.
2. Enter operands via prompting technique or from file specified.

[1] Operand exclusive to VM/SP.

[2] After entering the CTLCRD operand and associated data, no other operands can be entered on the same command line; the next operand must begin on a new line.

Description	Format
CPEREP (cont.) 3. If the TYPE= operand has no record type specified, then all record types are processed. 4. Only the privilege class F user (usually system support personnel or the IBM customer engineer (CE)) can erase the VM/SP error recording area.	[CUA=(addr[,addr,...])] [DATE=,yrday[,yrday])] [DEV=(devtype[,devtype,...])] [DEVSER=(serial[,serial,...])] [ERRORID=(seqno[cpuid,asid,hh,mm,ss,t])] [{Event[=Y] | [EVENT=N]}] [{HIST[=Y] | [HIST=N]}] [LINECT={nnn | 50}] [{MERGE[=Y] | [MERGE=N]}] [LIBADR=addr] [MOD=(modelno[,modelno...])] [PRINT={PS | PT | SU | NO}] [{RDESUM[=Y] | [RDESUM=N]}] [{SHORT[=Y] | [SHORT=N]}]

CPEREP (cont.)

```
[              (cua    )  [           (cua    )]    ]
[SHARE=(cpuser.{cuX    }  [,cpuser.   {cuX    }]...)]
[              (cua-cua)  [           (cua-cua)]    ]

[SYMCDE=(nnnn)]   [(SYSUM[=Y])]  [TABSE=sizeK]
[       {nnnx}]   [([SYSUM=N])]
[       {nnxx}]
[       (nxxx)]

 [(TERMINAL[=Y])]1 [TERMN=termname]
 [([TERMINAL=N])]

[THRESHOLD=(tempread,tempwrite)]  [(TRENDS[=Y])]
[TIME=(hhmm,hhmm)]                [([TRENDS=N])]

[TYPE=[C][D][E][H][I][M][O][S][T]]

[VOLID=(volid1[,volid2[,volid3[,volid4]]])]

[(ZERO[=Y])]
[([ZERO=N])]
```

[1] Operand exclusive to VM/SP.

Description	Format
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 CMS module TRAPRED	CPTrap { typenum [Vmblok nnnnnn DEVaddr nnnn COde nnnn OFF] .. STArt [[TO] userid *] ALL [ON] OFF ALLOWid userid CLOSE STOP }

```
DCP                       CP Class C,E   DCP   ( ┌MLhexloc1┐ ┌ ┌-┐ ┌hexloc2┐  ┐ )
   Displays real processor storage             ( │NLhexloc1│ │ └:┘ │  END  │  │ )
   on the terminal.                            ( │MThexloc1│ │     └       ┘  │ )
                                               ( │NThexloc1│ │                │ )
                                               ( │ Mhexloc1│ │     ┌          │ )
                                               { │ Nhexloc1│ │ {.} │bytecount││ }
                                               ( │ Lhexloc1│ │     │   END   ││ )
                                               ( │ Thexloc1│ │     └         ┘│ )
                                               ( │  hexloc1│ └                ┘ )
                                               ( │    0    │                    )
                                               ( └         ┘                    )

DDR                              CMS     DDR  [fn ft [fm|*]]
   Dumps, restores, prints, or
   copies data from DASD devices         I/O definition statements:
   and tape devices.
                                         {INput  } cuu type ┌volser┐ [(options...[)]]
                                         {OUTput }          │altape│
                                                            └      ┘
```

<table>
<tr><th>Description</th><th>Format</th></tr>
<tr><td>DDR (cont.)</td><td>options:
[REWind | LEave | UNload] [MOde 6250 | MOde 1600 | MOde 800] [SKip nn | SKip 0]

SYSPRINT control statement:
SYsprint [cuu|CONS]

Function control statements:
{DUmp | COpy | REstore} [FTR][1] [cyl1[TO][cyl2[Reorder][TO][cyl3]] | block1[TO][block2[Reorder][TO][block3]] | CPvol | NUcleus | ALL]

PRINT/TYPE function statements:
{PRint | TYpe} [cyl1 [hh1 [rr1]][To cyl2 [hh2[rr2]]][(options...[)] | block1[To block2]]

options:
[Hex] [Graphic] [Count]</td></tr>
</table>

DEBUG CMS	DEBUG
Enters the DEBUG environment to test and debug a program.	The format of each DEBUG subcommand follows:
Stops program execution at a specific instruction location.	BReak id [symbol hexloc]
Displays the Channel Address Word (CAW).	CAW
Displays the Channel Status Word (CSW).	CSW
Assigns a symbolic name to a specific storage address.	DEFine symbol hexloc [bytecount 4]
Dumps the contents of storage locations to the virtual printer.	DUmp [[symbol1 hexloc1 0] [symbol2 hexloc2 * 32] [ident]]

[1] The FTR option is valid only with the DUMP control statement.

Description	Format
DEBUG (cont.)	
Exits from the DEBUG environment and begins program execution.	GO [symbol hexloc]
Displays the contents of the specified general registers.	GPR reg1 [reg2]
Returns to CMS environment.	HX
Sets a base address.	ORigin { symbol|hexloc|0}
Displays old PSW.	PSW
Returns to CMS environment when DEBUG environment was entered via DEBUG command.	RETurn

Changes the contents of the specified register control word.	SET { CAW hexinfo
	CSW hexinfo [hexinfo]
	PSW hexinfo [hexinfo]
	GPR reg hexinfo [hexinfo] }

Stores up to 12 bytes of hexadecimal information in the specified virtual location.	STore { symbol hexloc } [hexinfo [hexinfo]]
Examines virtual storage locations.	X { symbol [n|length] hexloc [n|4] }

DEFAULTS CMS Set or display default options for the commands: FILELIST, NOTE RDRLIST, RECEIVE, PEEK and SENDFILE	DEFAULTS [Set command options...] [LIST [command]]
DEFINE CP Class 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 Printer PUnch [As]vaddr CONsole CTCa TIMer 1403 1443 2501 2540P 2540R 3203 3211 3262 3289E 3505 3525 3800 [As] vaddr [Size ww ll] [4Wcgm] [CFS] [NODatck] [2Wcgm] [BTS] [DAtch] CHANels [As] {SEL} {BMX})

Description	Format
DEFINE (cont.)	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 | DEFine   linkid {Class c          }[1]
  Temporarily adds a link and its    |                 {KEEP holdslot    }
  attributes to the existing link    |                 {LINE vaddr       }
  table.  Available to RSCS          |                 {TASK name        }
  operator only.                     |                 {TYPE driverid    }

DELETE                          RSCS | DELete   linkid
  Temporarily undefines a            |
  previously defined RSCS link.      |
  Available to RSCS operator         |
  only.                              |

DESBUF                           CMS | DESBUF
  Clears the console and program     |
  stack input and output buffers.    |

DETACH                    CP Class B | DETach   {raddr      } [From] {userid}
  Removes a real device from the     |          {raddr...   }        {SYSTEM}
  system or from a specific user.    |          {raddr-raddr}        {  *   }

DETACH                    CP Class G | DETach   {[vaddr [vaddr...]]}
  Removes a virtual device from the  |          {[vaddr-vaddr      ]}
  the virtual machine.               |
```

[1] One of these options must be specified, however, more than one may be entered.

Description	Format
DETACH CHANNEL CP Class B Removes the specified channel and all its related devices from the specified user.	DETach CHANnel c [PROC nn] [From] {userid|*}
DETACH CHANNEL CP Class G Detaches the specified channel from your virtual machine.	DETach CHANnel c
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] Console cuu devtype [class] Dedicate cuu {rdev|[VOLID] volser}[R/O][3330V] DIRectory cuu devtype volser Ipl iplsys Link userid ldev [cuu [mode]] Mdisk cuu devtype {cylr cyls volser [mode[pr[pw[pm]]]] T-DISK cyls}

DIRECT (cont.)	Option Realtimer Ecmode Virt=Real Acct Svcoff BMX CPUID bbbbbb AFFinity nn Isam VMsave STFirst 370E Maxconn nnnn SPEcial cuu devtype [IBM|Tele] Spool cuu devtype [class] User userid pass [stor [mstor [cl [pri [le [ld [cd [es]]]]]]]] ON ON ON ON OFF OFF OFF OFF
DISABLE CP Classes A,B Prevents communication lines from accessing VM/SP.	DISAble {raddr... SNA[userid] ALL}
DISCONN CP Class Any Disconnects the terminal from VM/SP system while virtual machine continues operation.	DISConn [HOld]
DISCONN RSCS Disconnects the RSCS operator's terminal from VM/SP while the RSCS virtual machine continues operation.	DISConn [userid]

Description	Format

DISK CMS
Dumps disk files to punched cards and restores disk files.
	DISK {DUMP fn ft [fm] }
	{LOAD [(OLDate[)]]}

DISPLAY CP Class G Displays storage locations (first-level only, registers, program status word, channel address word, and channel status word.	Display ([hexloc1] [[{-} [hexloc2]]] [Khexloc1] [[{:} [END]]] [Lhexloc1] [ ] [Thexloc1] [[{.} [bytecount]]] [[[END]]] Greg1 [{-} [reg2]] Yreg1 [{:} [END]] Xreg1 [ ] [{.} [regcount]] [[END]] Psw CAW CSW)

<u>DLBL</u> 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 used 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 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 ]

                          ┌CHANGE  ┐      ┌EXTENT┐
                          │NOCHANGE│      │MULT  │
                          └        ┘      └      ┘
                                          [CAT catdd]
                                          [BUFSP nnnnnn]
```

(Defaults underlined: CMS FILE ddname, CHANGE.)

Description	Format
<u>DMCP</u> CP Class C, E Dumps any area of System/370 real storage to a spool device.	DMCP ([MLhexloc1 | NLhexloc1 | MThexloc1 | NThexloc1 | Mhexloc1 | Nhexloc1 | Lhexloc1 | Thexloc1 | hexloc1 | <u>0</u>] [{:|-} [hexloc2 | <u>END</u>] ; {.} [bytecount | <u>END</u>]]) [*dumpid]
<u>DOSLIB</u> 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...[)]] } <u>options</u>: [TERM | <u>DISK</u> | PRINT]

DOSLKED CMS

In CMS/DOS, link-edits TEXT files from CMS disks, or object modules from DOS/VS 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...  │
       │ALL       │
       └          ┘
```

DRAIN RSCS

Deactivates the specified link after the current file processing is completed.

```
DRain [linkid]
```

DROPBUF CMS

Eliminates the most recently created program stack buffer and all buffers created after it.

```
DROPBUF n
```

Description	Format
DSERV CMS Obtains information contained in DOS/VS private or system libraries.	DSERV { CD [PHASE {name [nn 12]}] RD SD PD TD ALL } [d2 ... dn][(options...[)]] 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 END]] [{.} [bytecount END]]) [*dumpid]
DUMPSCAN IPCS Displays areas of the CP abend dump produced by the VMFDUMP command. Enters the cited subcommand into a table. &name recalls the subcommand. The & alone lists the table entries.	DUMPSCAN The DUMPSCAN subcommands are: &name subcommand

DUMPSCAN (cont.)	
Displays last subcommand entered.	?
Displays the registers, clocks, PSW, CSW, and CAW for attached processor.	Aregs
Displays the control registers.	C
Displays areas in the dump.	Display {hexloc } [nnnn] or Display {Thexloc } [nnnn] {hexloc%} {Thexloc%}
Locates the CP module that includes the address hexloc, displays its current loaded address and displacement of hexloc.	DMK? hexloc
Lists load map address and real address for module mmm, entry name ee.	DMKmmmee
Ends and returns to CMS.	END
Displays the general registers.	G
Displays a summary of subcommands.	HELP

Description	Format
DUMPSCAN (cont.)	
Ends and returns to CMS.	HX
Searches between the from and to address for the given string.	Locate string fromhexloc tohexloc [increment]
Adds the file NUC MAP A to the dump being processed.	MAP
Displays the registers, clocks, PSW, CSW, and CAW for main processor.	Mregs
Reissues the previous subcommand and prints the data.	Print PRT
-- or --	-- or --
Issues subcommand and prints data. Turns PRINT on. Turns PRINT off. Sends printed data to real printer. Displays PRINT status.	Print [subcommand] PRT [ON] [OFF] [CLOSE] [?]
Ends and returns to CMS.	QUIT
Displays registers, clocks, PSW, CSW, and CAW for failing processor.	Regs

<table>
<tr><td>DUMPSCAN (cont.)</td><td></td></tr>
<tr><td>Displays the RCHBLOK, RCUBLOK, and RDEVBLOK for cuu.</td><td>Rioblok cuu</td></tr>
<tr><td>Displays the area ahead of or behind the current area.</td><td>Scroll [U]
ScrollU</td></tr>
<tr><td>Displays trace table entries.</td><td>Trace [nnn|19]</td></tr>
<tr><td>Displays the VCHBLOK, VCUBLOK, and VDEVBLOK for cuu and userid.</td><td>Vioblok cuu [userid]</td></tr>
<tr><td>Lists userids, VMBLOK address, and status of logged-on users.</td><td>Vmblok</td></tr>
<tr><td>ECHO CP Class G
Returns data directly to the terminal a specified number of times.</td><td>ECho [nn
1]</td></tr>
<tr><td>EDIT CMS
Provides access to the EDIT environment.
Note: In all formats of the CMS EDIT subcommands and macros, use of the word "subcommand" means a CMS EDIT subcommand only.</td><td>Edit fn ft [fm|*] [(options...[)]]
options:
[LRECL nn] [NODISP]</td></tr>
</table>

Description	Format
EDIT (cont.)	The EDIT subcommands and macros are:
Duplicates the current line. This is a macro.	$DUP [1|n]
Moves n lines up m or down m lines. This is a macro.	$MOVE n {Up m Down m TO label}
Displays the last executed EDIT subcommand except for the REUSE(=) or ? (question mark) subcommands.	?
Scans n records, altering the specified character.	ALter char1 char2 [n [G]] [* [*]] [1]
Saves the file after the indicated number of changes or displays the current setting.	AUTOsave [n|OFF]
Repositions the current line pointer toward the top of the file.	BAckward [1|n]
Moves the current line pointer to the last line of the file.	Bottom

EDIT (cont.)

<table>
<tr><td>Translates to uppercase.</td><td>CASE [U|M]</td></tr>
<tr><td>Changes string1 to string2.</td><td>Change [/string1[/string2[/[n|*|1][G|*]]]]]</td></tr>
<tr><td>Enters CMS subset command mode.</td><td>CMS</td></tr>
<tr><td>Deletes n lines or to EOF.</td><td>DELete [n|1|*]</td></tr>
<tr><td>Moves the current line pointer to the nth line down from the current line.</td><td>DOwn [n|1]</td></tr>
<tr><td>Deletes lines from the current line down to (but not including) the line that contains the designated string.</td><td>DString /[string[/]]</td></tr>
<tr><td>Saves the file edited on disk and returns to CMS mode.</td><td>FILE [fn [ft [fm]]]</td></tr>
<tr><td>Searches the file for the specified line.</td><td>Find [line]</td></tr>
</table>

Description	Format
EDIT (cont.)	
Resets or displays the filemode.	FMode [fm]
Resets or displays the filename.	FName [fn]
Changes the mode of displaying data on a 3270 terminal from typewriter (line) style to full-screen display style or vice versa.	FORMat {DISPLAY|LINE}
Moves the current line pointer forward n lines.	FOrward [1|n]
Inserts some or all of the specified file.	Getfile {fn *} [ft * [fm * [firstrec 1 [numrec *]]]][1]
Expands text (including backspace and tab) into line images or displays current settings.	IMAGE [ON|OFF|CANON][2]
Inserts a line into the file or enters input mode.	Input [line]

EDIT (cont.)	
Sets or cancels line-number editing, or displays current setting.	LINEmode [LEFT|RIGHT|OFF][3] [L |R |]
Scans the file for the first occurrence of a string.	[Locate]|/[string[/]]
Enters LONG error message mode. LONG is the default mode.	LONG
Advances the current line pointer n lines toward the end of the file.	Next [n|1]
Replaces all or part of the current line.	Overlay [line]
Saves current mode settings.	PREserve
Sets the line increment for line-number editing.	PROMPT [n|10]
Terminates the EDIT session leaving the previous copy of the file intact.	QUIT

[1] Operands are positional.
[2] ON is the default for all filetypes except SCRIPT. CANON is the default for SCRIPT files.
[3] Line-number editing is the default for USBASIC and FREEFORT files.

Description	Format
EDIT (cont.)	
Sets or displays record format.	RECfm [F|V]
Recomputes line numbers.	RENum [strtno [incrno]] [10 [strtno]]
Executes the following OVERLAY request n times or to EOF.	REPEAT [n|1|*]
Replaces the current line with 'line' or deletes the line and enters input mode.	Replace [line]
Restores mode settings.	REStore
Returns to EDIT environment.	RETURN
Stacks (LIFO) the last EDIT subcommand and executes the stacked subcommands.	{REUSE} [subcommand] { = }
Saves the file on disk.	SAVE [fn [ft [fm]]]
Displays a number of lines above or below the current line.	{Scroll } [*|n|1] {S[croll]U[p]}

```
EDIT (cont.)

    Turns serialization on or off      SERial ( OFF   ┌    ┐ )
    in columns 73-80.                         (  ON   │incr│ )
                                              (  ALL  │ 10 │ )
                                              (  seq  └    ┘ )

    Enters SHORT error message         SHORT
    mode.

    Stacks n lines in the terminal     STACK [n|1|0|subcommand]
    input buffer.

    Sets the specified tabs.           TABSet n1 [n2 ... nn]

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

    Sets or displays the column of     TRUNC  [n|*]
    truncation.

                                              ┌ ┌ ┐┐
    Displays the specified number      Type   │m│n││
    of lines beginning with the               │*│*││
    current line.                             │1│ ││
                                              └ └ ┘┘

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

Description	Format
EDIT (cont.) Sets, displays, or resets verify mode.	Verify [ON] [[startcol] endcol] [OFF] [[1] *]
Assigns to X or Y the given EDIT subcommand.	{X} [subcommand|n|1] {Y}
Sets or displays the columns to be edited.	Zone [firstcol [lastcol]] [* [*]] [1]
Locates the line when using line-number editing.	{nnnnn } [text] {nnnnnnnn}

ENABLE CP Classes A,B
Activates communication lines.
	ENable {raddr... }
	{SNA[userid]}
	{ALL }

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/VS macros from system or private source statement E sublibraries to CMS disk files; or lists de-edited macros.	ESERV fn
EXEC CMS Invokes EXEC files.	[EXec] fn [args...] The formats of the EXEC control statements are:
Assigns a variable.	&variable = {ae|string|function|X'xxxxxx}
Defines or redefines arguments to special variables &1-&30.	&ARGS [arg1 [arg2 ... [arg30]]]
Displays the following unscanned error message lines at the terminal. The list of lines to be displayed must end with an &END control statement.	&BEGEMSG [ALL][1]
Punches the following lines into cards. End the list of lines with an &END control statement.	&BEGPUNCH [ALL]

[1] See the &EMSG statement for the format of message lines.

Description	Format
EXEC (cont.)	
Stacks lines into the terminal input stack. The list is terminated by the &END control statement.	&BEGSTACK [LIFO|FIFO] [ALL]
Displays lines at the terminal The list is terminated by the &END control statement.	&BEGTYPE [ALL]
Provides a branching address for EXEC branch statements.	&CONTINUE
Specifies the data to be displayed in the execution summary of an EXEC.	&CONTROL [OFF [TIME [NOPACK [NOMSG ERROR NOTIME] PACK] MSG] ALL CMS]
Displays a line of tokens to be edited as an error message.	&EMSG mmmnnns [tok1...[tokn]]
END statement for action started by &BEGPUNCH, &BEGSTACK &BEGEMSG, or &BEGTYPE.	&END
Provides error processing.	&ERROR [executable-statement|&CONTINUE]

```
EXEC (cont.)
    Exits from the EXEC file with       |   &EXIT   [return-code|0]
    a specified return code.            |
                                        |
    Transfers control to a              |   &GOTO  {TOP|line-number|-label}
    specified location.                 |
                                        |
    Initiates or terminates hexa-       |   &HEX  {ON|OFF}
    decimal conversion in an EXEC       |
    procedure.                          |
                                        |
    Allows statement execution if       |   &IF   (tok1) operator (tok2) executable-statement
    the comparison is satisfied.        |         {&$  }          {&$  }
    If the comparison is invalid,       |         (&*  )          (&*  )
    execution continues with the        |
    statement following the &IF         |
    statement.                          |
                                        |
    Describes a loop in an EXEC,        |   &LOOP { n     } {m         }
    and conditions for exit from        |         {-label } {condition }
    the loop.                           |
                                        |
    Punches a card with the             |   &PUNCH  [tok1 [tok2 ... [tokn]]]
    specified tokens.                   |
```

Description	Format
EXEC (cont.) Reads the next line (or lines) from the terminal stack.	&READ [n ARGS VARS[&var1[&var2 ... [&varn]]] 1]
Skips subsequent statements.	&SKIP [n|1]
Displays blank lines at the terminal.	&SPACE [n|1]
Places a line of tokens in the console input stack.	&STACK [FIFO LIFO] [tok1[tok2 ... [tokn]] HT RT]
Displays time information on the terminal after each CMS command is executed.	&TIME [ON|OFF|RESET|TYPE]
Displays the specified tokens at the terminal.	&TYPE [tok1 [tok2 ... [tokn]]
	The formats of EXEC built-in functions are:
Concatenates two or more tokens and assigns the result to a variable symbol.	&variable = &CONCAT tok1[tok2 ... [tokn]]

EXEC (cont.)	
Determines whether the specified token is alphabetic or numeric.	&variable = &DATATYPE token
Indicates number of nonblank characters in the specified token.	&variable = &LENGTH token
Allows the use of the literal value of the token without symbolic substitution.	[...] &LITERAL token [...]
Extracts a character string from the specified token.	&variable = &SUBSTR token i [j]
	The formats of EXEC special variables are:
Represents the numeric variables &1 through &30.	&n
Used to perform a collective test on all arguments passed to EXEC.	&* and &$
Contains the filename of the EXEC file.	&0

Description	Format
EXEC (cont.)	
Used to determine whether a disk is an OS, DOS, or CMS disk.	&DISKx
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.	&DISK?
Indicates the current status of the CMS/DOS environment (ON or OFF).	&DOS
Indicates the filename of the EXEC file.	&EXEC
Indicates the recursion level of the EXEC procedure that is currently executing (maximum - 19 levels).	&GLOBAL
Represents the numeric variables &GLOBAL0 through &GLOBAL9.	&GLOBALn

EXEC (cont.)	
Indicates the number of arguments passed to the EXEC procedure (maximum - 30).	&INDEX
Indicates the current line number in the EXEC file.	&LINENUM
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.	&READFLAG
Indicates the return code from the most recently executed CMS command.	&RETCODE
Indicates either the RT (resume typing) or HT (halt typing).	&TYPEFLAG
EXEC 2 CMS Invokes EXEC 2 files.	EXEC filename [arg1 [arg2 ... argn]] The formats of the EXEC 2 predefined variables are:
Initializes to its own name. This variable is initialized or maintained automatically.	&

Description	Format
EXEC 2 (cont.)	
Initializes to the first word of the command string passed to the EXEC 2 interpreter. This variable is initialized or maintained automatically.	&0
Initializes to the arguments arg1, arg2, ..., argn since they themselves are arguments and then are passed to the EXEC 2 file.	&1, &2, ..., &n
Initializes to the command string passed to the EXEC 2 file. This variable is treated as a single literal string.	&ARGSTRING
Assigns the value of a blank.	&BLANK
Initializes to zero and keeps the line number of the last EXEC 2 file issued command or subcommand.	&COMLINE
Initialized to the untranslated command string that is passed to the EXEC 2 file.	&CMDSTRING

EXEC 2 (cont.)	
Evaluates true date (primary meridian -- GMT) in the form: YY/MM/DD. (See also &TIME, below.)	&DATE
Keeps number of user-defined function and subroutine invocations to which return not yet made.	&DEPTH
Initializes to third qualifier of EXEC 2 file.	&FILEMODE
Initializes to first qualifier of EXEC 2 file.	&FILENAME
Initializes to second qualifier EXEC 2 file.	&FILETYPE
Initializes to zero and keeps line number of last executed &GOTO statement of EXEC 2 file.	&FROM
Keeps current line number of EXEC 2 file.	&LINE,&LINENUM
Keeps line number of currently executing user-defined function or subroutine invoker or is zero.	&LINK

Description	Format
EXEC 2 (cont.)	
Keeps the number of EXEC 2 arguments set. (See &1, &2, ... &n, above.)	&N,&INDEX
Initializes to zero, and keeps return code from last EXEC 2 issued command or subcommand.	&RC,&RETCODE
Evaluates true time-of-day (primary meridian -- GMT) in the form: HH:MM:SS. (See also &DATE, above.	&TIME
	The formats of the EXEC 2 control statements are:
Assigns word1, word2, ... wordn to arguments &1, &2, ... &n and discards previously set arguments.	&ARGS [word1 [word2 ...]]
Prints line1, line2, ... linen, truncated at column k if necessary. Does not remove surplus blanks or replace any EXEC 2 variables.	&BEGPRINT [n [k]] &BEGTYPE [label [*]] [1] [*] line1 line2 ...

```
EXEC 2 (cont.)
    Places line1, line2, ... linen,   &BEGSTACK ┌                   ┐
    in the console stack,                       │n ┌k ┌FIFO┐┐      │
    truncated at column k if                    │* │* │LIFO││      │
    necessary.  Does not remove                 │  └  └    ┘┘      │
    surplus blanks or replace any               │label             │
    EXEC 2 variables.                           │1                 │
                                                └                   ┘

                                      line1
                                      line2
                                      ...

    Discards lookaside buffer and     &BUFFER  n  [comment]
    its contents, then creates a               *
    new lookaside buffer for either
    designation.

    Invokes the routine located at    &CALL  line-number [arg1 [arg2...]]
    the specified label or line              label
    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.
```

Description	Format
EXEC 2 (cont.)	
Translates any lowercase alphabetic character to uppercase or allows mixed cases. If U or M is not specified, the current setting is not changed.	&CASE ⎡U [comment]⎤ ⎣M⎦
Issues the command comprising the word operands, each with one space between.	&COMMAND word1 [word2 ...]
Aids debugging the EXEC 2 interpreter DMSEXE. Intended to to be used by system support people only. Note: An asterisk (*), a hyphen (-), or an ampersand (&) starting a command must be given as an argument.	&CRASH [text]
Prints a line for each &1, &2, ..., &n argument or variables var1, var2, ..., varn.	&DUMP ARGS VAR[S] [var1 [var2 ...]]

EXEC 2 (cont.)

<table>
<tr><td>Sets the action to be automatically invoked on return from any command(s) or subcommand(s) that has a nonzero return code.</td><td>&ERROR action</td></tr>
<tr><td>Stops execution of the EXEC 2 file and yields the given numeric return code within the host system acceptable range.</td><td>&EXIT [return-code [comment]
<u>0</u>]</td></tr>
<tr><td>Transfers control to the designated line number or to the line with the label. The first character of label must be a hyphen.</td><td>&GOTO line-number [comment]
label</td></tr>
<tr><td>Executes the given executable statement if the condition is satisfied; otherwise, proceeds to next statement.</td><td>&IF word1 =|EQ [word2 executable statement]
¬=|NE
<=|LT
<=|¬>|LE|NG
>|GT
>=|¬<|GE|NL</td></tr>
</table>

Description	Format
EXEC 2 (cont.)	
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.	&LOOP n m label * WHILE condition UNTIL
Presumes that any statement without a beginning ampersand is to be issued to CMS or to the designated subcommand environment.	&PRESUME &COMMAND &SUBCOMMAND environment
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.	&PRINT [word1 [word2 ...]] &TYPE
Reads from the stack (if stack is not empty); otherwise, reads from the console the number of lines indicated, or assigns values as designated.	&READ n 1 * ARGS VAR[S] [var1 [var2...]] STRING var

EXEC 2 (cont.)

Returns control to the most recently invoked subroutine to which no return has as yet been made.	&RETURN [word] [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.	&SKIP [n [comment]] [1]
Stacks a line that contains word(s) that are separated by one space or stacks a null line if no words are given.	&STACK [FIFO [word1 [word2 ...]]] [LIFO]
Issues the designated subcommand comprising word(s) separated by one space to the appropriate environment.	&SUBCOMMAND environment [word1 [word2 ...]]

Description	Format
EXEC 2 (cont.) 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.	&TRACE [ON [output-action]] ERR ALL OFF ***
Sets the truncation column to k or the maximum value (*). If no argument is shown, the previous setting stays in effect.	&TRUNC [k [comment]] *
Translates any lowercase alphabetic characters to uppercase in the values of &1, &2, ... &n or the values of var1, var2, ..., varn.	&UPPER ARGS VAR[S] [var1 [var2 ...]]

EXEC 2 (cont.)	The formats of the EXEC 2 predefined functions are:
Concatenates the word(s) with no intervening space into a single word. If no word(s) appear, a null line results.	&CONCATENATION OF [word1 [word2 ...]] &CONCAT
Yields the value NUM if word represents a valid signed or unsigned number; otherwise, the value is CHAR.	&DATATYPE OF [word] &TYPE
Yields a numeric value representing the integral part of the division of the dividend by the divisor.	&DIVISION OF dividend divisor &DIV
Left-justifies word of length j. Truncates or pads with blanks on the right-hand side.	&LEFT OF word j
Gives either the number of characters in word or zero if word is not given.	&LENGTH OF [word]

Description	Format
EXEC 2 (cont.)	
Gives the literal string beginning with character after blank following OF and ending with the last nonblank character. Note: Any leading or embedded blanks are retained and search for replacement variables is suppressed.	&LITERAL OF [string]
Searches word for first occurrence of char(s) and gives the starting position number, or a zero when there is no matching string, or char(s) exceeds word length, or a word is not given.	&LOCATION OF needle [haystack]
Yields numeric value that results from the multiplying of given numeric signed or unsigned words. Note: There must be at least two of these.	&MULTIPLICATION OF i j [k...] &MULT

EXEC 2 (cont.)	
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.	&PIECE OF word i [j] &SUBSTR [*]
Compares and tries to match word1 with word(s). 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.	&POSITION OF word [word1 [word2 ...]]
Yields a string comprising words made by appending numbers to the stem ranging from n1 to n2 with one blank between each or, if n1 is greater than n2, yields a null string. *Note*: Appended numbers are stripped of any plus sign or redundant leading zeros.	&RANGE OF stem i j

Description	Format
EXEC 2 (cont.)	
Right-justifies word of length j. Truncates or pads with blanks on left-hand side.	&RIGHT OF word j
Gives the string beginning with character after blank following OF and ending with last nonblank character.	&STRING OF [string]
Compares each character in word1 with word2. If a match if found, the position of that matching character in word1 is replaced with the character in the same position from word3.	&TRANSLATION OF word1 [word2 [word3]] &TRANS
If word has trailing blanks, removes them. If word is not given, result is a null line.	&TRIM OF [word]
Gives the nth word in the list of words unless the number given is zero or exceeds the number of words in the list.	&WORD OF [word1 [word2 ... wordn]] i

EXEC 2 (cont.) 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.	The format of the EXEC 2 User-Defined functions are: line-number OF [arg1 [arg2...]] label
EXECIO CMS Do I/O operations between a device and the program stack.	EXECIO {lines | *} DISKR fn ft [fm [linum]] [([FINIs] options [a] [b]] [)] CARD [(options [a] [b]] [)] CP [(options [a] [b] [d]] [)] DISKW fn ft fm [lineum [recfm [lrecl]]] [([FINIs] options [b] [c] [d]] [)] PUNCH [(options [b] [c] [d]] [)] PRINT [([CC {CODE | DATA}]] options [b] [c] [d] EMSG [(options [b] [c] [d]] [)]

EXECIO (cont.)

```
 Option formats:
   (a)                   (b)                   (c)                   (d)

      FInd /chars/          Margins {n1 n2}       CAse {U}              STring xxx...
      LOcate /chars/                {1  * }            {M}
      Avoid /chars/         STRIP
      Zone {n1 n2}          NOTYPE
           {1  * }
      LIFO
      FIFO
      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.

Description	Format
EXTERNAL CP Class G Simulates an external interruption condition on the virtual machine and returns control to that machine.	EXTernal [code | 40]

FEEDBACK

Places comments about the operation of the system and/or the Programmable Operator in the feedback file.

```
FEEDBACK  text...
FB
```

FETCH CMS

Loads an executable phase into storage for execution.

```
FETch  phasename  [(options...[)]]
options:
[START]
[COMP]
[ORIGIN hexloc]
```

FILEDEF CMS

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

```
FIledef ┌ {ddname} ┌ Terminal [(optionA optionD[)]]                    ┐ ┐
        │ {nn    } │                                                    │ │
        │ {*     } │  PRinter [(optionA OPTCD[)]]                       │ │
        │          │  PUnch   [(optionA  [)]]                           │ │
        │          │  Reader                                            │ │
        │          │                                                    │ │
        │          │ 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[)]]       │                    │ │
        │          └ └                              ┘                    ┘ ┘
```

```
Description              Format

FILEDEF (cont.)                   DUMMY [(optionA[)]]

                                  TAP[n] [LABOFF
                                          BLP[n]
                                          SL[n] [VOLID volid][DISPMOD]
                                          SUL[n] [VOLID volid]
                                          NSL filename
                                          NL[n]                     ]

                                          cuu   [(optionA optionC optionE[)]]
                                  GRAF    cuu   [(optionA [)]]

                                  CLEAR

                         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][E][F][J][R]}
                                  [DISP  MOD] [MEMBER membername]
                                  [DSORG {PS|PO|DA}]

                         optionC: [7TRACK|9TRACK] [TRTCH {0|OC|OT|E|ET}]
                                  [DEN {200|556|800|1600|6250}]

                         optionD: [UPCASE|LOWCASE]

                         optionE: [LEAVE]
                                  [NOEOV]
```

<u>FILELIST</u> CMS Lists information about CMS disk files, with the ability to edit and issue commands from the list.	FILEList [fn [ft [fm]]] [(options...[)]] <u>options:</u> [Append] [Filelist | <u>Nofilelist</u>] [PROFile fn]
<u>FINIS</u> CMS Closes one or more files.	FINIS {fn} {ft} [fm] {*} {*} [*]
<u>FLUSH</u> CP Class D Halts and immediately purges or holds the current spool file.	Flush raddr [ALL] [HOld]
<u>FLUSH</u> RSCS Stops and deletes the current file from further processing.	Flush [linkid] [spoolid|*] [ALL][HOld]

Description	Format
FORCE CP Class A Forces logoff of the named user.	FORCE userid
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 [nocyl | noblk] [(options...[)]] options: [Blksize [800 | 1024 | 2048 | 4096 | 1K | 2K | 4K] Noerase Label Recomp]
FORMAT/ALLOCATE Service Aid Formats, allocates, and labels direct access volumes for paging, spooling, and CP file residence.	Format Service Aid Control Statements FORMAT,devadr,devtype,volser,startcyladr,endcyladr • Format Function FORMAT,devadr,devtype,volser,startcyladr,endcyladr • Allocate Function TEMP,startcyladr,endcyladr PERM,startcyladr,endcyladr TDSK,startcyladr,endcyladr DRCT,startcyladr,endcyladr END

FORMAT/ALLOCATE (cont.)		• Label Function FORMAT,devadr,devtype,volser,LABEL
FREE Releases previously held user spool files.	CP Class D	FRee userid [Printer PUnch ALL]
FREE Causes I/O transmission on a particular link to resume.	RSCS	FRee [linkid][nnn]
FWDSPACE Causes the file currently being processed to be repositioned in a forward direction.	RSCS	FWdspace [linkid][nnn]
GENDIRT Creates auxiliary module directories.	CMS	GENDIRT directoryname [targetmode[sourcemode]]

Description	Format
GENERATE CMS Generates or updates VM/SP or the VM/SP directory, or generates a new standalone copy of a service program.	GENERATE {VM/SP DIRECT [ONLY] DMKRIO [ONLY] DMKSYS [ONLY] DMKFCB [ONLY] DMKSNT [ONLY] IPLDECK SRVCPGM [CP CMS] NUCLEUS [NOLOAD] RSCS [BUILD]}
GENIMAGE CMS Presents input control file to the OS utility program IEBIMAGE. Creates text files used by the 3800 printer.	GENIMAGE [fn SYSIN] [ft FILE] [fm *] [sfn SYSPRINT] [sft LISTING] [sfn A1]

```
GENMOD                          CMS |Genmod  [fn [MODULE [fm|A1]]] [(options...[)]]
   Generates a nonrelocatable       |
   (MODULE) file on a CMS disk.     |options:
                                    |
                                    |┌     ┐ ┌     ┐                            ┌   ┐
                                    ||NOMAP| |STR  | [FROM entry1] [SYSTEM]    |OS |
                                    ||MAP  | |NOSTR|                           |DOS|
                                    |└     ┘ └     ┘ [TO   entry2]             |ALL|
                                    |                                          └   ┘
                                    |
GEN3705                         CMS |GEN3705 fn ft [fm] [(options...[)]]
   Invokes 3705 Stage 2 service aid.|options:
                                    |[RUN|NORUN][SAVE|NOSAVE]
                                    |
GET                                 |GET   ⎧FEEDBACK     ⎫
   Retrieves one of the Programmable|      ⎨FB           ⎬
   Operator files: the feedback file|      ⎩LOG [yymmdd] ⎭
   (FB) or the log file (LOG).      |
                                    |
GLOBAL                          CMS |GLobal   ⎧MACLIB ⎫ [libname1 ... libname8]
   Specifies CMS or CMS/DOS         |         ⎪TXTLIB ⎪
   libraries to be searched for     |         ⎨DOSLIB ⎬
   macros, copy files, subroutines, |         ⎩LOADLIB⎭
   or DOS executable phases when    |
   processing CMS commands.         |
```

Description	Format
GLOBALV CMS Set, maintain, and retrieve a collection of named variables. Note: Although this command may be used in CMS EXECs, it is designed for use with EXEC 2 EXECs. Restrictions/precaustions are listed in "Usage Notes for the CMS EXECs".	GLOBALV INIT SELECT [group UNNAMED] SET SETS name1[value1][name2 value2]... SETP SETL SELECT group SETLS UNNAMED SETSL name [value] SETLP SETPL LIST [name1 [name2]...] STACK name1 [name2]... STACKR SELECT group PURGE UNNAMED GRPLIST GRPSTACL PURGE
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 descriptions, formats, and parameters of CMS and CP commands and EXECs, and description of CMS and CP messages.	Help [Help | message | MENU | component MENU | [component | CMS] {name[(option[)]]}] options: [ALL|FORM|PARM|DESC]
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]

Description	Format
HOLD RSCS Suspends file transmission temporarily for a particular 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
IDENTIFY CMS Display or stack userid, nodeid, rscsid, date, time, time zone and day of week.	IDentify [(options...[)]] options: STACK [FIFO|LIFO] FIFO LIFO TYPE

IMAGELIB CMS

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

```
IMAGELIB namedsys
```

IMAGEMOD CMS

Allows changes to the 3800 named systems.

```
IMAGEMOD {GEN}   libname[modname[modname]..]}
         {ADD}
         {REP}
          DEL libname membername1 [membername2...]
          MAP libname [(options[)]]

options:  [TERM ]
          [PRINT]
          [DISK ]
```

INCLUDE CMS

Brings additional TEXT files into virtual storage.

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

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

[NOAUTO]  [LIBE  ]  [NODUP]  [MAP  ]  [NOINV]  [REP  ]  [SAME]
[AUTO  ]  [NOLIBE]  [DUP  ]  [NOMAP]  [INV  ]  [NOREP]
```

Description	Format
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 [LOAD USER [* | userid] Queues I/O PAGing [WAIT | ALL] FAVORed]
INDICATE CP Class G Displays the use of and contention for major system resources.	INDicate [LOAD USER]
IPL CP Class G Simulates an initial program load function for a virtual machine.	Ipl { vaddr [cylno | nnnnnn] [CLear | NOCLear] [STOP] [ATTN] systemname } [PARM {p1 p2...}]

LABELDEF CMS

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

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

options:

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

LINK CP Class G

Permits one user to access minidisks belonging to another user.

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.

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

Description	Format
LISTDS CMS Displays information about data sets, files, extents, or free space.	LISTDS [dsname] {fm} [(options...[)]] [?] {*} options: [FREE] [FORMAT] [EXTENT] [PDS]
LISTFILE CMS Lists information about CMS files.	Listfile [fn [ft [fm]]] [(options...[)]] [* [* [*]]] options: [Header|NOHeader][Exec[Trace][Args]]|Trace[Args]|APpend[Args] |Stack[Fifo|Lifo]|Fifo|Lifo] [Blocks] [%x] [FName|FType|FMode|FOrmat|ALloc|Date|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|APPEND] [STAT]

```
LKED                           CMS  LKED fn [(options...[)]]
  Used to create a CMS               options:
  LOADLIB or LOADLIB member.          [NCAL] [LET] [ALIGN2] [NE] [OL] [RENT]
                                      [REUS] [REFR] [OVLY] [XCAL] [TERM|NOTERM]
                                      [NAME membername] [LIBE libraryname]
                                      [XREF|MAP|LIST] [PRINT|DISK|NOPRINT]
                                      [SIZE {value1 value2 | value1 | value1, | ,value2 | , }]

LOAD                           CMS  LOAD  fn ... [(options...[)]]
  Brings TEXT files into storage    options:
  and establishes links.
                                    [CLEAR  ]  [RESET{entry}]  [NOMAP]  [NOINV]  [NOREP]  [START]
                                    [NOCLEAR]  [      {  *  }]  [MAP  ]  [INV  ]  [REP  ]


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

LOADBUF                 CP Class D  LOADBUF  {raddr UCS name [Fold] [Ver]  }
  Loads UCS (Universal Character             {raddr FCB name [Index [nn]] }
  Set) buffer or FCB (forms control
  buffer) on real printer.
```

Description	Format
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] *
LOADTBL Loads a new routing table to control the operation of the Programmable Operator.	LOADTBL [filename]
LOADVFCB CP Class G Loads a forms control image for a virtual 3211 or 3203 printer.	LOADVFCB vaddr Fcb name [Index [nn]]

```
LOCATE                   CP Class C, E  LOCate   {userid[vaddr]}
  Provides the addresses of CP                   {raddr        }
  control blocks related to a
  specified user, virtual device,
  or real device.

LOCK                       CP Class A   LOCK   {userid} firstpage lastpage [MAP]
  Locks specified pages in                     {SYSTEM}
  processor storage.

LOG                                     LOG  text...
  Writes a message to the current
  day's log file.

LOGOFF                   CP Class Any    LOGoff   [HOld]
  Terminates a terminal session.         LOGout

LOGON                    CP Class Any    Logon   userid [password]  [Noipl]
  Initiates all virtual machine          Login
  operation.

MACLIB                            CMS   MAClib  ( {GEN}                                   )
  Creates and updates macro                     ( {ADD} libname fn1[fn2...]               )
  libraries.                                    ( {REP}                                   )
                                                (  DEL libname membername1[membername2...])
                                                (  COMP libname                           )
                                                (  MAP libname  (options... )             )

                                        options:
                                        [DISK|PRINT|TERM]
```

<table>
<tr><th>Description</th><th>Format</th></tr>
<tr><td>MAKEBUF CMS
Creates a new buffer within the program stack.</td><td>MAKEBUF</td></tr>
<tr><td>MESSAGE CP Classes A and B
Sends text messages to other users, system operator or self.</td><td>Message {ALL|userid|*|OPerator} msgtext
MSG</td></tr>
<tr><td>MESSAGE CP Class Any
Sends text messages to other users, system operator or self.</td><td>Message {userid|*|OPerator} msgtext
MSG</td></tr>
<tr><td>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.</td><td>MIGrate [userid]</td></tr>
<tr><td>MODMAP CMS
Displays a MODULE file load map.</td><td>MODmap fn</td></tr>
<tr><td>MONITOR CP Classes A and E
Starts or stops the recording of interruptions and other events that occur in the real machine.</td><td>MONitor {Display [SPOOL|TAPE|ALL]
ENable {PERForm|RESPonse|SCHedule|USER|
INSTsim|DAStap|SEEKs|SYSprof}[1] }</td></tr>
</table>

MONITOR (cont.)

Note: Modifications to DMKMCC are necessary to use the Response option.

```
{ INTerval nnnnn [SEC|MIN][mm]                                  }
{                                                               }
{ STArt   [SPOOL   [to userid] [BUFFS nn]                  ]    }
{         [TAPE raddr [MODE { 800}] [BUFFS n]               ]    }
{         [           [     {1600}]                         ]    }
{         [           [     {6250}]                         ]    }
{         [                                                 ]    }
{         [CPTRACE                                          ]    }
{                                                               }
{ STOP    [SPOOL|TAPE|CPTRACE]                                  }
{                                                               }
{ CLOSE                                                         }
{ AUTOdisk {ON|OFF}                                             }
{                                                               }
{ TIME       {FROM h1:m1 TO h2:m2}                              }
{            {FOR  hh:mm         }                              }
{            {ALL                }                              }
{            {NONE               }                              }
{                                                               }
{ LIMIT n [NOSTOP|STOP|SAMPLE]                                  }
{                                                               }
{ SEEKS      {INclude raddr raddr...}                           }
{            {EXclude raddr raddr...}                           }
{            {DELete                }                           }
{            {DISplay               }                           }
```

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

Description	Format
MOVEFILE CMS Moves data from one device to another device.	MOVEfile [inddname [outddname]] [(PDS[)]] [INMOVE [OUTMOVE]]
MSG RSCS Allows message text to be sent via RSCS remote or local stations to any RSCS facility or VM/370 user.	Msg linkid [userid] msgtext
MSGNOH CP Class B Allows a virtual machine to send messages without the standard header associated with the MESSAGE command.	MSGNOH {userid} msgtext {ALL }
NAMEFIND CMS Display/stack information from a NAMES file. (default 'userid NAMES').	NAMEFind :tag value [:tag [value]]... [options...[)]] options: [STACK [n|*| 1] [FIFO|LIFO]] [FIle fn] [FIFO [n|*| 1]] [LINenum] [LIFO [n|*| 1]] [STARt recnum] [TYPE [n|*| 1]] [SIze[n|*| 8]]
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 |NCPDUMP  [DUMPxx]
|  Processes CP spool reader files|
|  created by 3705 dumping        | [([ERASE] [NOFORM] [MNEMONIC][NCPBUFF][)]]
|  operations.                    |
|                                 |
|NETWORK               CP Class A |NETWORK  SHUTDOWN [raddr|ALL]
|  Stops communications to 370x   |
|  controllers or resources or 3270|
|  remote equipment.              |
|NETWORK         CP Class A and B |NETWORK (LOAD raddr ncpname                              )
|  Controls the 370x control      |        |DUMP raddr [IMMED|OFF|AUTO]                      |
|  program and its resources.  Also|       |ENable [ALL[resid[resid...]]                     |
|  provides a means of altering   |        |DISAble [ALL[resid[resid...]]                    |
|  binary synchronous line poll   |        |Query[ACTive|OFFline|FREe|ALL|resid[resid...]]   |
|  delay interval.                |        {                                                 }
|                                 |        |POLLdlay nnnn [ALL|raddr]                        |
|                                 |        |VARY {ONline|OFFline[resid...]                   |
|                                 |        |ATTach resid [TO] userid [AS] cuu                |
|                                 |        |                                                 |
|                                 |        (DETach resid [FROM] userid                       )
|                                 |
|NOTE                         CMS |NOTE [{nickname1          } [nickname2          ]]
|  Prepare a 'note' for one or more|    [{userid1 [AT node]} [userid2 [AT node]]...
|  computer users, to be sent via |
|  the SENDFILE command.          |     [CC: {nickname1          } [nickname2          ]]]
|                                 |     [    {userid1 [AT node]} [userid2 [AT node]]...]]
```

<table>
<tr><th>Description</th><th>Format</th></tr>
<tr><td>NOTE (cont.)</td><td>[(options...[)]]

options:
[ACk| NOAck]
[ADd]
[Cancel]
[NOTebook fn|NOTebook *|NONotebook]
[LOG|NOLog]
[LONg|Short]
[Replace]
[PROFile fn]</td></tr>
<tr><td>NOTREADY CP Class G
Simulates loss of ready status on virtual device.</td><td>NOTReady vaddr</td></tr>
<tr><td>NUCXDROP CMS
Deletes specified nucleus extensions.</td><td>NUCXDROP {name1 [name2...]}
{*}</td></tr>
<tr><td>NUCXLOAD CMS
Loads a nucleus extension</td><td>NUCXLOAD {name [fn]}
{name member ddname}
[([SYstem] [SErvice] [Push] [)]]</td></tr>
<tr><td>NUCXMAP CMS
Identifies existing nucleus extensions.</td><td>NUCXMAP [ALL] [([STACK] [LIFO] [)]]
[FIFO]</td></tr>
</table>

```
OPTION                           CMS |OPTION  [options...]
  In CMS/DOS, changes any or all of  |options:
  options in effect for the DOS/VS   |
  COBOL compiler.  Only specified    |   ┌      ┐┌      ┐┌      ┐┌       ┐┌     ┐
  options are changed.               |   |DUMP  ||DECK  ||LIST  ||LISTX  ||SYM  |
                                     |   |NODUMP||NODECK||NOLIST||NOLISTX||NOSYM|
                                     |   └      ┘└      ┘└      ┘└       ┘└     ┘
                                     |
                                     |   ┌      ┐┌      ┐┌   ┐┌      ┐
                                     |   |XREF  ||ERRS  ||48C||TERM  |
                                     |   |NOXREF||NOERRS||60C||NOTERM|
                                     |   └      ┘└      ┘└   ┘└      ┘
                                     |
ORDER                     CP Class D |ORDer ┌      ┐ (READER )(CLass c1 Class c2...          )
  Places closed spool files in a     |      |userid| {Printer}{spoolid1 spoolid2...            }
  specified order by device type.    |      |SYSTEM| (PUnch  )(FORM form1 FORM form2 ...    )
  (A combination of CLASS and        |      |  *   |
  spoolid specifications may be      |      └      ┘
  entered.)                          |
                                     |
ORDER                     CP Class G |ORDer  (Reader )(CLass c1 CLass c2...   )
  Places closed spool files in a     |       {Printer}{FORM form1 FORM form2...}
  specified order by device type.    |       (PUnch  )(spoolid1 spoolid2...    )
                                     |
ORDER                           RSCS |ORDer  [linkid] {spoolid1 [spoolid2 ...]}
  Reorders file queue for a link.    |
                                     |
OSRUN                            CMS |OSRUN member [PARM=parameters]
  Executes a load module from a CMS  |
  LOADLIB or an OS module library.   |
```

Description	Format
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]
PRINT CMS Directs a specified spool file to the virtual printer.	PRint fn ft fm [(options...[)]] * options: CC [HEADer] TRC NOCC [UPCASE] NOTRC LINECOUN nn MEMBER * 55 membername [HEX]

PRB IPCS

Updates the STATUS, LASTFNCT, SEV, or PTF information for a specific problem number.

```
PRB  {       {APAR aparnumber              }¹}
     {       {CLOSE                        } }
     {       {DSPLY                        } }
     {       {DUPOF mmmmm                  } }
     {nnnnn  {IBM                          } }
     {       {NEEDINFO                     } }
     {       {PTFIS [filename] filetype    } }
     {       {PTFON                        } }
     {       {SEV   s                      } }
     {       {USER                         } }
     {HELP                                   }
```

PROB IPCS

Enters a non-CP abend problem into the system or adds information to an existing problem report.

```
PROB
```

PSERV CMS

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

```
PSERV procedure [ft|PROC] [(options...[)]]
options:
[DISK] [PRINT] [PUNCH] [TERM]
```

[1] The sequence of nnnnn and the other keywords can be reversed. Note, however, that nnnnn is a more recent problem than mmmmm.

Description	Format
PUNCH CMS Directs a specified spool file to the virtual punch.	PUnch fn ft [fm|*] [(options...[)]] options: [HEADER] [MEMBER {membername}] [NOHEADER] {*}

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.)
	PURge [userid] {Reader [ALL]}
	[SYSTEM] {Printer [CLass c1 CLass c2 ...]}
	[*] {PUnch [spoolid1 spoolid2 ...]}
	[FORM form1 FORM form2 ...]
	{ALL}

PURGE CP Class G
Deletes a closed file before reading, printing, or punching occurs. (A combination of CLASS and spoolid specifications may be entered.)
	PURge {Reader [CLass c1 CLass c2 ...]}
	{Printer [FORM form1 FORM form2 ...]}
	{PUnch [spoolid1 spoolid2 ...]}
	{ALL [ALL]}

PURGE RSCS
Removes inactive file queued on a specified link before it has been selected for processing.
	PURge [linkid] {ALL}
	{spoolid1 [spoolid2...]}

QUERY CP Classes A,B,C,D,E,F,G
Displays system software maintenance information, log messages, the number of logged-on-users; lists logged-on users.
	Query {CPLEVEL}
	{LOGmsg}
	{Names}
	{Users [userid]}
	{userid}

QUERY CP Class A

Displays system status, paging, scheduling, and machine configuration information.

```
Query {S370E }
      {SPMODE}
```

QUERY CP Classes A,E

Provides the paging activity index or specified user priority or status of the Virtual Machine Assist feature.

```
Query   { PAGing                                           }
        { PRIORity userid                                  }
        { SASsist¹                                         }
        { CPAssist¹                                        }
        { AFFinity [userid]                                }
        { QDROP                                            }
        { JOurnal²                                         }
        { ATTach resid [To] userid [As] cuu                }
        { DETach resid [From] userid                       }
        { SRM {APAGes  }                                   }
        {     {DSPSlice}                                   }
        {     {MAXWss  }                                   }
        {     {MAXDrum }                                   }
        {     {PCI     }                                   }
        {     {PB      }                                   }
        {     {IB      }                                   }
        {     {PGMStat }                                   }
        {     {PGMTlim }                                   }
        {     {MHFULL  }                                   }
```

[1] Combined use of QUERY CPASSIST and QUERY SASSIST determines the current status of the expanded virtual machine assist portion of VM/SP Extended Control Program Support (ECPS).

[2] The JOURNAL operand is valid only if STQUERY=YES is specified in the SYSJRL macro instruction in DMKSYS.

Description	Format
QUERY CP Class B Provides the current status of all system devices.	Query DAsd [Sysvirt] [ACTive] [PATHS] [Virtual] [OFFline] TApes [FREe] LINES[1] [ATTach] UR [ALL] GRaf ALL DAsd volid TDsk STORage raddr1 [-raddr2] SYStem raddr DUMP Lnnn
QUERY CP Class D Provides data on spooling operations.	Query Files [CLass c] [FORM form] [userid|*] [NOHold][2] [HOld] [USERhold] [SYShold] Reader [CLass c][FORM form] [userid] [NOHold] [ALL] PUnch [*] [NOld] [TBL] Printer [SYShold] [USERhold] HOLD spoolid

```
QUERY                          CP Class G |Query  ⎧   Time                                                        ⎫
   Provides system status and             |       ⎪   Set                                                         ⎪
   machine configuration                  |       ⎪   Terminal                                                    ⎪
   information.                           |       ⎪                                                               ⎪
                                          |       ⎪   Files   [CLass c] [FORM form] |NOHold  |                    ⎪
                                          |       ⎪                                 |HOld    |                    ⎪
                                          |       ⎪                                 |USERhold|                    ⎪
                                          |       ⎪                                 |SYShold |                    ⎪
                                          |       ⎪                                 └        ┘                    ⎪
                                          |       ⎪ ⎧                  ┌               ┐⎫                          ⎪
                                          |       ⎪ ⎪                  |CHANnels       |⎪                          ⎪
                                          |       ⎪ ⎪                  |GRAF           |⎪                          ⎪
                                          |       ⎨ ⎪                  |CONsole        |⎪                          ⎬
                                          |       ⎪ ⎪                  |DAsd           |⎪                          ⎪
                                          |       ⎪ ⎪                  |TApes          |⎪                          ⎪
                                          |       ⎪ ⎨                  |LINES          |⎬                          ⎪
                                          |       ⎪ ⎪ [Virtual]        |UR             |⎪                          ⎪
                                          |       ⎪ ⎪                  |STORage        |⎪                          ⎪
                                          |       ⎪ ⎪                  |ALL            |⎪                          ⎪
                                          |       ⎪ ⎪                  |vaddr[-vaddr]  |⎪                          ⎪
                                          |       ⎪ ⎩                  └               ┘⎭                          ⎪
                                          |       ⎪   Links vaddr                                                 ⎪
                                          |       ⎪   CPUid                                                       ⎪
                                          |       ⎩   SECuser                                                     ⎭
```

[1] Query LINES is not effective for 3704/3705 resources unless the 3704/3705 is operating in 270x Emulation Program (EP) mode. VM/SP does not support the Network Control Program (NCP) mode nor the MTA (Multiple Terminal Access) feature of NCP. However, if the VTAM Service Machine (VSM). component of SNA (Systems Network Architecture) is in place, the appropriate network control program will be in effect. VM/SP also does not support the Partitioned Emulation Program (PEP) mode.

[2] These operands may be in any sequence.

Description	Format
QUERY (cont.)	Reader Printer PUnch [CLass c] [FORM form] spoolid [NOHold | HOld | USERhold | SYShold] [ALL] [TBL] PF [nn] SCReen PROCessr VMSAVE S370E SPMODE USERID
QUERY CP Classes A,B,C, and E Displays the online processors in the system.	Query PROCessr
QUERY CMS In CMS or CMS/DOS mode, provides information about the virtual machine environment.	Query ABBREV BLIP DISK [mode|*|R/W|MAX] FILEDEF IMPCP IMPEX INPUT LABELDEF LDRTBLS LIBRARY MACLIB OUTPUT PROTECT [(options...[)]] options: STACK [FIFO]|LIFO] FIFO LIFO

```
QUERY (cont.)                        |        (RDYMSG                             )
                                     |        |LOADLIB                            |
                                     |        |TXTLIB                             |
                                     |        |CMSLEVEL                           |
                                     |        <REDTYPE                            >
                                     |        |RELPAGE                            |
                                     |        |SEARCH                             |
                                     |        |SYNONYM {SYSTEM|USER|ALL}          |
                                     |        (SYSNAMES                           )
                                     |
                                     |     CMS/DOS Functions:
                                     |        (DLBL                               )
                                     |        |DOS                                |
                                     |        |DOSLIB                             |
                                     |        <DOSLNCNT                           >
                                     |        |DOSPART                            |
                                     |        |OPTION                             |
                                     |        (UPSI                               )
                                     |
QUERY                                |QUERY  RTABLE
  Indicates the name of the Prog-    |
  rammable Operator's active         |
  routing table.                     |
                                     |
QUERY                          RSCS  |Query (linkid [Stat|Def|Queue]        )
  Queries RSCS linkid, file or       |      <File spoolid [Stat|RSCS|VM]    >
  system status information.         |      (SYstem [Active]                )
                                     |
QVM                      CP Class A  |QVM userid [NORETURN]
  Requests the transition from       |
  VM/370 to a particular virtual     |
  machine, running in native mode.   |
```

Description	Format
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 isssue commands from list.	RDRlist [(options...[)]] RList options: [PROFile fn] [Append]

READCARD CMS
Reads data from the virtual card reader, and creates CMS disk files containing the data.
	READcard { fn ft [fm|A] }
	{ * [* [fm|A]] }

READY CP Class G Makes a device end interruption pending for the specified device.	READY vaddr

RECEIVE CMS Read onto disk a file or note that is in your virtual reader	RECEIVE [spoolid][fn [ft [fm]] [(options...[)]]] options: [NOTebook fn|NOTebook *] [Log | NOLog] [Purge] [Replace] [Olddate| NEwdate] [STack]
RELEASE CMS Makes a disk and its directory inaccessible to a virtual machine.	RELease {cuu|mode} [(DET [)]]
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 {[nn|1] }
	{[nn]HOLD}

REQUEST CP Class G Makes an attention interruption pending.	REQuest
RESET CP Class G Clears all pending interruptions; resets error conditions on the device specified.	RESET vaddr

Description	Format
REWIND CP Class G Rewinds a real tape drive.	REWind vaddr
RO CMS Immediate Command Resumes recording of trace information previously suspended by the SO Immediate command.	RO
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...[)]]
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] [CAMOD {0}] [NAME ncpname] [LIBE libraryname] [CXFINIT] [{1}] [fn] [fn]

SAVESYS CP Class E

Creates a copy of virtual machine storage, registers, and PSW.

```
SAVESYS  systemname
```

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  {      {[ {extcolor}  {exthilight} ] }1 }
        { area {[ {DEFault }  {NONe      } ] }  }
        {      {[ {         }  {          } ] }  }
```

SEND CP Class G

Passes commands and message replies to the designated disconnected virtual machines for execution.

```
 SEND [CP] userid [text]
```

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]{nickname1        }{[nickname2        ]...
 SFile                  {userid1[AT node]}{[userid2[AT node]]

                             [options...[)]]]
        options:
          [Ack|NOAck]
          [Filelist|NOFilelist]
          [Log|NOLog]
          [NEw|Old]
          [NOTE]
          [Type|NOType]
```

SENTRIES CMS

Determines the number of lines currently in the program stack.

```
SENTRIES
```

[1] Each time you enter the command, you must specify at least one screen 'area' operand with at least one 'extcolor' and/or 'exthilight' value. You may specify more than one 'area' operand on the same command line.

Description	Format
SET CP Class A Sets special CP preferred options.	SET AFFinity [userid] [ON|OFF|nn] CPAssist {ON|OFF} [PROC[nn]] FAVORed userid [xxx[xx]|OFF] REServe userid {xx|OFF} SASsist {ON|OFF} [PROC[nn]] PRIORity userid nn JOurnal[1] {LOgon} [ON|OFF] {LInk} S370E {ON|OFF} [[PROC]addr] QDROP userid {ON|OFF}[USERS]
SET CP Class B Establishes disposition for log messages and dumps.	SET LOGmsg [nn|NULL] DUMP {AUTO|raddr} [ALL|CP]
SET CP Class E Sets SRM function and the number to be used in the working set size estimate control algorithm.	SET SRM MAXWss {nnnn|OFF} MAXDrum {nnnn|OFF} PCI {DRUM|DISK} PB nn IB n PGMTlim MHFULL {nnn|OFF} APAGES nnnn DSPSlice nn.n PAGing nn

<u>SET</u> 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}
```

<u>SET</u> CP Class G

The SET command controls various functions within your virtual machine.

```
Set  ( (ACNT    ) (    ) )
     ( (AUTOPoll) (    ) )
     ( (RUN     ) (ON  ) )
     ( (LINEDit ) (OFF ) )
     ( (NOTRans ) (    ) )
     ( (ECmode  ) (    ) )
     ( (ISAM    ) (    ) )
     ( (PAGEX   ) (    ) )
     { EMSG {ON|OFF|CODE|TEXT|IUCV}  }
     ( TIMER {ON|OFF|REAL}           )
     ( CPUid bbbbbb                  )
     ( IMSG ( ON )                   )
     ( MSG  {OFF}                    )
     ( SMsg (IUCV)                   )
     (                               )
     ( CPCONIO {OFF }                )
     ( VMCONIO {IUCV}                )
```

[1] The JOURNAL operand is valid only if STQUERY=YES is specified in the SYSJRL macro instruction DMKSYS.

Description	Format
SET (cont.)	ASsist {[ON][SVC|NOSVC][TMR|NOTMR]} OFF AFFinity [ON|OFF] PFnn [IMMed][pfdata1#pfdata2#...pfdatan] [DELayed] PFnn [TAB n1 n2...nn] [resid] PFnn [COPY {[cuu]}] [luname] PFnn RETrieve VMSAVE[1] {ON|OFF|name} STBypass {[{nnnnnK} [NOVERIFY]] nnM VR OFF} STMulti {[n [[USEG xx]CSEG yyy]] ON OFF} 370E [ON|OFF]

```
|SET                            CMS |SET function
|  Control various functions within |  functions: [LDRTBLS nn] [OUTPUT xx a]   [RDYMSG|LMSG]
|  your virtual machine.  (Only one |                                          |RDYMSG SMSG|
|  function may be specified per SET|                                          L           J
|  command.                         |     [INPUT{a xx|xx yy}][RELPAGE{OFF|ON}][ABBREV{OFF|ON}]
|                                   |
|                                   |     [BLIP(string [(count)])]  [PROTECT{OFF|ON}] [CMSTYPE{HT|RT}]
|                                   |     |    {ON              }|
|                                   |     |    (OFF             )|
|                                   |     L                      J
|                                   |
|                                   |      [IMPEX OFF] [IMPCP OFF] [REDTYPE OFF] [AUTOREAD ON ]
|                                   |      |IMPEX ON | |IMPCP ON | |REDTYPE ON | |AUTOREAD OFF|
|                                   |      L         J L         J L           J L            J
|                                   |
|                                   |      [                                  ] [                     ]
|                                   |      |SYSNAME  (CMSDOS )  entryname| |NONSHARE  (CMSDOS )|
|                                   |      |         )CMSVSAM(            | |          )CMSVSAM(|
|                                   |      |         <CMSAMS >            | |          <CMSAMS >|
|                                   |      |         )CMSSEG (            | |          )CMSSEG (|
|                                   |      |         (CMSBAM )            | |          (CMSBAM )|
|                                   |      L                              J L                   J
|                                   |
|                                   |      [CMSTYPE {HT}]
|                                   |      |        {RT}|
|                                   |      L            J
|                                   |    CMS/DOS functions:
|                                   |
|                                   |      [DOS ON [mode[(VSAM[)]]]] [UPSI nnnnnnnn ] [DOSPART nnnnnK ]
|                                   |      |DOS OFF                | |UPSI OFF      | |DOSPART OFF    |
|                                   |      L                       J L              J L               J
|                                   |      [DOSLNCNT nn]
```

[1] When specifying this operand, virtual machine size cannot exceed eight megabytes.

Description	Format

SET
Enter and exit from Programmable Operator DEBUG mode.
	SET DEBUG { ON }
	{ OFF }

SETKEY CMS Sets the storage key for a discontiguous 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[(]mmmm[n][)]]
SHUTDOWN CP Class A Checkpoints and terminates the current VM/370 operation.	SHUTDOWN
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 |SMsg  userid msgtext
  Sends a 'special message' to a         |
  virtual machine which is running       |
  with SET SMSG ON.                      |
                                         |
SO                 CMS Immediate Command |SO
  Suspends the recording of trace        |
  information during the execution       |
  command or program.                    |
                                         |
SORT                                 CMS |SORT  fileid1 fileid2
  Sorts records within a file and        |
  creates a new file containing the      |
  sorted records.                        |
                                         |
SPACE                         CP Class D |SPAce  raddr
  Forces single spacing on the           |
  printer.                               |
                                         |
SPMODE                        CP Class A |SPMode {ON }
  Establishes or resets the              |       {OFF}
  single processor mode.                 |
                                         |
                                         |                 { [          ]  [      ] [     ]  [      ] }1
SPOOL                         CP Class G |SPool {Reader}  { [CLass {*}]  [CONt  ] [EOF  ]  [HOLD  ] }
  Changes spooling control options.      |      {vaddr }  { [       {c}]  [NOCont] [NOEof]  [NOHOLD] }
```

[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.

Description	Format
SPOOL (cont.)	

```
                                                                          1
 {Printer}  ( [ [To ] {userid} ] [HOld  ] [CONt  ]                        )
 {PUnch  }  ( [ [For] {  *   } ] [NOHold] [NOCont]                        )
 {CONsole}  ( [       {SYSTEM} ]                                          )
 {vaddr  }  ( [                ]                                          )
            ( [  OFF           ] [CLass c][COpy[*]nnn]                    )
            (                            3                                )
            ( [CLOSE]   [FLash name nnn    ]  [FORM {form}]               )
            ( [PURGE]   [MOdify name[n]    ]  [     {OFF }]               )
            (           [Chars name1       ]                              )
            (           [      [name2      ]                              )
            (           [      [name3      ]                              )
            (           [      [name4]]]   ]                              )
            (           [CHars name1       ]                              )
            (           [ [CHars name2     ]                              )
            (           [ [CHars name3     ]                              )
            (           [ [CHars           ]                              )
            (           [        name4]]]  ]                              )
            (           [FCB name          ]                              )
            (                                          2          2  1    )
            ( [Dist {distcode}]  { [STArt]  [TErm   ]  }              )
            ( [     {OFF     }]  { [STop ]  [NOTErm ]  }              )
```

SPTAPE CP Class D

Dumps output spool files to tape or loads output spool files from tape.

```
SPTape  { STOP   raddr                                                          }
        {                                                                       }
        { CANCEL raddr                                                          }
        {                                                                       }
        { SCAN raddr [LEAVE ]                                                   }
        {            [REWind]                                                   }
        {            [RUN   ]                                                   }
        {                                                                       }
        { LOAD raddr {Printer} {spoolid1 [spoolid2]   } [LEAVE ]                }
        {            {PUnch  } {         [END     ]   } [REWind]                }
        {            {Reader } {                      } [RUN   ]                }
        {                      {CLass c1 CLass c2...  }                         }
        {                      {FORM form             }                         }
        {                      {ALL                   }                         }
        {        [[SYSHOLD][USERHOLD][NOHOLD]]                                  }
        {                                                                       }
        { DUMP raddr{Printer} {spoolid1 [spoolid2]} [MODE { 800}]               }
        {           {PUnch  } {         [END     ]} [     {1600}]               }
        {           {Reader } {                   } [     {6250}]               }
        {                     {CLass c1 CLass c2..}                             }
        {                     {FORM form          }                             }
        {                     {ALL                }                             }
```

[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 D/T 3800 as a virtual spooling device.

Description	Format
SPTAPE (cont.)	{ [LEAVE REWind RUN] [[SYSHOLD][USERHOLD][NOHOLD]][PURGE] }
SSERV CMS In CMS/DOS, copies, displays, prints, or punches a book from a DOS/VS source statement library.	SSERV sublib bookname [ft|COPY] [(options...[)]] options: [DISK] [PRINT] [PUNCH] [TERM]
START CP Class D Restarts a drained device or changes its output spooling class.	STArt [Reader Printer PUnch ALL raddr[CLass c...] [FORM form FORM *] [NOsep] [Auto SETup MAnual] [NO3800 BEG3800 ANY3800] [FLash name CHars name FCB name IMage namedsys PUrge]]

```
START                        CMS  START   ┌┌entry┐ ┌args...┐┐
  Begins program execution.               ││  *  │ │       ││
                                          │└     ┘ └       ┘│  option:
                                          │(option[)]       │    NO
                                          └                 ┘

START                       RSCS  STArt  [linkid] ┌CLass c        ┐ [Parm...]
  Activates an RSCS link that is                  │LINE vaddr     │
  in the deactive state to start                  │TASK name      │
  processing files.                               │TYPE driverid  │
                                                  └               ┘

                                                  Status Keywords:        Failure Keywords:
STAT                        IPCS  STAT  { nnnnn }  ┌          ┐            ┌               ┐
  Provides current status of a          { ALL   }  │OPENUSER  │            │ABEND          │
  problem, a specific subset of                    │OPENIBM   │            │{DOC|DD}       │
  problems, or all problems.                       │OPEN      │            │INcorr         │
                                                   │APARED    │            │INF            │
                                                   │NEEDINFO  │            │{LOOP|LP}      │
                                                   │PTFRCVD   │            │MSg            │
                                                   │PTFON     │            │{PERFORM|PR}   │
                                                   │CLOSED    │            │{WAIT|WS}      │
                                                   └          ┘            └               ┘

STATE/STATEW                 CMS  {STATE }  {fn}  {ft}  {[fm]}
  STATE verifies the existence of a {STATEW}  {* }  {* }  { *  }
  file on any accessed disk.
  STATEW verifies the existence of
  a file on a read/write disk.
```

Description	Format
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 hexdata NShexloc
STOP Stops operation of the Programmable Operator.	STOP
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 |SVCTrace  {ON }
  Records information about       |          {OFF}
  supervisor call instructions.   |
                                  |
SYNONYM                       CMS |SYNonym [fn [              [fm]]] [(options...[)]]
  Specifies alternate names for   |        [   [SYNONYM       [A1]]]
  invoking CMS commands.          |        [   [              [ *]]]
                                  |
                                  |options:
                                  |[NOSTD|STD] [CLEAR]
                                  |
SYSTEM                 CP Class G |SYStem   {CLEAR  }
  Simulates virtual machine console|        {RESET  }
  functions and clears virtual    |         {RESTART}
  storage and storage keys to     |
  binary zeros.                   |
                                  |
TAG                    CP Class G |TAg  DEv {Printer} [tagtext]
  Appends or queries the TAG text |         {PUnch  }
  to a VM/SP spool file utilized  |         {CONsole}
  by subsystems (such as          |         {vaddr  }
  RSCS).                          |
                                  |     FIle spoolid [tagtext]
                                  |
                                  |     QUery {      {Printer} }
                                  |           {      {PUnch  } }
                                  |           { DEv  {CONSole} }
                                  |           {      {vaddr  } }
                                  |           { FIle spoolid   }
```

Description	Format
<u>TAPE</u> CMS Performs tape to disk or disk to tape operations for CMS files.	TAPE { DUMP {fn | *} {ft | *} [fm | *] [(optionA optionB optionD[)]] LOAD [{fn | *} {ft | *} [fm | <u>A</u>]] [(optionB optionC optionD[)]] SCAN [{fn | *} {ft | *}] [(optionB optionC optionD[)]] SKIP [{fn | *} {ft | *}] [(optionB optionC optionD[)]] DVOL1 [(optionD optionE[)]] WVOL1 volid [owner] [(optionD optionE[)]] MODESET [(optionD[)]] tapcmd [n|<u>1</u>] [(optionD[)]] } <u>Note</u>: The tapcmd operand can be one of the following: [BSF|BSR|ERG|FSF|FSR|REW|RUN|WTM]

```
TAPE (cont.)                     |optionA:                               optionB:     optionC:
                                 |
                                 |┌      ┐ ┌                ┐           ┌        ┐   ┌     ┐
                                 ||WTM   | |BLKSIZE{4096}   |           |NOPRint |   |EOF n|
                                 ||NOWTM | |       { 800}   |           |PRint   |   |EOT  |
                                 |└      ┘ └                ┘           |DISK    |   |EOF 1|
                                 |                                      |Term    |   └     ┘
                                 |                                      └        ┘
                                 |
                                 |optionD:
                                 |
                                 |┌┌     ┐ ┌    ┐┐                                    ┌      ┐
                                 |||TAPn | |cuu ||  [TRTCH {0|OC|OT|E|ET}]            |7TRACK|
                                 |||TAP1 | |181 ||                                    |9TRACK|
                                 |└└     ┘ └    ┘┘                                    └      ┘
                                 |
                                 |      [DEN {200|556|800|1600|6250}]
                                 |
                                 |optionE:
                                 |
                                 |┌      ┐
                                 ||LEAVE |
                                 ||REWIND|
                                 |└      ┘
                                 |
TAPEMAC                       CMS|TAPEMAC fn ┌                                ┐ [(options...[)]]
  Creates a CMS MACLIB from an   |           |SL [labeldefid]                 |
  unloaded partitioned data set  |           |NSL filename [ID=identifier]    |
  (PDS) from a tape created by the|          └                                ┘
  IEHMOVE utility program under OS.|options:
                                 |
                                 |┌     ┐ ┌            ┐
                                 ||TAPn | |ITEMCT yyyyy|
                                 ||TAP1 | |ITEMCT 50000|
                                 |└     ┘ └            ┘
```

Description	Format
TAPPDS CMS Creates CMS disk files from tapes which are input to or output from the IEBPTPCH, IEBUPDTE, and IEHMOVE OS utility programs.	TAPPDS fn ft fm SL [labeldefid] * * * NSL filename [ID=identifier] A1 [(options...[)]] options: UPDATE COL1 TAPn END MAXTEN NOPDS NOCOL1 TAP1 NOEND NOMAXTEN PDS
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 {nickname } message {userid [AT node]}
TERMINAL CP Class G Controls virtual console functions.	TERMinal { {CHardel LINEDel LINENd EScape TABchar[2]} {ON OFf char} {APL TEXT ATtn HIlight SCRNsave} {ON OFf} }[1]

```
TERMINAL (cont.)                      |       ⎧ MODE {CP|VM}                 ⎫
                                      |       ⎪                              ⎪
                                      |       ⎪ LINESize {nnn|OFF}           ⎪
                                      |       ⎪                              ⎪
                                      |       ⎪ CONmode {3215|3270}          ⎪
                                      |       ⎪                              ⎪
                                      |       ⎨ BREakin {IMmed|GUESTctl}     ⎬
                                      |       ⎪                              ⎪
                                      |       ⎪ BRKkey {PA1|PF1|...|PF24}    ⎪
                                      |       ⎪                              ⎪
                                      |       ⎪ TYpe {3101|TTY}              ⎪
                                      |       ⎪                              ⎪
                                      |       ⎪ PROMpt {VM|TTY}              ⎪
                                      |       ⎪                              ⎪
                                      |       ⎩ SCROLL {nnn|CONT}            ⎭
TRACE                     CP Class G  |TRace ⎧⎧SVC      ⎫³ ┌⎧PRINTer       ⎫            ┐⎫
  Traces and records program          |      ⎪⎪I/O      ⎪  │⎪              ⎪            │⎪
  execution.                          |      ⎪⎪PROgram  ⎪  │⎨┌TERMinal┐    ⎬ ┌NORun┐    │⎪
                                      |      ⎪⎪EXTernal ⎪  │⎪│BOTH    │    ⎪ │RUN  │    │⎪
                                      |      ⎪⎪PRIV     ⎪  │⎩└        ┘    ⎭ └     ┘    │⎪
                                      |      ⎨⎨SIO      ⎬  │OFf                        │⎬
                                      |      ⎪⎪CCW      ⎪  └                           ┘⎪
                                      |      ⎪⎪BRanch   ⎪                                ⎪
                                      |      ⎪⎪INSTruct ⎪                                ⎪
                                      |      ⎪⎪ALL      ⎪                                ⎪
                                      |      ⎪⎩CSW      ⎭                                ⎪
                                      |      ⎩ END                                       ⎭
```

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

```
    TRACE SVC PROGRAM SIO PRINTER
```

Description	Format
TRACE RSCS Traces and records certain line and error activity for a specified link. Available to RSCS operator only.	TRace linkid [ALL|ERRors|END]
TRANSFER CP Class D Transfers closed reader spool files.	TRANsfer [userid] [PRinter] {spoolid } {[TO] { * }} [Printer] [SYSTEM] [PUnch] {CLass c } { {userid}} [PUnch] [*] [Reader] {FORM form} { } [Reader] {ALL } {FROM {ALL }} { {userid}}
TRANSFER CP Class G Transfers closed reader spool files.	TRANsfer [Printer] {spoolid } {[TO] {userid}} [Printer] [PUnch] {CLass c } { { * }} [PUnch] [Reader] {FORM form} {FROM {userid}} [Reader] {ALL } { {ALL }}
TRAPRED CMS A reduction program that will display to a terminal or print a CPTRAP READER file.	TRAPRED filenum

TXTLIB CMS
Updates a library of TEXT files (object modules).
	TXTlib {GEN libname fn1 [fn2...] }
	{ADD libname fn1 [fn2...] }
	{DEL libname membername1 [membername2...]}
	{MAP libname [(options...[)]] }
	options (for MAP only): [TERM] [PRINT] [DISK]

TYPE CMS Types or displays all or part of a file at a terminal.	Type fn ft [fm] [rec1 [recn]] [(options...[)]] [*] [* [*]] [1 [ ]]

TYPE (cont.)

```
options:

[COL  {xxxxx}  -[yyyyy ]]  [HEX]  [MEMBER{*   }]
[     { 1   }   [lrecl ]]         [      {name}]
```

UNLOCK — CP Class A
Releases pages of storage.

```
UNLOCK  {{userid|SYSTEM} firstpage lastpage}
        {                                  }
        {{VIRT=REAL|V=R}                   }
```

UPDATE — CMS
Makes changes in a program source file as defined by control statements in a record file.

```
Update fn1 [ft1      [fm1 [fn2 [ft2 [fm2]]]   ]]  [(options...[)]]
           [ASSEMBLE [A1                      ]]
options:

[REP  ] [SEQ8  ] [INC  ] [NOCTL] [STK  ] [TERM  ] [PRINT] [NOSTOR]
[NOREP] [NOSEQ8] [NOINC] [CTL  ] [NOSTK] [NOTERM] [DISK ] [STOR  ]

Control Statements:
  ./ S  [seqstrt [seqincr [label]]]
  ./ I seqno [$ [seqstrt [seqincr]]]
  ./ D seqno1 [seqno2] [$]
  ./ R seqno1 [seqno2] [$ [seqstrt [seqincr]]]
  ./ * [comment]
```

VARY — CP Class B
Varies the availability of a device.

```
VARY    {ONLine } {raddr...     }
        {OFFline} {raddr-raddr  }
                  {PROCessor    }
        OFFline PROCessor nn [options] [FORCE]
          options:

          [VPHY]
          [VLOG]
```

Description	Format
VMDUMP CP Class G Dumps storage for virtual machine. It enables sending dumps to other users. Used in conjunction with the VM/IPCS Extension.	VMDUMP [hexloc1] [{-} [hexloc2]] [TO *] [FORMAT vmtype] [0] [{:} [END]] [TO userid] [SYSTEM] [{.} [bytecount]] [DSS] [*dumpid] [END]
VMFDUMP IPCS,Service Aid Formats and prints, or erases, an existing dump.	VMFDUMP [PRBnnnnn] [ERASE] [NOMAP] [NOHEX] . . . [NOFORM] [NOVIRT]

WARNING CP Classes A,B
Sends high priority messages.
	{Warning} {userid } msgtext
	{WNG } {OPerator}
	{ALL }

XEDIT CMS Creates, modifies, and manipulates CMS disk files. This is the VM/SP System Product editor. Once XEDIT is invoked you may execute XEDIT subcommands and use the EXEC 2 macro facility.	XEDIT [fn [ft [fm]]] [options...[)]] options: [Width nn] [NOSCreen] [PROFile macroname] [NOCLear] [NOPROFil]

XEDIT (cont.) Note: In all formats of the XEDIT subcommands and macros, use of the word "subcommand" means an XEDIT subcommand only.	options valid only in Update mode: Update Seq8 Ctl fnl [Merge] NOUpdate NOSeq8 NOCtl [Incr nn] [SIDcode string]
	The XEDIT subcommands and macros are:
Redisplays the subcommand and allows reexecution by pressing the ENTER key.	& [subcommand]
Displays the last executed XEDIT subcommand except for the = (equal sign) or the ? (question mark) subcommands.	?
Reexecutes the last subcommand or macro that was entered. Also executes a specified subcommand and then reexecutes the last one entered.	= [subcommand]
Inserts blank lines immediately following the current line.	Add [n|1]

Description	Format
XEDIT (cont.)	
Changes a single character to another character unavailable on terminal keyboard by referencing their hexadecimal values. This is a macro.	ALter char1 char2 [target|1 [n|*|1|G [p|1]]]
Scrolls backward the number of screen displays specified.	BAckward [n|*|1]
Makes the last line of the file or of the range (see SET RANGE) the new current line.	Bottom
Terminates the editing session for all of the files. This is a macro.	CANCEL
Appends specified text to the end of the current line. This is a macro.	CAppend [text]
Deletes one or more characters from the current line, starting at the column pointer.	CDelete [column-target|1]
Moves the column pointer to the beginning of the zone (see SET ZONE).	CFirst

XEDIT (cont.)	
Changes a specified group of characters on one or more lines at one time.	Change /old [/new/ [target [p [q]]]] [*] 1 1 1
Inserts text into the current line immediately ahead of the column pointer.	CInsert text
Moves the column pointer to the end of the zone (see SET ZONE).	CLAst
Scans the file for a specified column-target starting at the column following (or preceding) the column pointer in the curent line. Also finds successively all occurences of a character string. string.	CLocate column-target
Forces the editor to transmit a command to CMS for execution or causes the editor to enter CMS subset mode.	CMS [commandline]
Displays a message in the command line; intended for issuance from a macro.	CMSG [text]

Description	Format
XEDIT (cont.)	
Causes the editor to execute a specified XEDIT command without first checking for a synonym or macro with the same name.	COMMAND [commandline]
Prepares one or more lines for automatic repositioning of data (see SET TABS).	COMPress [target|1]
Copies one or more lines at a specified location in the file.	COpy target1 target2
Displays the number of times a specified character string appears in one or more lines, beginning at the current line.	COUnt /string [/target|1]
Selectively replaces one or more characters in the current line with the same number of characters keyed in.	COVerlay text
Transmits commands to the VM/SP control program environment during an editing session.	CP [commandline]
Replaces one or more characters in the current line.	CReplace text

XEDIT (cont.)

Moves the cursor to a specified position.	CURsor CMdline [colno|1] Column File lineno [colno] Screen lineno [colno]
Deletes one or more lines from a file beginning with the current line.	DELete [target|1]
Moves the line pointer down a specified number of.lines toward the end of the file.	Down [n|*|1]
Duplicates one or more lines beginning with the current line.	DUPlicat [n|1[target|1]]

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.	EMSG {text}
	{[mmmnnns text]}

Description	Format
XEDIT (cont.)	
Repositions data in one or more lines that contain tab characters (X'05').	EXPand [target|1]
Writes the edited file onto disk and optionally overrides the originally supplied file identifier.	FILE [fn [ft [fm]]] [= [= [=]]]
Searches forward, starting with the current line, for the operand-specified text.	Find text
Searches backward, starting with the current line for the operand-specified text.	FINDUp text FUp
Scrolls toward the end of the file the operand-specified number of screen displays.	FOrward [n|*|1]
Inserts all or part of a specified CMS file following the current line of the edit file.	GET [fn|=[ft|=[fm|=|*][firstrec|1[numrec|*]]]]

XEDIT (cont.)	
Displays a list of all XEDIT subcommands and macros and their descriptions, formats, and parameters, or invokes the CMS HELP command.	Help [MENU|HELP|name]
Displays a specified number of lines in both hexadecimal and EBCDIC. This is a macro.	HEXType [target|1]
Inserts a single line into a file. Also used to leave edit mode for entry into input mode.	Input [line]
Merges two lines at the column pointer or at the cursor. This is a macro. Also merges two or more lines at a specified column number(s) or inserts a specified character string(s) before appending the next line.	Join [Column CURSOR colno ... /string/ ...]
Allows viewing of columns not currently visible on the screen that lie to the left of column one.	LEft [n|1]

Description	Format
XEDIT (cont.)	
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. Note: Options specified in the XEDIT command override those specified in the LOAD subcommand.	LOAD [fn [ft [fm]]] [(options...[)]] options: [Width nn] [NOSCreen] [PROFile macroname | NOPROFil] [NOCLear] options valid only in update mode: [Update|NOUpdate] [Seq8|NOSeq8] [CTl fn1|NOCtl] [Merge] [Incr nn] [SIDcode string]
Scans file for a specified target, which (when found) becomes the current line.	[Locate] target [subcommand]
Changes all uppercase letters to lowercase letters in one or more lines.	LOWercas [target|1]
Causes the specified operand to be executed as a macro.	MACRO [macroline]

XEDIT (cont.)

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.

```
MODify keyword

keyword operands:
  APL        FType      PACK       SYNonym
  ARBchar    HEX        PFn        TABLine
  AUtosave   IMage      Point      TABS
  CASE       IMPcmscp   PREfix     TERMinal
  CMDline    LINENd     RANge      TEXT
  COLPtr     LRecl      RECFm      TOFEOF
  COLumn     MACRO      SCALe      TRunc
  CURLine    MASK       SCReen     VARblank
  ESCape     MSGMode    SERial     Verify
  FILler     NONDisp    SPAN       VERShift
  FMode      NULls      STAY       WRap
  FName      NUMber     STReam     Zone
```

Moves one or more lines, beginning with the current line, to a specified place in the file.

```
MOve target1 target2
```

Displays a message in the message area of the screen.

```
MSG [text]
```

Advances the line pointer a specified number of lines toward the end of the file.

```
Next [n|*|1]
```

Description	Format
XEDIT (cont.)	
Searches forward for the first line that does not start with the text specified in the operand.	NFind text
Searches backward for the first line that does not start with the text specified in the operand.	NFINDUp text NFUp
Replaces, selectively, one or more characters with nonblank characters starting at the first tab column of the current line.	Overlay text
Helps in writing new macros by scanning the new macro(s) to see whether or not the format specified operands match those in the macro. This is a macro.	PARSE startcol Alphaword Number String Dblstring ... Target Word Line
Enters an input mode where data can be keyed in as though the screen were one long line.	POWerinp

XEDIT (cont.)

Saves the settings of various XEDIT variables until a subsequent RESTORE subcommand is issued.

```
PREServe

settings saved include:
   ARBCHAR    IMAGE      PACK       TABS
   AUTOSAVE   LEFT       PREFIX     TOFEOF
   CASE       LINEND     RECFM      TRUNC
   ESCAPE     LRECL      RIGHT      VERIFY
   FILLER     MACRO      SERIAL     VARBLANK
   FMODE      MASK       SPAN       WRAP
   FNAME      MSGMODE    STAY       ZONE
   FTYPE      NULLS      STREAM     =
   HEX        NUMBER     SYNONYM
```

Removes a copy of a macro in virtual storage.

```
PURge macroname
```

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.

```
PUT [target [fn [ft [fm]]]]
    [1      [=  [=  [= ]]]]
```

<table>
<tr><th>Description</th><th>Format</th></tr>
<tr><td>XEDIT (cont.)
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.</td><td>PUTD [target|1 [fn|= [ft|= [fm|=]]]]</td></tr>
<tr><td>Displays the current setting of various editing options.</td><td>Query [option]

options (specify only one each time):
APL
ARBchar
AUtosave
CASE
CMDline
COLPtr
COLumn
CTLchar[char]
CURLine
CURSor
EOF
ESCape
FILler
FMode
FName
FType
HEX
IMage
IMPcmscp
LASTmsg
LENgth
LIne
LINENd
LRecl
LScreen
MACRO
MASK
MSGMode
NBFile
NONDisp
NUMber
NUMber
PACK
PF[n]
Point[*]
PREfix
RANge
RECFm
RESERved
RING
SCALe
SCReen
Seq8
SERial
SIDcode
SIZe
SPAN
STAY
STReam
SYNonym[*|name]
TABLine
TABS
TARGet
TERMinal
TEXT
TOF
TOFEOF
TRunc
UPDate
VARblank
Verify
VERshift
Width
WRap
Zone
=</td></tr>
</table>

XEDIT (cont.)	
Terminates the editing session and leaves the previous copy intact.	QUIT [n]
Places data from the terminal into the console stack (LIFO). This subcommand generally is issued from a macro.	READ [Cmdline | All [NUMBER] | Nochange [NUMBER]] [TAG | NOTAG]
Replaces a specified number of lines removed by a DELETE or PUTD subcommand or a D (delete) prefix subcommand.	RECover [n|*|1]
Renumbers the line numbers of VSBASIC and/or FREEFORT files.	RENum [start-number|10[increment]]
Advances the line pointer and executes the last subcommand entered.	REPEat [target | 1]
Replaces the current line with a specified line or keyed in text or deletes the current line and enters input mode.	Replace [text]

Description	Format
XEDIT (cont.)	
Removes all prefix subcommands when the screen is in a "pending" or "incomplete" status.	RESet
Restores the settings of the XEDIT variables to the values in effect when last the PRESERVE subcommand was issued.	RESTore
Allows viewing of data in columns not currently visible on the screen. These columns are to the right of the rightmost column on the screen.	RIght [n|<u>1</u>]
Enters the file that is currently being edited onto disk without returning control to CMS.	SAVE [fn|= [ft|= [fm|=]]]
Locates every occurrence of a string and changes the string only when specified to do so. This is a macro.	SCHANGE [keynumber]
Changes the settings of various editing options while editing is in progress.	SET option

<table>
<tr><th>XEDIT (cont.)</th><th>options (must specify one):</th></tr>
<tr><td>Shows whether keys are available.</td><td>[SET] APL ON|OFF</td></tr>
<tr><td>Defines an arbitrary character used in a target definition. Note that the initial setting is OFF $.</td><td>[SET] ARBchar ON|OFF [character]</td></tr>
<tr><td>Sets or resets the automatic save function of the editor. Note that the initial setting is OFF.</td><td>[SET] AUtosave n|OFF[mode|A]</td></tr>
<tr><td>Controls letters entered and specifies significance in target searches.</td><td>[SET] CASE Uppercase|Mixed[Respect|Ignore]</td></tr>
<tr><td>Specifies the position of the command line on the screen.</td><td>[SET] CMDline On|Top|Bottom</td></tr>
<tr><td>Determines (on typewriter terminals) whether or not the column pointer (underscore) is displayed.</td><td>[SET] COLPtr ON|OFF</td></tr>
<tr><td>Define control character.</td><td>[SET] CTLchar char Escape
OFF
Protect[High|Nohigh|Invisible]
Noprotect[High|Nohigh|Invisible]
OFF</td></tr>
</table>

Description	Format
XEDIT (cont.)	
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] CURLine ON n
Allows entry of subcommand (on typewriter terminals) when in input mode without leaving input mode.	[SET] ESCape ON|OFF [character]
Defines a character to be used when expanding a line (see EXPAND subcommand).	[SET] FILler [character]
Changes the filemode of the edited file.	[SET] FMode fm
Changes the filename of the edited file.	[SET] FName fn
Changes the filetype of the edited file.	[SET] FType ft
Allows subcommand operands and targets to be specified in hexadecimal. Note that the initial setting is OFF.	[SET] HEX ON|OFF

XEDIT (cont.)	
Determines how tab characters (X'05') and backspace characters (X'16') are handled.	[SET] IMage ON|OFF|CANON
Determines whether or not non-XEDIT recognized subcommands are transmitted implicitly to CMS and later to CP for execution.	[SET] IMPcmscp ON|OFF
Determines whether or not # (pound sign) or other character is used as the line end character.	[SET] LINENd ON|OFF [character]
Defines a new logical record length for writing file to disk.	[SET] LRecl n|*
Controls sequence of editor's search for subcommands and macros. Note that the initial setting is OFF.	[SET] MACRO ON|OFF
Changes contents of mask. Note that the initial setting is a blank line.	[SET] MASK Define Immed [text] Modify

Description	Format
XEDIT (cont.)	
Controls message display. Note that the initial setting is ON LONG.	[SET] MSGMode ON [Short|Long] OFF
Defines a character to use in place of a nondisplayable character.	[SET] NONDisp [character]
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] NULls ON|OFF
Determines whether or not line numbers are displayed in prefix area. Note that the initial setting is OFF.	[SET] NUMber ON|OFF
Specifies whether or not packed file is entered on disk.	[SET] PACK ON|OFF
Defines or removes a meaning for a specified program function (PF) key. Note that TABKEY is the intitial setting of the PF4 key.	[SET] PFn [string TABKEY COPYKEY]

XEDIT (cont.)	
Defines or redefines the symbolic name for the current line.	[SET] Point .symbol [OFF]
Controls display of the prefix area. Also defines a synonym for a prefix subcommand.	[SET] PREfix ON [Left|Right]|OFF PREfix Synonym new old
Defines new limits for line pointer movement.	[SET] RANge target1 target2
Defines the record format for the file.	[SET] RECFm F|V|FP|VP
Reserves a specific line on the screen for displaying blank or specified information, which can be highlighted.	[SET] RESERved[+|-]n High[text] Nohigh[text] OFF
Displays a scale line under the current line (the default) or on a specified line.	[SET] SCALe ON[n]|OFF

Description	Format
XEDIT (cont.)	
Divides the screen into a specified number of logical screens to allow editing of multiple files.	[SET] SCReen n Size n1 [n2] ...
Controls file serialization.	[SET] SERial ON[incrno|10[startno|10]] ALL[incrno|1000[startno|1000]] string[incrno|10[startno|10]] OFF
Specifies whether a target-search character string must be included in one line or span a certain number of lines. Note that the initial setting is OFF Blank 2.	[SET] SPAN ON [Blank|Noblank[n|*]] OFF

XEDIT (cont.)

Specifies whether or not the line pointer moves when target-search object is not found. Note that the initial setting is OFF.	[SET] STAY 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] STRream ON|OFF
Specifies whether or not to look for synonyms. Also assigns a synonym to any existing subcommand or macro and defines an abbreviation for it. Also rearranges sequence of operand of subcommand the sequence of which is different than that of XEDIT subcommand operands.	[SET] SYNonym ON|OFF SYNonym newname[n]oldname SYNonym newname[n[form1...formn]]oldname[&1...&n]
Displays a "T" in every tab column according to current tab settings.	[SET] TABLine ON[n] OFF

<table>
<tr><th>Description</th><th>Format</th></tr>
<tr><td>XEDIT (cont.)</td><td></td></tr>
<tr><td>Defines the logical tab stops for a file.</td><td>[SET] TABS n1[n2 ... n28]</td></tr>
<tr><td>Specifies whether a terminal is to be used in line mode or in full screen mode.</td><td>[SET] TERMinal Typewriter
Display</td></tr>
<tr><td>Shows whether keys are available.</td><td>[SET] TEXT ON|OFF</td></tr>
<tr><td>Controls the display of TOP OF FILE, END OF FILE, TOP OR RANGE, and END OF RANGE null lines. Note that the initial setting is ON.</td><td>[SET] TOFEOF ON|OFF</td></tr>
<tr><td>Defines last column in which data may be entered.</td><td>[SET] TRunc n|*</td></tr>
<tr><td>Controls whether or not the number of blanks between two words is significant in target search. Note that the initial setting is OFF.</td><td>[SET] VARblank ON|OFF</td></tr>
</table>

XEDIT (cont.)

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] Verify [ON|OFF][[H]c1 c2]...
Controls use of wraparound. Note that the initial setting is OFF.	[SET] WRap ON|OFF
Defines starting and ending column of each record for target search scanning.	[SET] Zone zone1 zone2|*
Inserts specified string into the equal buffer (see = subcommand).	[SET] = string
Moves data either to the left or to the right. Note:. Data loss is possible.	SHift Left|Right[cols|1[target|1]]

Description	Format
XEDIT (cont.)	
Arranges a specified number of file lines in ascending or descending EBCDIC sequence according to specified sort columns.	SORT target [A|D] c1 c2 [c1 c2]...
Provides a set of functions used mainly in XEDIT macros or assigned to PF keys.	SOS option options: ALarm NField n Right n BField n NLine n TABB n CLEAR NUlls TABCmd Down n NUlls On|OFF TABCMDB Left n PFn TABCMDF LINEAdd POP TABF n LINEDel PUsh Up n
Splits a line into two or more lines at the column pointer or at the cursor. This is a macro.	SPlit [Column | CURSOR] [colno...] [[Before|After]/string/...]

XEDIT (cont.)

Places part or all of a specified number of lines into the console stack, starting with the current line.

```
STack ┌target ┌startcol ┌length┐┐┐
      │1      │1        │*     │││
      │       └         └      ┘┘│
      └                          ┘
```

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.

```
STATus [filename]
```

Moves the line pointer to the null line above the first line of the file or of the range (see SET RANGE).

```
TOP
```

Access, within a macro, specified editing variables and places their values in the console stack for subsequent reading by the EXEC 2 &READ control statements.

```
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
```

Description	Format
XEDIT (cont.)	TRAnsfer keyword (cont.) COLumn CTLchar[char] CURLine CURSor EOF ESCape FILler FMode FName FType HEX LRecl LScreen MACRO MASK MSGMode NBFile NONDisp NULls NUNber PACK SCALe SCReen Seq8 SERial SIDcode SIZe SPAN STAY STReam SYNonym[name] TOFFEO Trunc UPDate VARblank Verify VERShift Widtht WRap Zone =
Displays a specified number of lines, starting with the current line.	Type [target|1]
Moves the line pointer a specified number of lines toward the top of the file.	Up [n|*|1]
Translates all lowercase characters to uppercase ones, starting at the current line.	UPPercas [target|1]
Edits multiple files in virtual storage.	Xedit [fn [ft [fm]]] [(options...[)]] options: Same as the command options (see XEDIT command above).

XEDIT (cont.)	The XEDIT Prefix subcommands (line subcommands) are used as follows:	
	<u>Prefix</u>	<u>Meaning and/or Action</u>
Adds one or more lines immediately after the line in which subcommand is entered.	A	Add one line
	nA	Add n lines
	An	Add n lines
Copies one or more lines and must have the F (following) or P (preceding) prefix subcommand to indicate the destination of the lines.	C	Copy one line
	Cn	Copy n lines
	nC	Copy n lines
	CC	Copy block of lines
Deletes one or more lines.	D	Deletes one line
	Dn	Delete n lines
	nD	Delete n lines
	DD	Delete block of lines
Extends a logical line by one more physical line.	E	Extend logical line
Identifies the line <u>after</u> which copied or moved data (using the C or M prefix subcommands) is to be placed.	F	Data is entered following this point

Description	Format	
XEDIT (cont.)		
Inserts one or more lines immediately following the line in which prefix subcommand is entered.	I nI In	Insert one line Insert n lines Insert n lines
Moves one or more lines from one location to another in the file.	M Mn nM MM	Move one line Move n lines Move n lines Move block of lines
Identifies the line <u>before</u> which copied or moved data (using the C or M prefix subcommands) is to be placed.	P	Data is entered preceding this point.
Displays the scale on the line.	SCALE	Scale becomes visible
Displays every tab column in the line.	TABL	The letter "T" shows where a tab is coded
Sets the current line or identifies a line that will be the new current line.	/[n] [n]/	Make this line current and move the column pointer under the nth column.
Duplicates one line or a block of lines. Must be a double quote symbol.	" "n n" "" ""n n""	Duplicates one line Duplicate line n times Duplicate line n times Duplicate block of lines Duplicate block n times Duplicate block n times

```
XEDIT (cont.)                                 |
    Assigns xxxx as symbolic name             |   .xxxx    Four-character symbol
    to this line.                             |
                                              |
ZAP                                       CMS |     (MODULE  )
  Modifies or dumps MODULE,                   |ZAP {LOADLIB  } [libname1 ... libname3][(options...[)]]
  LOADLIB, or TXTLIB files.                   |     (TXTLIB  )
                                              |options:
                                              |[TERM|INPUT filename] [PRINT|NOPRINT]
                                              |
                                              |Control Statements:
                                              | BASE   address
                                              |
                                              | DUMP {membername} [csectname [startaddress [endaddress]]]
                                              |      {modulename} [ALL                                   ]
                                              |
                                              | LOG {fixname} [ZAPLOG|filetype [userdata]]
                                              |
                                              | NAME {membername|modulename} [csectname]
                                              |
                                              | REP   disp  data
                                              |
                                              | {VERIFY|VER}  disp   data
                                              |
                                              | * comment
                                              |
                                              | END
```

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 40 ARMONK, N.Y.

POSTAGE WILL BE PAID BY ADDRESSEE:

International Business Machines Corporation
Department G60
P. O. Box 6
Endicott, New York 13760

IBM Virtual Machine/System Product: Quick Guide for Users — Reference Summary

Order No. SX20-4400-1

Your suggestions help us produce better publications. We would appreciate any comments you have about the clarity, accuracy, and especially the usability of this reference summary.

If requesting a reply, please fill out the information below:

Name ______________________

Address ______________________

SX20-4400-1

IBM

Printed in U.S.A.