GX20-1926-1

IBM Reference Summary

IBM Virtual Machine Facility/370 Quick Guide for Users

IBM Corporation, VM/370 Publications,
24 New England Executive Pk., Burlington, Mass. 01803

GX20-1926-1

IBM Reference Summary

IBM Virtual Machine Facility/370 Quick Guide for Users

This publication describes the essential VM/370 operations for the new user. It also provides a brief description of all VM/370 commands for the experienced user. Only a limited amount of prior VM/370 knowledge is assumed for the section on VM/370 operations. See the "Preface" for prerequisite publications. The user of the command description section should have a thorough understanding of VM/370 command syntax and usage.

Second Edition (April 1973)

This edition is a major revision of GX20-1926-0 and makes that edition obsolete. The specifications in this publication correspond to Release 1 PLC 5 (Program Level Change) of the IBM Virtual Machine Facility/370, and to all subsequent modifications unless otherwise indicated in new editions or Technical Newsletters.

Changes are periodically made to the specifications herein; before using this publication in connection with the operation of IBM systems, refer to the latest _IBM System/360 and System/370 Bibliography_, GA22-6822, and the _IBM System/370 Advanced Function Bibliography_, GC20-1763, for the editions that are applicable and current.

Significant technical and other changes have been made throughout this publication; therefore, the user should read it in its entirety.

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 Corporation, VM/370 Publications, 24 New England Executive Park, Burlington, Massachussetts, 01803.

PREFACE

This publication contains information for both the beginning and the advanced VM/370 user. The first four sections are intended for the new user and the last section, including a reference card, is intended for the experienced user.

The first two sections, "What You Should Know Before You Start Using the VM/370 System" and "VM/370 System Information", are designed to help the new VM/370 user become acquainted with the system. These sections contain information for getting started and setting up a virtual machine. The third section, "Using CMS", discusses using the CMS facilities as a file creation and maintenance tool.

The fourth section, "Functions of VM/370 Commands", summarizes the commands by their intended use (for example, debugging and file maintenance commands). A new VM/370 user should find these function summaries a helpful and time-saving tool.

The last section, "Summary of VM/370 Commands", is a reference card intended for the experienced VM/370 user. The reference card contains a brief syntactic description and explanation of each VM/370 command.

Information in this publication (if any) about the following is for planning purposes only:

- The CMS Batch Facility.
- The IBM System/370 Models 165 II and 168.

The new user should use some of the VM/370 publications in conjunction with the first four sections of this Quick Reference Guide.

Prerequisite Publications

- IBM Virtual Machine Facility/370: Introduction, GC20-1800
- IBM Virtual Machine Facility/370: Command Language User's Guide, GC20-1804
- IBM Virtual Machine Facility/370: EDIT Guide, GC20-1805

Corequisite Publications

- IBM Virtual Machine Facility/370: Assembler Programmer's Guide, GC20-1802
- IBM Virtual Machine Facility/370: System Messages, GC20-1808
- IBM Virtual Machine Facility/370: Terminal User's Guide, GC20-1810

The experienced user should use the following publications in conjunction with the "Summary of VM/370 Commands" section.

Corequisite Publications

- IBM Virtual Machine Facility/370: Planning and System Generation Guide, GC20-1801
- IBM Virtual Machine Facility/370: BASIC Language Reference Manual, GC20-1803
- IBM Virtual Machine Facility/370: Command Language User's Guide, GC20-1804
- IBM Virtual Machine Facility/370: EDIT Guide, GC20-1805
- IBM Virtual Machine Facility/370: Programmer's Guide to Debugging, GC20-1807

CONTENTS

FIGURES

WHAT YOU SHOULD KNOW BEFORE YOU START USING THE VM/370 SYSTEM

The environment of the IBM Virtual Machine Facility/370 (VM/370) is one of virtual machines. A virtual machine is a functional simulation of a real computer and its I/O devices. VM/370 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 CPU, and virtual channels and I/O devices. However, since the virtual machines are simulated, their configurations may differ from each other and from the real machine. 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.

Regardless of the configuration, you control your virtual machine from your terminal, which is, effectively, your operator's console. The work to be done by the virtual machine is scheduled and controlled by an operating system that can run under VM/370. An example of a virtual machine operating system is the Conversational Monitor System (CMS), which was specifically designed to run in a virtual machine under control of VM/370. CMS provides, at a remote terminal, a full range of conversational capabilities: creation and management of files; compilation, testing, and execution of problem programs; and execution of application programs. The "Using CMS" section of this Quick Reference Guide describes how you can use a CMS virtual machine under VM/370.

Before you can start using VM/370 you must have:

- A user identification and password.
- A virtual machine defined for your use. (The virtual machine definition should include all the devices you will need. For example, a console, spooled unit record devices, and disk space.)
- Properly formatted disk space. (If you wish, you may format your disk space after you log on.)

GETTING STARTED (IDENTIFICATION AND PASSWORD)

Before you can use VM/370, you must be assigned:

- A user identification (userid) that identifies you to the system, and
- A password that is checked when you log in.

(Examples in this book use a userid of PUBS.) Assignment of a userid and password is normally handled (and approved) by the VM/370 system operations manager.

Once you have your userid and password, you can communicate with the VM/370 system from a remote terminal such as an IBM 2741 Communications Terminal or IBM 1050 Data Communication System (or equivalent). Depending on your terminal installation, you either dial the central VM/370 computer, or are connected directly. Refer to the _IBM Virtual Machine Facility/370: Terminal User's Guide_, GC20-1810, for a description of the communication procedures for each type of terminal.

VIRTUAL MACHINE CONFIGURATION

When you have a virtual machine defined for you, an entry is made in the Control Program directory. The systems operation group usually sets up your directory for you. This directory lists the devices and device addresses available to your virtual machine. The following is an example of a typical CMS virtual machine configuration.

Virtual Device	Virtual Device Address
console	009
card reader	00C
card punch	00D
printer	00E
CMS system disk	190
primary disk for user files	191
CMS system disk extension	19E

FORMATTING DISK SPACE

All disk space must be formatted for use with CMS. The systems operations group usually makes sure your disk space is formatted. The CMS Format program is executed under CMS via the FORMAT command. There is an example of the use of the CMS FORMAT command in the "Using CMS" section of this manual.

VM/370 SYSTEM INFORMATION

TYPING CONVENTIONS

Because certain special characters can be assigned logical editing functions to enter input data via VM/370 terminals, the following typing conventions should be observed. Input data may be entered in either upper or lower case. The examples in this book show the lines you might enter in shading. System responses may be in upper or lower case.

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

- Line Delete symbol (¢):
  The line delete symbol deletes all characters in the current logical line and itself. A line delete symbol cannot be deleted by a character delete symbol.

- Line End symbol (#):
  The line end symbol indicates the end of a logical input line. Use of this character permits more than one logical input line to be entered in the same real input string.

- Logical Escape symbol ("):
  The logical escape symbol causes the character following it to be interpreted as a data character (that is, ignored as an input line editing character). This allows any of the line editing characters to be interpreted literally.

For example, consider how the following line must be entered into the system:

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

Under the VM/370 input conventions, this line could not be entered as shown, since it would be affected by the #, @, and ¢ line editing symbols. For example, the # symbol would end the line. However, the line is correctly interpreted if entered as follows:

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

The logical escape characters (") are not put in the file.

- Line Length:
  For CP console functions and CMS commands an input line may be up to 130 characters long (including the carriage return character). Any line longer than 130 characters, including blanks, backspaces, underscores, the line editing characters, and the tab character, is truncated to 130 characters.

- Line Termination:
  An input data line from an IBM 2741 Communications Terminal is transmitted to the computer by pressing the Return key. Other terminals have similar line termination keys.

  Note: For some terminals, such as the IBM 1050, you have to press an Alternate Coding key and some other multiple-function key at the same time.

TERMINAL OPERATING PROCEDURES

Refer to the *IBM Virtual Machine Facility/370: Terminal User's Guide*, GC20-1810, for a description of the various terminal operating procedures.

LOGGING ON

When you establish communication with the VM/370 computer, the system sends the 2741 terminal one of these messages:

```
   vm/370 online    xxxxxx xxxxxx

              -- or --

   xxxxxx xxxxxx    vm/370 online
```

The top line is typed if the terminal is a 2741 equipped with a PTTC/EBCD character set. The bottom line is typed if the 2741 terminal has a standard Selectric (or Correspondence) character set. In either case, the 'xxxxxx xxxxxx' portion of the message consists of meaningless characters, and should be ignored.

Press the Attention (ATTN) key (or equivalent) once to unlock the terminal keyboard, and identify yourself by entering your user identification (userid) as follows:

```
   login pubs
```

Then press the Return key. If the userid entered is not found in the CP directory, the message

```
   DMKLOG053E  userid NOT IN CP DIRECTORY
```

types at the terminal, and you can try to enter another userid.

If the userid entered is found in the CP directory, the VM/370 system responds with:

```
   ENTER PASSWORD:
```

At this point, you should type in your password, followed by a carriage return (press the Return key). If your terminal does not inhibit printing of the password, you can press the Return key (or equivalent) to request VM/370 to print a mask string, as follows:

```
   88888888
```

The system then waits for you to type in your password over this mask. If the password entered is incorrect, the message

```
   DMKLOG050E  PASSWORD INCORRECT
```

types at the terminal. You must start the login procedure from the beginning by entering your userid again. If you do not do this, the message:

```
   restart    xxxxxxx

           -- or --

   xxxxxxx    restart
```

types at the terminal. The top line is what you would see at a non-correspondence terminal and the bottom line is what you would see at a correspondence terminal. You must start the login process from the beginning.

If the userid and password entered are valid, but someone else has already logged on with this userid, the VM/370 system issues the message:

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

where nnn indicates the line 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 or try again later.

Once you have successfully logged on, the VM/370 system replies with a log message, such as:

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

A logon message from the VM/370 operator (if any) also prints at this time.

Once you have successfully logged on the VM/370 system, you can start using the virtual machine that you have set up for your userid.

LOGGING OFF

When you are finished using the system and want to end your terminal session, you do so by logging out from the VM/370 Control Program (CP). Even if you are in CMS mode, you need only type the command:

```
   logout
```

and press the Return key. The system responds with:

```
   CONNECT= 00:11:43   VIRTCPU= 000:05.21
      TOTCPU= 000:21.03
   LOGOFF AT 11:34:44 EST THURSDAY 11/30/72
```

and the connection with the VM/370 system is terminated. The connect time is in hours, minutes, and seconds. The virtual CPU and total CPU times are in minutes, seconds, and hundredths of a second. When the logout procedure is completed, you can turn the terminal power off. Or, you may log out by turning the terminal power off.

Note: If you had logged in on a dialed line, you could specify that the communication line be left connected, by issuing

```
    logout hold
```

When you issue LOGOUT HOLD, you do not have to dial the line before logging on again.

VM/370 ENVIRONMENT

Each input line typed at the terminal by a user is transmitted to the VM/370 system, where it is processed (examined, and accepted or rejected) by a given routine. The portion of VM/370 that has control at the time a particular input line is entered determines which routine processes the input. Each portion of the VM/370 system which can accept input constitutes a unique environment, and only a subset of all possible input is acceptable to any given environment. Figure 1 shows the environments of the VM/370 system and shows how each environment is entered.

There are three input-processing environments:

- Control Program (CP environment)

- Central CMS service routines (CMS command environment)

- CMS command environments (DEBUG, EDIT, or a user-written command)

Input lines that are acceptable to the CP and CMS environments are referred to as commands.

Certain CMS commands cause CMS subenvironments to be entered. Examples of these are the DEBUG and EDIT commands. Lines acceptable to the environments of these commands are referred to as "subcommands," or merely "input," depending on the particular mode which is entered when the command is issued.

If the EDIT command is entered for a file that already exists, the EDIT environment is entered, allowing the contents of the existing file to be examined, added to, deleted, changed, or rearranged. You may cause the input environment to be entered so that you can add to the existing file.

The CMS EDIT subenvironment can exit to another subenvironment called Input Mode. If an EDIT command is issued for a file that does not currently exist, the EDIT environment is entered, and you may cause the INPUT environment to be entered so that you may directly create a file on disk, thus eliminating the keypunch step. In INPUT mode lines of data typed at the terminal become a part of the file being created. Because no check is made to determine the acceptability of input to this environment, lines which are keyed in are termed "input."

The ECHO environment is entered when the CP command ECHO is keyed in. All data lines entered in the ECHO environment are transmitted unchanged back to the terminal from which they were received.

Press the RETURN key (or its equivalent) with no characters entered, to determine which environment you are in.

You can take various actions to cause control to pass from one environment to another. By pressing the Attention key (or its equivalent) twice, quickly, you can always transfer control to the CP environment from any of the other environments. These actions are discussed throughout the *IBM VM/370: Command Language User's Guide*, GC20-1804.

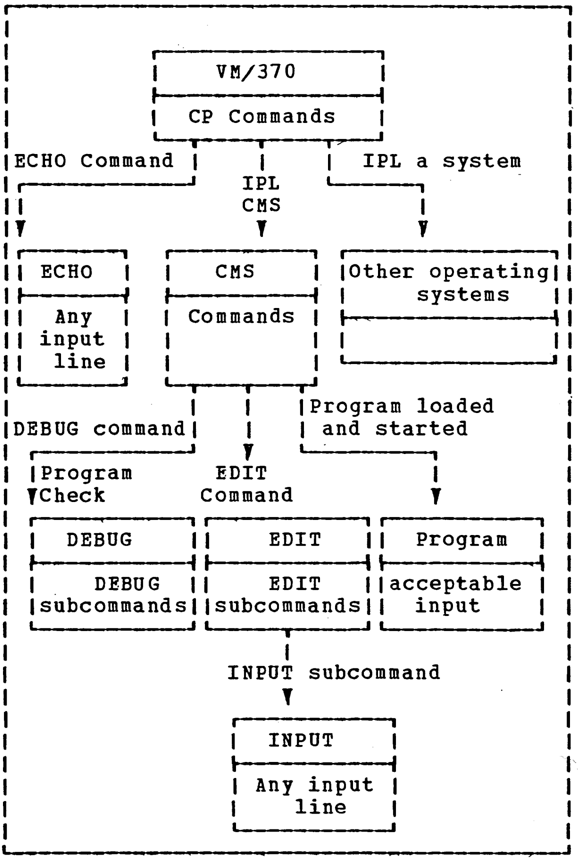

Figure 1. VM/370 Environments

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 directory entry with the devices needed for a CMS user.
- 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 "VM/370 System Information" section. The IPL and disk formatting procedures are described in the following section.

HOW TO IPL CMS

After you have logged on the VM/370 system, you can IPL an operating system. The format of the CP IPL command is

```
┌──────────────────────────────────────────────────┐
|   Ipl     |   {                     ┌        ┐ } |
|           |   { vaddr [cylno] |CLear   | } |
|           |   { systemname    |NOCLear | } |
|           |   {                     └        ┘ } |
└──────────────────────────────────────────────────┘
```

where:

vaddr	specifies the virtual address (ccu) of the device which contains the nucleus of the system to be loaded.
cylno	if this operand is specified, CP loads the IPL data from the specified virtual cylinder instead of from the default, which is virtual cylinder zero.
systemname	specifies that a copy of the named system has been previously saved by using the CP SAVESYS command, and is to be brought into virtual storage and given control.
CLEAR	specifies that the virtual storage space is to be cleared to binary zeros before the system is loaded.
NOCLEAR	specifies that the virtual storage space is not to be cleared to binary zeros before the system is loaded.

Assume that CMS is the _systemname_ of your CMS operating system and that 190 is the CMS system disk. You can IPL this CMS system with either of the following commands:

```
ipl cms
ipl 190
```

FORMATTING YOUR MINIDISK SPACE

Before you can use CMS in your virtual machine, you must have disk space which has been formatted for use by CMS. Usually the system operator provides formatted disk space for a new user. However, you can format your own disk space. This formatting procedure is performed only when new disk space is being initialized for your virtual machine; it should not be done each time you log on to the system. Formatting a disk destroys the contents of that disk.

The disk space for userid PUBS is defined in the CP directory as minidisk 191. This virtual disk space is the PUBS A-disk (or primary user disk). If you attempt to use CMS before formatting your A-disk, an error message is issued. For example, assume that you (with userid PUBS) have logged on and now want to IPL CMS in your virtual machine, but your A-disk (virtual address 191) was never formatted. The terminal sheet would look like this:

```
ipl cms
CMS..VERSION 1.0 mm/dd/yy

(Press the Return key.)

Y (19E) R/O.
DMSACC112S 'A (191) ' DEVICE ERROR.
R; T=0.01/0.07 11:25:17
```

The "Y (19E) R/O." message tells you that the CMS system you just loaded (via IPL) has a Y-disk at address 19E which is a read-only extension of the system disk (S-disk).

The "DEVICE ERROR" message indicates that your A-disk (in this example, 191) was not correctly formatted prior to use.

To execute the CMS Format program, you must type:

```
format 191 a
```

where 191 indicates the virtual disk address, and "a" indicates that it is the A-disk. The Format program then issues prompting messages to which you must reply:

```
DMSFOR603R FORMAT WILL ERASE ALL FILES
   ON DISK 'A(191)'. DO YOU WISH TO
   CONTINUE? (YES|NO):
yes
DMSFOR605R ENTER DISK LABEL:
pubs01
FORMATTING DISK 'A'.
'3' CYLINDERS FORMATTED ON 'A(191)'.
R; T=0.15/1.60 11:26:03
```

If any files existed, they are erased. The disk space, that contains 3 cylinders, is labeled PUBS01. When your PUBS A-disk is formatted and the CMS virtual machine is operating, you can use CMS to do some further setup work.

If you know your disk is not formatted at the time you IPL, enter the commands:

```
ipl cms
access (nodisk
```

before pressing the Return key. The error message, DMKACC112S, will not type. You should then issue the command

```
format 191 a
```

to format your A-disk.

WRITING A PROFILE EXEC

Although you can use CMS without a PROFILE EXEC, it is often convenient to use one. The PROFILE EXEC is a special EXEC procedure that is executed as the first command after you IPL CMS. If you want to use the System Assembler to assemble programs under CMS, it is a good idea to include the CMS and OS macro library in your PROFILE EXEC definition. You can do this by putting the appropriate GLOBAL command in your PROFILE EXEC. Other additions for your PROFILE EXEC might be:

- The short form of the "Ready" message (R;).
- A blip character of "*" to indicate seconds of virtual CPU time.

You create your PROFILE EXEC by using the facilities of the CMS EDIT command. The EDIT command is fully described in the <u>IBM Virtual Machine Facility/370: EDIT Guide</u>, GC20-1805. Only the EDIT subcommands used to create your PROFILE EXEC are included here. Your PROFILE EXEC for userid PUBS may be created by issuing the EDIT command with the filename and filetype of "PROFILE EXEC". The Edit program will not find the file you specified and will return the message "NEW FILE" and enter the edit mode. You should type "input". When the edit program responds with "INPUT:", you can start entering the statements of your PROFILE EXEC file. Refer to the <u>Edit Guide</u> for a description of these subcommands. The entire terminal listing would appear as follows:

```
edit profile exec
NEW FILE:
EDIT:
input
INPUT:
&control off
set rdymsg smsg
global maclib cmslib osmacro
set blip * (1)
```

(Press the Return key to leave INPUT mode.)

```
EDIT:
file
R; T=0.21/0.84 11:31:37
```

Now that your PROFILE EXEC has been created and filed, you can verify that it contains the desired commands by requesting a typeout of it at the terminal:

```
type profile exec

&CONTROL OFF
SET RDYMSG SMSG
GLOBAL MACLIB CMSLIB OSMACRO
SET BLIP * (1)

R; T=0.12/0.58 11:32:58
```

Note that the PROFILE EXEC does not execute immediately (the "Ready" message is still the long message). The PROFILE EXEC is not executed until the next time you issue IPL CMS or the next time you type "profile" during your terminal session.

EXAMPLE OF CMS PROGRAM DEVELOPMENT FACILITIES

This section illustrates several CMS functions that are useful in creating and manipulating CMS files. In addition, the flow of the primary example shows how CMS can be used to control program execution.

First IPL CMS. Note that, if you have followed the preceding instructions, the disk space is already formatted and no error message appears. Also, the short form of the Ready message types because your PROFILE EXEC file is in effect.

CREATING AN ASSEMBLER LANGUAGE SOURCE FILE

The program used as an illustration 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 Edit facility.

```
edit manip assemble
NEW FILE:
EDIT:
input
INPUT:
manip     csect
          print nogen
          save  (14,12),,*
          balr  12,0
          using *,12       establish addressability
          la    2,8(,1)    r2=addr of input file in plist
          la    3,32(,1)   r3=addr of output file in plist
* determine if input file exists
          fsstate (2),error=err1
* read a record from input file and write on output file
rd        fsread  (2),error=eof,buffer=buff1,bsize=80
          fswrite (3),error=err2,buffer=buff1,bsize=80
          b     rd         loop back for next record
* come here if error reading input file
eof       equ   *
          la    15,7       test code for read error
          c     15,=f'12'  end of file?
          bne   err3       error if not
          return (14,12),rc=0
* if input file does not exist
err1      wrterm 'file not found',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
buff1     ds    cl80
          end   manip
(Press the RETURN key to leave Input mode.)

EDIT:
file
R;
```

The Editor did not find a file with the filename and filetype of MANIP ASSEMBLE, so it created the file for you. You then issue INPUT and enter your program. You must issue the "FILE" subcommand in order to save your program.

Note that this program uses several CMS macros; when it is assembled, this program will require the CMS macro library. However, your PROFILE EXEC (for the PUBS user) has specified that the CMS macros be included; no further action is necessary to include the CMS macros.

The Load Address (LA) instruction following EOF is inserted only for testing; it will be deleted when the function has been tested.

ASSEMBLING A SOURCE FILE

To assemble the MANIP program, you enter the "ASSEMBLE MANIP" command, then wait for the System Assembler to complete processing:

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


 MAN00331              B      ERRT
 IEU024  NEAR OPERAND COLUMN   1--UNDEFINED SYMBOL

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

Each asterisk (*) on the second line indicates 2 seconds of virtual CPU time. The message IEU024 indicates there is an error in your program. The line in your program containing the error has a sequence number of MAN00331. Print your listing file to find this line.

At this point, there are 3 files associated with your program. First, the file, MANIP ASSEMBLE, contains the source statements of your program. This file was the input used by the Assembler Language program. The output from the System Assembler is 2 permanent files. One of these files, MANIP TEXT, contains the object module. The other file, MANIP LISTING, contains a listing of the source statements, assembled machine code, and other associated information based on the options selected on 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 to change or correct this Assembler file. When you issue the EDIT MANIP ASSEMBLE command this time, the Editor finds your file and enters the 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 sheet would look as follows:

```
edit manip assemble
EDIT:
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
*****i
 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 command name, MANIP, 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.

```
edit manip assemble
EDIT:
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
*****i
 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, type the first few lines of file MANIP1 to make sure that MANIP 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 ADDRESSABILIT

R;

erase manip1 assemble
R;
```

The LISTFILE 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 CLS 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. In addition, 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 for 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;
```

READING

When the user PAYROLL logs on the VM/370 system, the following message will type during the login procedure:

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

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;
```

Note, however, that the PAYROLL user can decide whether or not he wants the file before he reads it by invoking the command:

```
query reader all
FILE CLS RECDS ORIGIN DATE      TIME     NAME   YPE
029  A   00051 PUBS   11/30/72 13:29:03 MANIP   EXT
```

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
```

The Conversational Monitor System (CMS) can be used for many other purposes. Those functions illustrated in the previous discussion are intended to help the novice VM/370 user become acquainted with the system and its capabilities. Once you have become familiar with these commands and functions, you will have a sound base upon which to build a more thorough understanding of the VM/370 system.

The next section of the Quick Guide is a brief summary of the functions that can be performed by issuing a CP or CMS command. The last section of this Quick Guide briefly describes the function of each VM/370 command and presents its syntactic illustration. Experienced users may find the functional and command summaries a useful supplement to the more detailed information contained in the _IBM Virtual Machine Facility/370: Command Language User's Guide_, Order No. GC20-1804.

FUNCTIONS OF VM/370 COMMANDS

Figures 2 through 10 present a functional summary of commands available in the VM/370 system.

Function	Command		Type
Begin terminal session (identify user to VM/370 system).	LOGIN		CP
End terminal session.	LOGOUT		CP
Communicate with other VM/370 users and with the system operator.	MSG		CP
Connect a terminal to a multiaccess virtual machine.	DIAL		CP
Disconnect a user's terminal from a virtual machine.	DISCONN		CP
Test terminal hardware.	ECHO		CP
Control terminal input and output.			
• Indicate if accounting data is to be	SET	ACNT	CP

received at the terminal.			
• Indicate if messages from other users are to be received at the terminal.	SET	MSG	CP
	SET	WNG	CP
• Control line editing functions.	SET	LINEDIT	CP
• Control format of messages received at the terminal.	SET	REDType	CMS
	SET	EMSG	CP
• Get information about terminal control parameters.	QUERY	TERMINAL	CP
• Specify method of password entry.	TERMINAL	MASK	CP
• Specify use of additional translation tables.	TERMINAL	APL	CP
• Specify terminal line size.	TERMINAL	LINESIZE	CP
• Specify ATTN key handling procedures.	TERMINAL	ATTN	CP
• Specify that ATTN key will not stop the virtual machine.	SET	RUN	CP

Figure 2. Commands to Control Terminal Session (Part 1 of 2)

Function	Command	Type
• Specify characters to indicate CPU time interval reporting.	SET BLIP	CMS
• Specify format of CMS READY message.	SET RDYMSG	CMS
• Indicate character translations to be done during terminal input and output.	SET INPUT	CMS
	SET OUTPUT	CMS

Figure 2. Commands to Control Terminal Session (Part 2 of 2)

Function	Command	Type
Create a source program file from the terminal.	EDIT	CMS
Invoke the System Assembler to assemble a source program.	ASSEMBLE	CMS
Invoke the BASIC Compiler.	BASIC	CMS

Create or list macro libraries to be used during assemblies or compilations.	MACLIB		CMS
Create or list subroutine libraries.	TXTLIB		CMS
Specify macro libraries to be searched during assemblies or compliations.	GLOBAL	MACLIB	CMS
Specify subroutine libraries to be searched during LOAD and INCLUDE function.	GLOBAL	TXTLIB	CMS
Bring object code into storage.	LOAD INCLUDE		CMS CMS
Create a MODULE (core-image) file.	GENMOD		CMS
Print a storage map of a MODULE file.	MODMAP		CMS

Figure 3. Commands to Develop Programs and Process Data (Part 1 of 2)

Function	Command	Type
Bring MODULE files into storage.	LOADMOD	CMS
Build auxiliary module directories.	GENDIRT	CMS
Begin execution of programs which were previously loaded into main storage.	START	CMS
Simulate OS Data Definition (DD) JCL cards during program execution.	FILEDEF	CMS
Load and execute object (TEXT) files.	RUN	CMS
Compile, load, and execute source files.	RUN	CMS
Load and execute core-image (MODULE) files	RUN	CMS

Figure 3. Commands to Develop Programs and Process Data (Part 2 of 2)

Function	Command		Type
Stop execution at a specified virtual	ADSTOP		CP
machine location.	DEBUG	BREAK	CMS
Resume execution of a stopped	BEGIN		CP
virtual machine.	DEBUG	GO	CMS
Display virtual storage, registers,	DISPLAY		CP
PSW, etc.	DEBUG	GPR	CMS
	DEBUG	PSW	CMS
	DEBUG	X	CMS
Print the contents of virtual storage	DUMP		CMS
locations on the spooled printer.	DEBUG	DUMP	CMS

Figure 4. Commands to Test and Debug a Program (Part 1 of 2)

Function	Command	Type
Change the contents of registers and storage locations.	STORE	CP
	DEBUG STORE	CMS
	DEBUG SET	CMS
Trace virtual machine SVC calls.	SVCTRACE	CMS
Trace virtual machine instructions, I/O operations, SVC calls, etc.	TRACE	CP
Convert system ABEND dumps to printed output.	VDUMP	CMS

Figure 4. Commands to Test and Debug a Program (Part 2 of 2)

Function	Command	Type
Create a file from terminal input.	EDIT	CMS
Create a file from card input.	READCARD	CMS
	DISK LOAD	CMS
Verify the existence of a file on disk.	STATE	CMS
Erase a file (or files) from disk.	ERASE	CMS
	ACCESS ERASE	CMS
List names of files on disk and their attributes.	LISTFILE	CMS
Type the contents of a file at the terminal.	TYPE	CMS

Figure 5. Commands to Maintain Data Files (Part 1 of 3)

Function	Command	Type
Print the contents of a file or a member of a library on a spooled printer.	PRINT	CMS
Punch the contents of a disk file on a spooled punch.	PUNCH	CMS
	DISK DUMP	CMS
Sort the records of a file into ascending order based on specified sort fields.	SORT	CMS
Copy one disk file to another disk file.	MOVEFILE	CMS
	COPYFILE	CMS
Combine several files into one file.	COPYFILE	CMS
Copy data from one device type to another device type.	MOVEFILE	CMS
Change the name of a CMS file.	RENAME	CMS

Compress a file by encoding multiple contiguous occurrences of a single character.	COPYFILE	PACK	CMS
Rearrange the contents of records in a disk file.	COPYFILE		CMS
Perform character translations on specified characters in a disk file.	COPYFILE	TRANS	CMS
Append one file to the end of another file.	COPYFILE	APPEND	CMS
Remove trailing blanks from records in a file.	COPYFILE	TRUNC	CMS
Compare the contents of two disk files.	COMPARE		CMS

Figure 5. Commands to Maintain Data Files (Part 2 of 3)

Function	Command	Type
Change records in a file based on record sequence numbers.	UPDATE	CMS
Change attributes of a spooled file.	CHANGE	CP
Terminate spooling operations on a virtual unit record device.	CLOSE	CP
Load the virtual forms control buffer (FCB).	LOADVFCB	CP
Indicate order of processing for spooled files.	ORDER	CP
Remove spooled files from the system.	PURGE	CP
Print documents according to to control words in the document file.	SCRIPT	CMS

Change options in effect for spooling operations.	SPOOL		CP
Route spooled input files to another user.	TRANSFER		CP
Copy data from disk to tape.	TAPE	DUMP	CMS
Copy data from tape to disk.	TAPE	LOAD	CMS
Convert OS partitioned data set (PDS) files or card-image files on tape to CMS format on disk	TAPPDS		CMS

Figure 5. Commands To Maintain Data Files (Part 3 of 3)

Function	Command	Type
Logically connect a disk to a virtual machine.	LINK	CP
Logically disconnect a device from a virtual machine.	DETACH	CP
Make files on a disk available to a user.	ACCESS	CMS
Remove accessibility to files.	RELEASE	CMS
Dump a disk to tape, and restore disk from tape.	DDR DUMP DDR RESTORE	CMS CMS
Format disk space in CMS format.	FORMAT	CMS
Format real or virtual disk volumes.	MINIDASD	CMS

Figure 6. Commands for Disk Maintenance and Control

Function	Command	Type
Load a virtual machine operating system.	IPL	CP
Alter the virtual machine configuration.	DEFINE	CP
Disconnect user terminal from VM/370.	DISCONN	CP
Enter control program commands from a CMS virtual machine.	CP	CMS
Communicate with other virtual machine users or with the system operator.	MSG	CP
Simulate an external interrupt for a virtual machine.	EXTERNAL	CP
Simulate 'not ready' for a virtual device	NOTREADY	CP

Figure 7. Commands for Virtual Machine Control (Part 1 of 2)

Function	Command	Type
Simulate functions of buttons on the main computer console.	SYSTEM	CP
Place virtual machine console in dormant state with keyboard locked.	SLEEP	CP
Perform tape rewind action.	REWIND	CP
	TAPE REW	CMS
Establish User Directory Entries.	DIRECT	CMS
Simulate device end interrupt to a virtual machine device.	READY	CP
Reset virtual device's pending interrupts	RESET	CP

Define means of entering CMS command	SYNONYM		CMS
CMS command names	SET	ABBREV	CMS
	SET	IMPEX	CMS
	SET	IMPCP	CMS
Indicate if I/O is to be done as specified by the virtual machine with no CCW translation by CP.	SET	NOTRANS	CP
Control timer updating.	SET	TIMER	CP
Change or set the number of loader tables for a CMS virtual machine.	SET	LDRTBLS	CMS
Release pages of storage after certain CMS commands execute.	SET	RELPAGE	CMS
Get virtual machine status information.	QUERY		CP

Figure 7. Commands for Virtual Machine Control (Part 2 of 2)

Function	Command	Type
Create accounting records for logged on users and reset accounting data.	ACNT	CP
Logically connect or dedicate devices to a virtual machine or to CP.	ATTACH	CP
Logically disconnect devices from a virtual machine or from CP.	DETACH	CP
Disable communication lines.	DISABLE	CP
Enable communication lines.	ENABLE	CP
Force a specific user to log out from CP.	FORCE	CP
Terminate active channel CP program on a specific device.	HALT	CP

Control paging activity.	LOCK UNLOCK	CP CP
Request information about real and virtual machine characteristics.	QUERY	CP
Establish system parameters.	SET	CP
Terminate functions and checkpoint system.	SHUTDOWN	CP
Transmit high priority messages to users.	WNG	CP

Figure 8. Commands For VM/370 Control

Function	Command	Type
Restart or reposition the current output of an output spooling device.	BACKSPAC	CP
Change attributes of a closed spool file.	CHANGE	CP
Halt operations of specified spooling devices following completion of current activity.	DRAIN	CP
Cancel current output on a real unit record device.	FLUSH	CP
Cancel spool HOLD status.	FREE	CP
Defer spooled output of a particular user.	HOLD	CP
Load printer UCS or FCB buffer.	LOADBUF	CP

Cause spooled files to be processed in a specific order.	ORDER	CP
Remove closed spool files from the system.	PURGE	CP
Request information about spool files.	QUERY	CP
Repeat printing or punching of current file on a specific output device.	REPEAT	CP
Force a printer to single space output.	SPACE	CP
Start spooling device after draining or changing output class.	START	CP
Direct spooled files to a user's card reader.	TRANSFER	CP

Figure 9. Commands For Spooling Control

Function	Command	Type
Display real storage at terminal.	DCP	CP
Dump real storage to virtual spooled printer.	DMCP	CP
Find locations of control blocks.	LOCATE	CP
Save virtual machine storage space on disk.	SAVESYS	CP
Perform intensive recording of device activity information.	SET RECORD	CP

Set the error recording mode for soft machine checks.	SET MODE	CP
Dump error information which has been recorded by error recording routines.	CPEREP	CMS

Figure 10. Commands for System and Hardware Analysis

SUMMARY OF VM/370 COMMANDS

The following is a listing of all the CP and CMS commands. The CMS DEBUG subcommands, EXEC control statements and built-in functions, and EDIT commands are described first. Then, a reference card lists the CP and CMS commands; a brief description precedes a syntactic representation of each command.

The commands and subcommands are shown in upper and lower case; the upper case represents the minimum truncation of the command that the system accepts. Where options are given, the underscore indicates the default option. Options in brackets may be omitted. See the _IBM Virtual Machine Facility/370: Command Language User's Guide_, Order No. GC20-1804, for an explanation of other notational conventions.

DEBUG Command Description	Command/Subcommand Format
<u>DEBUG</u> CMS Enters the DEBUG environment to perform program analysis and repair.	DEBUG DEBUG environment entered; the formats of each DEBUG subcommand are as follows:
Stops program execution.	BReak {symbol|hexloc}
Types the Channel Address Word.	CAW
Types the Channel Status Word.	CSW
Assigns a symbolic name to a specific storage address.	DEFine symbol hexloc [n] [<u>4</u>]
Dumps the contents of storage locations to the virtual printer.	DUmp [symbol1] [symbol2] [ident] [hexloc] [hexloc2] [*]
Exits from the DEBUG environment.	GO [symbol] [hexloc]
Types the contents of the specified general registers.	GPR reg1 [reg2]
Returns to CMS environment.	HX

Sets a base address.	ORigin { symbol|hexloc }
Types the contents of the old Program Status Word.	PSW
Returns to CMS environment.	RETurn

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

Stores information in the specified virtual location.	STore { symbol hexloc } hexinfo
Indicates means of handling I/O.	TIN { CMS|DEB }
Examines virtual storage locations.	X { symbol hexloc } [n 4]

EXEC Command Description	Control Statements and Built-in Functions
<u>EX</u>EC CMS Invokes EXEC files.	EXec fn [args...] The formats of the EXEC control statements and built-in functions are as follows:
Defines or redefines arguments.	&ARGS [arg1 [arg2 ...]]
Punches the following lines into cards.	&BEGPUNCH [ALL]
Stacks the following lines into the terminal input buffer.	&BEGSTACK [LIFO] [ALL] [<u>FIFO</u>]
Types the following lines at the terminal.	&BEGTYPE [ALL]
Combines token1 and token2.	&CONCAT tok1 {tok2 ...}
Provides a branching address for EXEC branch statements.	&CONTINUE
Supplies the parameters for the execution phase of the EXEC file.	&CONTROL [OFF] [TIME] [PACK] [ERROR] [NOTIME] [NOPACK] [CMS] [ALL]

Allows the defined token to be known from this point on by its composition, that is, numeric or character data.	&DATATYPE tok
END statement for action started by &BEGPUNCH, &BEGSTACK or &BEGTYPE.	&END
Provides error return processing.	&ERROR action
Exits from the EXEC file with a given return code.	&EXIT [return-code | <u>0</u>]
Transfers control to a defined location.	&GOTO {TOP | line-number | label}
Allows statement execution if the comparison is satisfied.	&IF {tok1 | &$ | &*} {EQ | NE | LT | LE | GT | GE} {tok2 | &$ | &*} executable statement

EXEC Command Description	Control Statements and Built-in Functions
Indicates the number of non-blank characters in the following token.	&LENGTH tok
Allows the use of the literal value of the token.	&LITERAL tok
Repetitively executes a sequence of statements.	&LOOP { n label } [(m) (condition)]
Punches a card with the defined tokens.	&PUNCH tok1 [tok2 ...]
Reads the next line (or lines) from the terminal.	&READ [n ARGS VARS [var1 [var2 ...]] 1]
Skips subsequent statements.	&SKIP [n 1]
Types blank lines at the terminal.	&SPACE [n 1]

Places a line of tokens in the console stack.	&STACK [LIFO / <u>FIFO</u>] [tok1 [tok2 ...]
Extracts the desired string from the given token.	&SUBSTR tok i [j]
Types time information on the terminal.	&TIME [ON / OFF] [RESET / TYPE]
Prints the tokens at the terminal.	&TYPE tok1 [tok2 ...]

EDIT Command Description	Command/Subcommand Format
EDIT CMS Provides access to the EDIT environment.	Edit fn ft [fm] [(LRECL nn)] The EDIT subcommands are as follows:
Scans records, altering the specified character.	ALter {char1 hex1} {char2 hex2} [n * 1 [G *]]
Points to the last line of the file.	Bottom
Translates to uppercase.	CASE [U M]
Changes string1 to string2.	Change /string1/string2 [/[{n * 1}[G *]]]
Enters CMS subset command mode.	CMS
Deletes n lines or to EOF.	DELete [n|1|*]
Points to the nth line down from the current line.	DOwn [n|1]

Saves the file edited on disk and returns to CMS.	FILE fn [ft [fm]]]
Searches the file for the given line.	Find [line]
Resets or types the filemode.	FMode [fm]
Resets or types the filename.	FName [fn]
Inserts some or all of the given file.	Getfile fn $\left[\begin{matrix}\text{ft}\\ *\end{matrix}\left[\begin{matrix}\text{fm}\\ \underline{*}\end{matrix}\left[\begin{matrix}\text{m}\\ \underline{1}\end{matrix}\left[\begin{matrix}\text{n}\\ \underline{*}\end{matrix}\right]\right]\right]\right]$
Expands text into line images or displays current settings.	IMAGE $\left\{\begin{matrix}\text{ON}\\ \text{OFF}\\ \text{CANON}\end{matrix}\right\}$
Inserts 'line' in the file or enters Input mode.	Input [line]
Sets or types line numbering.	LINEmode [Left | Right | OFF]
Scans the file for the first occurrence of 'string'.	Locate /string[/]
Enters LONG error message mode.	LONG

EDIT Command Description	Command/Subcommand Format
Points to the nth line down from the current line.	Next [n|1]
Replaces all or part of the current line.	Overlay line
Saves current mode settings.	PREserve
Sets the line increment.	PROMPT [incr]
Terminates the EDIT session.	QUIT
Sets or displays record format.	RECfm [F|V]
Executes the following OVERLAY request n times.	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.	{REUSE} [edit subcommand] {=}
Saves the file on disk.	SAVE [fn [ft [fm]]]
Turns serialization on or off in columns 73-80.	SERial { OFF seq [incr] ON [10] ALL }
Enters SHORT error message mode	SHORT
Stacks n lines in the terminal input buffer.	STACK [n|1|edit subcommand]
Sets the given tabs	TABSet tabs...
Points to the beginning of the file.	TOP
Sets or displays the column of truncation.	TRUNC [n|*]
Types the specified number of lines beginning with the current line.	Type [m [n]] * * 1

EDIT Command Description	Command/Subcommand Format
Points to the line n lines above the current line.	Up [n|1]
Sets, displays, or resets verify mode.	Verify [ON OFF] [n *]
Assigns to X or Y the given EDIT subcommand.	X [edit subcommand|m|1] Y
Sets or displays the columns to be edited.	Zone [m [n]] * 1 [*]
Types the last EDIT subcommand.	?
Locates the line.	nnnnn [text]
Duplicates the current line.	$DUP [1|n]
Moves n lines up or down m lines.	$MOVE n {[UP m|DOWN m]|[TO label]}

Trim Along This Line

IBM Virtual Machine Facility/370: Quick Guide for Users — Reference Summary **Order No. GX20-1926-1**

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.

FIRST CLASS
PERMIT NO. 172
BURLINGTON, MASS.

BUSINESS REPLY MAIL

NO POSTAGE STAMP NECESSARY IF MAILED IN U.S.A.

POSTAGE WILL BE PAID BY

IBM CORPORATION
VM/370 Publications
24 New England Executive Park
Burlington, Massachusetts 01803

Command Description	Command Format
ACCESS — CMS Defines direct access space to a CMS virtual machine and relates it to a logical directory.	ACcess { ccu mode [/ext [fn [ft [fm]]]] [(NOPROF) | (ERASE) (NODISK)] }
ACNT — CP Class A Creates accounting records.	ACNT {userid1 userid2 . . . | ALL}
ADSTOP — CP Class G Halts the virtual machine's execution.	ADSTOP {hexloc | OFF}
ASSEMBLE — CMS Invokes the System Assembler.	Assemble fn [(options...)] options: [NOXREF | XREF] [RENT | NORENT] [DECK | NODECK] [NOLOAD | LOAD] [PRINT | NOPRINT | DISK] [DOS | OS] [TEST | NOTEST] [LINECNT nn | LINECNT 55] [NOTERM | TERM] [NONUM | NUM] [STMT | NOSTMT] [NOLIST | LIST]
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}
ATTACH CHANNEL — CP Class B Attaches a channel to a designated user.	ATTach CHANnel c [To] userid
BACKSPAC — CP Class D Restarts a current spool file.	Printer Format BAckspac raddr [File | pages | 1] Punch Format BAckspac raddr [File]
BASIC — CMS Invokes the VM/370 BASIC Language Processor.	BAsic fn [(LONG)]
BEGIN — CP Class G Starts the execution of a virtual machine.	Begin [hexloc]
CHANGE — CP Classes D,G Alters the attributes of a closed spool file.	CHange [userid] [SYSTEM] { Reader {Class a | spoolid | ALL} CLass b Printer | PUnch {CLass a | spoolid | ALL} {CLass b | COpy nn | [HOld | NOHold] | DIst distcode} } [NAme {fn [ft] | dsname}]
CLOSE — CP Class G Terminates spooling operations on a virtual reader, printer, or punch.	CLose { Reader | vaddr [HOld | NOHold] Printer | PUnch | vaddr [PUrge | [HOld | NOHold] [DIst distcode] [NAme {fn [ft] | dsname}]] }
COMPARE — CMS Compares all or part of records in two existing files.	COMpare fileid1 fileid2 [(COL mm-nn)]
COPYFILE — CMS Copies files according to operand specifications.	COPYfile fileid1 [fileid2...] [fileido] [(options)] options: [Type | NOType] [OLDDate | NEWDate] [RECfm F | RECfm V] [NOPrompt | PRompt] [TRAns] [UPcase | LOWcase] [From recno | FRLabel xxxxxxxx] [FOR recno | TOLabel xxxxxxxx] [REPlace | OVly | APpend | NEWFile] [Fill c | Fill hh | Fill 40] [TRUnc | NOTRunc] [PAck | UNPack] [EBcdic] [LRecl nn] [SPecs | NOSPecs]
COUPLE — CP Class G Connects virtual channel to channel adapters.	COUPLE vaddr1 [To] userid vaddr2
CP — CMS Permits entry of CP console functions from the CMS environment.	CP [line]
DCP — CP Class E Displays real processor storage on the terminal.	DCP {hexloc1 | Lhexloc1 | Thexloc1} [{- | :} [hexloc2 | END]]
DDR — CMS Dumps, restores, or copies data between DASD devices and tape devices.	DDR [fn ft [fm]]
DEBUG — CMS Enters the DEBUG environment to perform program analysis and repair.	DEBUG
DEFINE — CP Class G Reconfigures the user's virtual machine.	DEFine { Reader | Printer | PUnch | CONsole | CTCA | TIMer | 1403 | 3211 [As] vaddr LIne [As] vaddr [IBM[1] | TEL[E2]] vaddr1 [As] vaddr2 T2314 | T2319 | T3330 | T2305 [As] vaddr [CYL] nnn STORage [As] nnnnK }
DETACH — CP Class B Removes a real device from the system or from a specific user.	DETach raddr [From] {userid | SYSTEM}
DETACH — CP Class G Detaches a virtual device from the virtual machine.	DETach vaddr
DETACH CHANNEL — CP Class B Removes the specified channel and all its related devices from the specified user.	DETach CHANnel c [From] userid
DIAL — CP Class ALL Attaches a terminal device to a multiple access system.	Dial userid [vaddr]
DISABLE — CP Classes A,B Inhibits the use of communication lines.	DISAble {raddr... | ALL}
DISCONN — CP Class ALL Disconnects the terminal from virtual machine operation.	DISConn [HOld]
DISK — CMS Dumps and restores disk files.	DISK {DUMP fn ft [fm] | LOAD}
DISPLAY — CP Class G Displays storage locations and registers within the virtual machine.	Display { hexloc1 | Lhexloc1 | Thexloc1 | Khexloc1 [{- | :} [hexloc2 | END]] Greg1 | Yreg1 | Xreg1 [{- | :} reg2] Psw | CAW | CSW }
DMCP — CP Class E Dumps any area of System/370 real storage to a spool device.	DMCP {hexloc1 | Lhexloc1 | Thexloc1} [{- | :} [hexloc2 | END]] [dumpid]
DRAIN — CP Class D Stops spooling activity on the specific device after the current file is finished spooling.	DRain [Reader | Printer | PUnch | raddr... | ALL]
DUMP — CP Class G Dumps virtual machine registers and storage to the virtual printer.	DUmp {hexloc1 | Lhexloc1 | Thexloc1} [{- | :} [hexloc2 | END]] [dumpid]
ECHO — CP Class G Returns data directly to the terminal.	ECho [nn | 1]
EDIT — CMS Provides access to the EDIT environment.	Edit fn ft [fm] [(LRECL nn)]
ENABLE — CP Classes A,B Activates communication lines.	ENable {raddr... | ALL}
ERASE — CMS Deletes files from a user's disk	ERASE fn ft [fm] [(Type) | (Notype)]
EXEC — CMS Invokes EXEC files.	EXec fn [args...]
EXTERNAL — CP Class G Creates an external interrupt condition on the virtual machine.	EXTernal [code]
FILEDEF — CMS Simulates OS JCL data definition (DD) statements.	FIledef [ddname | nn | *] { Terminal [(optionA optionD] Printer | PUnch | Reader DISK [fn ft [fm] [optionB optionD] TAPi [optionC optionD] DUMMY CLEAR } optionA: [UPCASE | LOWCASE] optionB: [KEYLEN n] [EXTENT {n | 50}] [LIMCT n] [OPTCD {A | E | F | R}] [DISP MOD] optionC: [7TRACK | 9TRACK] [TRTCH {O | OC | OT | E | ET}] [DEN {200 | 556 | 800 | 1600}] optionD: [PERM] [CHANGE | NOCHANGE] [RECFM{F | FB | V | VB | U | VS | VBS | FS | FBS | A | M}] [LRECL n] [{BLOCK | BLKSIZE} n]
FLUSH — CP Class D Halts and immediately purges or holds the current spool file.	FLush raddr [ALL] [HOld]
FORCE — CP Class A Forces logout of the named user.	FORCE userid [HOld]
FORMAT — CMS Formats a disk for use by CMS.	FORMAT ccu mode [nocyl][(Recomp) | (LABEL)]
FREE — CP Class D Releases previously held user spool files.	FRee userid [Printer | PUnch | ALL]
GENDIRT — CMS Creates auxiliary module directories.	GENDIRT directory-name
GENMOD — CMS Generates absolute nonrelocatable files (MODULE files).	Genmod fn [ft [fm]] [(options...)] options: [NOMAP | MAP] [STR | NOSTR] [FROM ent1] [TO ent1] [SYSTEM]
GLOBAL — CMS Defines CMS libraries to be searched for macros and subroutines.	GLobal {MACLIB | TXTLIB} [libname...]
HALT — CP Class A Stops any active channel program on the real device specified.	HALT raddr
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]
HT — CMS Immediate Command Halts typing at the terminal.	HT
HX — CMS Immediate Command Halts the execution of the current CMS operation.	HX
INCLUDE — CMS Brings additional TEXT files into storage.	INclude fn... [(options...)] options: [CLEAR | NOCLEAR] [RESET entry | RESET *] [NOMAP | MAP] [NOINV | INV] [NOREP | REP] [TYPE | NOTYPE] [NOAUTO | AUTO] [NOLIBE | LIBE] [ORIGIN hexloc] [START] [SAME]
IPL — CP Class G Initiates a program load on the virtual machine.	Ipl {vaddr [cyl-no] | system-name} [CLear | NOCLear]
LINK — CP Class G Permits one user to access minidisks belonging to another user.	LINK [To] userid vaddr1 [As] vaddr2 [mode] [[PASS=] password]
LISTFILE — CMS Lists information about CMS files	Listfile [[fn [ft [fm]]] [(options)] options: [Header | NOHeader] [EXec | APpend] [FName | FType | FMode | FOrmat | ALloc | Date | Label]
LOAD — CMS Brings TEXT files into storage and establishes linkages.	LOAD fn ... [(options)] options: [CLEAR | NOCLEAR] [RESET entry | RESET *] [NOMAP | MAP] [NOINV | INV] [NOREP | REP] [TYPE | NOTYPE] [NOAUTO | AUTO] [NOLIBE | LIBE] [ORIGIN hexloc | ORIGIN TRANS] [START]
LOADBUF — CP Class D Loads UCS buffer on the real printer device.	LOADBUF raddr {UCS name [Fold] [Ver] | FCB name}
LOADMOD — CMS Brings a single MODULE file into storage.	LOADMod fn [ft fm]
LOADVFCB — CP Class G Loads a forms control image for a virtual 3211 printer.	LOADFVCB vaddr FCB name
LOCATE — CP Class E Provides the addresses of CP control blocks related to a specified user, virtual device, or real device.	LOCate {userid [vaddr] | raddr}
LOCK — CP Class A Locks specified pages in processor storage.	LOCK userid fpage lpage
LOGIN — CP Class ALL Initiates all virtual machine operation.	Login userid [password] [Mask] [Noipl]
LOGOUT — CP Class ALL Terminates a terminal session.	LOGout [HOld] LOGoff
MACLIB — CMS Performs maintenance on macro libraries.	MAClib { GEN | ADD | REP libname fn1 [fn2...] DEL libname macname[macname2...macnameN] COMP libname MAP libname [(TERM) | (PRINT) | (DISK)] }

```
         Command Description            |                Command Format
----------------------------------------+-----------------------------------------------------------
MODMAP                               CMS |MODmap  fn
  Types a MODULE file load map.          |
                                         |
MOVEFILE                             CMS |MOVEfile  [input-ddname  [output-ddname] ]
  Moves data from one device to          |          [  INMOVE      [  OUTMOVE    ] ]
  another device.                        |
                                         |
MSG                         CP Class B   |Msg  {ALL      }  msg-text
  Sends text messages to other           |     {userid   }
  users or the system operator.          |     {  *      }
                                         |     {OPerator }
                                         |
MSG                         CP Class ALL |Msg  {userid  }   msg-text
  Sends text messages to other           |     {OPerator}
  users or the system operator.          |     {  *     }
                                         |
NOTREADY                    CP Class G   |NOTReady  vaddr
  Simulates the loss of ready            |
  status on a virtual spooled unit       |
  record device.                         |
                                         |
ORDER                   CP Classes D,G   |ORDer [userid] {Reader } {CLass c1 CLass c2...}
  Provides a technique for ordering      |      [SYSTEM] {Printer} {spoolid1 spoolid2...}
  closed spool files.                    |               {PUnch  }
                                         |
PRINT                                CMS |PRint  fn ft [fm] [(options)]
  Directs a specified spool file         |options:
  to the virtual printer.                |
                                         |[CC  ] [MEMber  *  ] [UPcase] [HEX]
                                         |[NOCC] [MEMber name]
                                         |
PUNCH                                CMS |PUnch fn ft [fm] [([NOHeader] [MEMber  *   ]) ]
  Directs a specified spool file to      |                 [ [Header  ] [MEMBER name ]  ]
  the virtual punch.                     |
                                         |
PURGE                   CP Classes D,G   |PURge  [userid]
  Deletes a spooled file before          |       [SYSTEM]
  reading, printing, or punching         |
  occurs.                                |         {Reader  [ {CLass c1 CLass c2 ...} ]}
                                         |         {Printer [ {spoolid1 spoolid2 ...} ]}
                                         |         {PUnch   [  ALL                    ]}
                                         |         {ALL                                }
                                         |
QUERY                   CP Classes A,E   |Query   {PAGing         }
  Provides the paging activity           |        {PRIORity userid}
  index or specified user priority.      |
                                         |
QUERY                       CP Class B   |Query   {DAsd              }
  Provides the current status of         |        {TApes             }
  all system devices.                    |        {LINES             }
                                         |        {UR                }
                                         |        {STORage           }
                                         |        {ALL               }
                                         |        {raddr             }
                                         |        {                  }
                                         |        {SYStem    raddr   }
                                         |        {DUmp              }
                                         |
QUERY                       CP Class D   |Query   {Files               }
  Provides data on spooling              |        {Reader   [spoolid]  }
  operations.                            |        {Printer  [       ]  }
                                         |        {PUnch    [ALL    ]  }
                                         |        {Hold                }
                                         |
QUERY                       CP Class G   |Query   {Time                          }
  Provides the virtual machine user      |        {Files                         }
  with the current status of his         |        {Set                           }
  virtual machine, spooling devices      |        {TERMinal                      }
  and spool files.                       |        {                              }
                                         |        {[Virtual]    {DAsd   }        }
                                         |        {             {TApes  }        }
                                         |        {             {LINES  }        }
                                         |        {             {UR     }        }
                                         |        {             {STORage}        }
                                         |        {             {ALL    }        }
                                         |        {             {vaddr  }        }
                                         |        {                              }
                                         |        {Links         vaddr           }
                                         |        {                              }
                                         |        {Reader       [spoolid]        }
                                         |        {Printer      [ALL    ]        }
                                         |        {PUnch                         }
                                         |
QUERY                       CP Class ALL |Query   {LOGmsg          }
  Provides the remaining portion of      |        {Names           }
  the log message, and the names         |        {Users  [userid] }
  and real address of other logged       |        {userid          }
  on users.                              |
                                         |
QUERY                                CMS |Query   {BLIP                   }
  Permits the user to obtain             |        {RDYMSG                 }
  specified information about his        |        {LDRTBLS                }
  virtual machine's CMS functions.       |        {RELPAGE                }
                                         |        {IMPCP                  }
                                         |        {IMPEX                  }
                                         |        {ABBREV                 }
                                         |        {REDTYPE                }
                                         |        {PROTECT                }
                                         |        {SEARCH                 }
                                         |        {                       }
                                         |        {DISK     [mode]        }
                                         |        {         [ *  ]        }
                                         |        {                       }
                                         |        {         {SYSTEM}      }
                                         |        {SYNONYM  {USER  }      }
                                         |        {         {ALL   }      }
                                         |        {                       }
                                         |        {FILEDEF                }
                                         |        {MACLIB                 }
                                         |        {TXTLIB                 }
                                         |        {LIBRARY                }
                                         |        {INPUT                  }
                                         |        {OUTPUT                 }
                                         |
READCARD                             CMS |READcard   {fn ft [fm]          }
  Reads data from the spooled card       |           {      [A ]          }
  input device.                          |           {                    }
                                         |           {* [* [fm] ]         }
                                         |           {     [A ]           }
                                         |
READY                       CP Class G   |READY  vaddr
  Makes a device end interrupt           |
  pending for the specified virtual      |
  device.                                |
                                         |
RELEASE                              CMS |RELease   {ccu }
  Makes a disk and its directory         |          {mode}
  inaccessible to a virtual machine      |
                                         |
RENAME                               CMS |Rename   fileid1 fileidN [(options)]
  Changes the name of a CMS file         |
  or files.                              |options: [TYPE  ] [NOUPdirt]
                                         |         [NOTYPE] [UPdirt  ]
                                         |
REPEAT                      CP Class D   |REPeat  raddr  [  [nn]          ]
  Increases the copies of, or            |               [  [1 ]          ]
  holds, an output spool file.           |               [                ]
                                         |               [  [nn] HOld     ]
                                         |
RESET                       CP Class G   |RESET  vaddr
  Clears all pending interrupts and      |
  resets error conditions on the         |
  device specified.                      |
                                         |
REWIND                      CP Class G   |REWIND  vaddr
  Rewinds a real tape drive.             |
                                         |
RT             CMS Immediate Command     |RT
  Resumes terminal typing.               |
                                         |
RUN                                  CMS |RUN  fn [ft [fm]] [(args...)]
  Initiates a series of functions        |
  for a file.                            |
                                         |
SAVESYS                     CP Class E   |SAVESYS  system-name
  Creates a copy of virtual machine      |
  contents as they currently exist.      |
                                         |
SET                         CP Class A   |SET    {FAVored  userid  [xx ]}
  Sets special CP preferred              |       {REServe  userid  [OFF]}
  options.                               |       {                      }
                                         |       {PAGing   nn           }
                                         |       {                      }
                                         |       {PRIORity userid nn    }
                                         |
SET                         CP Class B   |SET    {        [nn  ]              }
  Establishes disposition for log        |       {LOGmsg  [NULL]              }
  messages and dumps.                    |       {                            }
                                         |       {DUmp   {AUTO }   [ALL]      }
                                         |       {       {raddr}   [CP ]      }
                                         |
SET                         CP Class F   |SET  RECord {OFF                                   }
  Sets recording mode for a device,      |            {ON raddr LIMIT nn BYTE nn BIT n       }
  or enables/disables soft machine       |            {                                      }
  check interrupts.                      |            {   [{AND} BYTE nn BIT n]              }
                                         |            {   [{OR }              ]              }
                                         |
                                         |           Mode {RETRY} {Quiet }
                                         |                {MAIN } {Record}
                                         |
SET                         CP Class G   |SET   {ACnt          {ON }   }
  Sets virtual machine options.          |      {MSg           {OFF}   }
                                         |      {Wng                   }
                                         |      {Run                   }
                                         |      {Linedit               }
                                         |      {NOTRans               }
                                         |      {                      }
                                         |      {Emsg          {ON  }  }
                                         |      {              {OFF }  }
                                         |      {              {CODE}  }
                                         |      {              {TEXT}  }
                                         |      {                      }
                                         |      {Timer         {ON  }  }
                                         |      {              {OFF }  }
                                         |      {              {REAL}  }
                                         |
SET                                  CMS |SET  function
  Sets or resets CMS virtual             |
  machine characteristics.               |functions:
                                         |
                                         |[BLIP string (count)]  [ABBREV           ]
                                         |[BLIP ON            ]  [REDTYPE    {ON } ]
                                         |[BLIP OFF           ]  [IMPEX      {OFF} ]
                                         |                       [IMPCP            ]
                                         |[LDRTBLS nn]           [PROTECT          ]
                                         |
                                         |[RDYMSG SMSG] [INPUT a xx] [OUTPUT xx a]
                                         |[RDYMSG LMSG] [INPUT     ] [OUTPUT     ]
                                         |
SHUTDOWN                    CP Class A   |SHUTDOWN
  Checkpoints and terminates the         |
  current VM/370 operation.              |
                                         |
SLEEP                       CP Class ALL |SLeep
  Places the virtual machine in a        |
  dormant state with the terminal        |
  keyboard locked.                       |
                                         |
SORT                                 CMS |SORT  fn1 ft1 fm1 fn2 ft2 fm2
  Rearranges records within a file.      |
                                         |
SPACE                       CP Clas D    |SPace  raddr
  Forces single spacing on the           |
  printer.                               |
                                         |
SPOOL                       CP Class G   |SPool  {Reader}
  Changes spooling control options.      |       {vaddr }  {[CLass a] [CONt  ] [HOld  ]}
                                         |                 {          [NOCont] [NOHold]}
                                         |                 {[EOF  ]                    }
                                         |                 {[NOEof]                    }
                                         |
                                         |       {Printer} {[[To] userid] [HOld  ]             }
                                         |       {PUnch  } {[OFF        ] [NOHold]             }
                                         |       {vaddr  } {                                   }
                                         |                 {[CONt  ] [CLass a] [COpy nn]       }
                                         |                 {[NOCont]                           }
                                         |
START                       CP Class D   |STArt  [ Reader                              ]
  Restarts a drained device or           |       [ Printer                             ]
  changes its output spooling            |       [ PUnch                               ]
  class.                                 |       [ ALL                                 ]
                                         |       [                                     ]
                                         |       [[raddr [CLass c] [NOSep]]...         ]
                                         |
START                                CMS |START  [ {entry} [args...]]
  Begins program execution.              |       [ {  *  }          ]
                                         |
STATE                                CMS |STATE  fn ft [fm]
  Verifies the existence of a file.      |
                                         |
STCP                        CP Class F   |STCP  {{ hexloc} hexwd1 [hexwd2...] }
  Alters real storage locations.         |      {{Lhexloc}                    }
                                         |      { Shexloc  hexdata            }
                                         |
STORE                       CP Class G   |STore   {hexloc                           }
  Alters virtual machine storage,        |        {Lhexloc   hexwd1 [hexwd2...]     }
  PSW, and registers.                    |        {                                 }
                                         |        {Shexloc   hexdata                }
                                         |        {                                 }
                                         |        {Greg                             }
                                         |        {Yreg      hexwd1 [hexwd2...]     }
                                         |        {Xreg                             }
                                         |        {                                 }
                                         |        {Psw     [hexwd1] hexwd2          }
                                         |
SVCTRACE                             CMS |SVCTrace  {ON }
  Records information about              |          {OFF}
  supervisor call instructions.          |
                                         |
SYNONYM                              CMS |SYNonym  [[fn [ft [fm]]]  [(options...)]]
  Specifies alternate names by           |
  which CMS commands may be              |options:  [NOSTD]  [CLEAR]
  invoked.                               |          [STD  ]
                                         |
SYSTEM                      CP Class G   |SYStem  {CLEAR  }
  Simulates virtual machine console      |        {RESET  }
  functions.                             |        {RESTART}
                                         |
TAPE                                 CMS |TAPE {DUMP {fn} {ft} [fm] [(optA optB optC)] }
  Performs tape to disk or disk to       |     {     {* } {* } [* ]                    }
  tape operations for CMS data           |     {                                       }
  sets.                                  |     {LOAD [fn [ft [fm]]] [(optA optB optC)] }
                                         |     {     [*  [*  [A ]]]                    }
                                         |     {                                       }
                                         |     {SCAN [fn [ft ]]     [(optA optB optC)] }
                                         |     {     [*  [*  ]]                        }
                                         |     {                                       }
                                         |     {SKIP {fn} {ft}      [(optA optB optC)] }
                                         |     {     {* } {* }                         }
                                         |     {                                       }
                                         |     {MODEset [(optD)]                       }
                                         |     {                                       }
                                         |     {tapcmd [n] [(optD)]                    }
                                         |     {       [1]                             }
                                         |
                                         |optA: [WTM  ] optB: [NOPRint] optC: [EOF n]
                                         |      [NOWTM]       [PRint  ]       [EOT  ]
                                         |optD:               [DISK   ]       [EOF 1]
                                         |[DEN  200]          [TERM   ]
                                         |[DEN  556]
                                         |[DEN  800]  [TAPi|ccu] [TRTCH {O|OC|OT|E|ET}]
                                         |[DEN 1600]  [TAP1|181]
                                         |
                                         |tapcmd: [BSF|BSR|ERG|FSF|FSR|REW|RUN|WTM]
                                         |
TAPPDS                               CMS |TAPPDS  [fn [ft [fm]]]  [(options...)]
  Loads an OS partitioned data set       |options
  (PDS) file or card-image records       |
  from tape to disk.                     |[NOPDS] [COL1  ] [TAPn] [END  ] [MAXTEN  ]
                                         |[PDS  ] [NOCOL1] [TAP1] [NOEND] [NOMAXTEN]
                                         |
TERMINAL                    CP Class G   |TERMinal    {CHardel   {ON  }  }
  Changes parameters for terminal        |            {LINEDel   {OFf }  }
  operations.                            |            {LINENd    {char}  }
                                         |            {EScape            }
                                         |            {                  }
                                         |            {Mask        {ON } }
                                         |            {APL         {OFF} }
                                         |            {ATTn              }
                                         |
TRACE                       CP Class G   |TRace  {SVC                                      }
  Traces and records program             |       {I/O                                      }
  execution.                             |       {PROgram                                  }
                                         |       {EXTernal   [ Printer                  ]  }
                                         |       {PRIV       [                          ]  }
                                         |       {SIO        [ [BOTH    ]  [RUN  ]      ]  }
                                         |       {CCW        [ [TERMinal]  [NORUN]      ]  }
                                         |       {BRanch     [                          ]  }
                                         |       {INSTruct   [ OFf                      ]  }
                                         |       {ALL                                      }
                                         |       {CSW                                      }
                                         |       {                                         }
                                         |       {END                                      }
                                         |
TRANSFER                CP Classes D,G   |TRANsfer  [userid] {spoolid} [To] userid
  Transfers closed reader spool          |          [SYSTEM] {CLass a}
  files.                                 |                   {ALL    }
                                         |
TXTLIB                               CMS |TXTlib   {GEN   libn fn ...                        }
  Performs maintenance on a              |         {ADD                                      }
  library of TEXT files (object          |         {                                         }
  modules).                              |         {DEL   libn member ...                    }
                                         |         {                                         }
                                         |         {MAP   libn [(TERM)|(PRINT)|(DISK)]       }
                                         |
TYPE                                 CMS |Type  fn ft [fm] [{rec1} [recN]] [(options)]
  Types all or part of a file at a       |options:         [{ *  } [ *  ]]
  terminal.                              |
                                         |[COL  xxx  - {yyy  }] [HEX] [MEMber   * ]
                                         |[      1     {lrecl}]       [MEMber name]
                                         |
UNLOCK                      CP Class A   |UNLOCK  {userid fpage lpage}
  Releases storage.                      |        {VIRT=REAL         }
                                         |
UPDATE                               CMS |Update fn1[ft1[fm1[fn2[ft2[fm2]]]]] [(options)]
  Makes changes in a file as             |options:
  defined by control cards in a          |
  record file.                           |[REP  ] [NOSEQ8] [INC  ] [CTL  ] [STK  ]
                                         |[NOREP] [SEQ8  ] [NOINC] [NOCTL] [NOSTK]
                                         |
                                         |[NOTERM] [PRINT]
                                         |[TERM  ] [DISK ]
                                         |
VARY                        CP Class B   |VARY  DEV raddr {ONLINE }
  Varies the availability of a           |                {OFFline}
  device.                                |
                                         |
WNG                     CP Classes A,B   |Wng  {userid  }  msg-text
  Sends high priority messages.          |     {OPerator}
                                         |     {ALL     }
                                         |
*                           CP Class ALL |*  any-comment
  Permits comments.                      |
                                         |
*                                    CMS |*  any-comment
  Permits comments                       |
```

GX20-1926-1

IBM

International Business Machines Corporation
Data Processing Division
1133 Westchester Avenue, White Plains, New York 10604
[U.S.A. only]

IBM World Trade Corporation
821 United Nations Plaza, New York, New York 10017
[International]

Printed in U.S.A.