Fifth Edition (December 1986)

This edition, SX20-4400-4, is a major revision of SX20-4400-3. It applies to Release 5 of the Virtual Machine/System Product, and to all subsequent releases unless otherwise indicated in new editions or Technical Newsletters. Changes are made periodically to the information herein; before using this publication in connection with the operation of IBM systems, consult the latest IBM System/370, 30xx, and 4300 Processors Bibliography, GC20-0001, for the editions that are applicable and current.

Summary of Changes

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

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

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

Ordering Publications

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

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

Preface

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

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

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

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

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

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

Using This Publication

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

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

**Chapter 1. Using the VM/SP System** ........................................ 1
Getting Started ........................................................................ 2
Terminal Operating Procedures .............................................. 3
   Logging On ........................................................................... 3
   Logon Exceptions .................................................................. 6
   Logging Off .......................................................................... 8
Input Conventions ..................................................................... 10

**Chapter 2. Using CMS** ....................................................... 15
   How To Initial Program Load (IPL) CMS ................................ 16
   System Product Editor ......................................................... 16
   EXEC 2 Interpreter .............................................................. 17
   System Product Interpreter .................................................. 18
   CMS Session Services .......................................................... 19
Enhanced Connectivity Facilities on VM/SP ................................ 21
Example of CMS Program Development Facilities ..................... 22
   Creating an Assembler Language Source File ....................... 22
   Assembling a Source File ..................................................... 24
   Correcting Errors .................................................................. 25
   Creating a Load Module ....................................................... 26
   Testing and Correcting a Program ........................................ 26
   Erasing Unwanted Files ....................................................... 28
CMS File System ...................................................................... 29
   Printing, Sending, and Reading Files .................................... 29
   CMS Fileids ........................................................................ 32
Return Codes ........................................................................... 39
   CMS Return Codes ............................................................ 39
   CP DIRECT Command Return Codes ................................... 41
   CMS DDR Command Return Codes ...................................... 42
   System Product Editor (XEDIT) Command Return Codes .......... 42
   IPCS Commands Return Codes ............................................ 45
   Example of a Return Code from a CP Command .................... 45

**Chapter 3. Summary of VM/SP Commands and Service Aids** ........ 47
Notational Conventions ........................................... 48
CP Privilege Classes ............................................... 50

Chapter 4. CP, CMS, GCS, TSAF, RSCS, and IPCS Commands .... 53

Summary of Changes .............................................. 273
  The HELP Facility ............................................... 273
  NLS Parser Summary ............................................ 273
  Enhancements for EXECs in Storage ............................ 274
  Advanced Printer Subsystem Support ........................... 274
  Transparent Services Access Facility .......................... 274
  Enhanced 3270 Usability ........................................ 275
  New Commands for Release 5 .................................... 275
  TSAF Commands .................................................. 276
  Changed Commands for Release 5 .............................. 276

Bibliography ....................................................... 277
  Prerequisite Publications ....................................... 277
  Corequisite Publications ....................................... 278

Index .................................................................. 281
Figures

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

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

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

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

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

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

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

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

Getting Started

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

- A user identification and password. Your user identification
  (userid) identifies you to VM/SP. Your password is checked
  when you log on, and protects your virtual machine from
  unauthorized use. The VM/SP system operations group usually
  assigns userids and passwords.

- A virtual machine defined for your use. The virtual machine
  definition should include all the devices you expect to use, for
  example:

  A console
  Card reader
  Card punch
  Printer
  Disk space.

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

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

---

**Terminal Operating Procedures**

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

---

**Logging On**

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

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

If your terminal is a 3270-type, with a screen size of 20x80 or larger, you may log on directly from the logo screen.
Below the actual VM/SP logo you will find instructions for filling in your userid and password. For security reasons, your password will not appear when you enter it.

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

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

DMKLOG050E LOGON unsuccessful - incorrect password

or

DMKLOG053E userid not in CP directory

Enter one of the following commands:

<table>
<thead>
<tr>
<th>Command</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>LOGON userid</td>
<td>LOGON VMUSER1</td>
</tr>
<tr>
<td>DIAL userid</td>
<td>DIAL VMUSER2</td>
</tr>
<tr>
<td>MSG userid message</td>
<td>MSG VMUSER GOOD MORNING</td>
</tr>
<tr>
<td>LOGOFF</td>
<td></td>
</tr>
</tbody>
</table>

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

DMKLOG050E userid not in CP directory

Enter password (It will not appear when typed):

You may also enter your userid in the USERID input area without your password or enter the LOGON command, followed by your userid, in the COMMAND input area. The following prompt will appear:
Enter password (It will not appear when typed):

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

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

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

Enter one of the following commands:

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

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

1 userid (short for LOGON userid)

Then press the Enter (or Return) key.

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

Enter password (it will not appear when typed):

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

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

If you make an error during the logon procedure the system will give you an error message and you must start the logon procedure from the beginning by entering "1 userid" again.
For example, if you type your password incorrectly VM/SP will give you this error message:

DMKLOG050E LOGON unsuccessful--incorrect password

Enter one of the following commands: ......

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

DMKLOG054E Already logged on devtype rdev

Enter one of the following commands: ......

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

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

Unknown CP command: name

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

- Informational messages regarding linkage status of minidisks attached to your virtual machine

- LOGMSG setup time (the time and date that the operator created a log message)

- The log message line that the operator created

- The FILES message (which tells you if you have any virtual reader, print, or punch files)

- A LOGON message, such as:
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:

\texttt{log} \quad \text{(short for LOGOFF)}

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

\texttt{CONNECT=hh:mm:ss VIRTCPU=mmm:ss.ss TOTCPU=mmm:ss.ss}
\texttt{LOGOFF AT hh:mm:ss zzz weekday mm/dd/yy}

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

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

\texttt{log hold} \quad \text{(short for LOGOFF HOLD)}

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

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

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

Logoff and Security

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

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

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

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

**Note:** The same sequence of events occurs:

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

Input Conventions

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

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

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

Using Logical Line Editing Characters

If you make a typing error when entering a command on a line-editing terminal, you can correct it by using one of four special characters. These four “logical line-editing” characters are more useful with typewriter-like terminals. (which do not have cursor-controlled line-editing) than with display terminals, but are valid for use with both. The characters listed in Figure 1 are the default characters in the system. They can be changed to others if your terminal keyboard does not have these characters. You can define some other infrequently used keyboard character to perform these editing functions, by using the CP TERMINAL command. For details on using the CP TERMINAL command to change default values, see VM/SP CP Command Reference.
<table>
<thead>
<tr>
<th>Character</th>
<th>Function</th>
<th>Usage and Result Obtained</th>
</tr>
</thead>
<tbody>
<tr>
<td>@</td>
<td>Logical Character Delete</td>
<td>abc#@@ results in ab abc@# results in abd @#def results in def abc@@ deletes the whole line</td>
</tr>
<tr>
<td>#</td>
<td>Logical Line End</td>
<td>down 1#type 1#top results in execution of the commands as though entered: down 1 type 1 top</td>
</tr>
<tr>
<td>c</td>
<td>Logical Line Delete</td>
<td>abc@c results in abc abc#def@c results in abc abc#def#c-results in abc def abc#def#cghi results in abc defghi Note: The logical line delete symbol (c) deletes all typed characters back to and including the previous logical line end symbol (#). Use the logical line delete symbol when you have made several errors in a line.</td>
</tr>
<tr>
<td>&quot;</td>
<td>Logical Escape</td>
<td>abc&quot;@ results in abc@d abc&quot;@ results in abc &quot;abc&quot; results in &quot;abc&quot; Note: The editor ignores any quotation mark (&quot;) that appears as the last character of a line.</td>
</tr>
</tbody>
</table>

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

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

**Line Termination**
An input data line from an IBM 2741 Communications Terminal is transmitted to the processor by pressing the Return key. The same function is performed on the 3270 Information Display System Terminal by pressing the Enter key. Other terminals have similar line termination keys. As you do work on your terminal, the lower right corner of the screen will display various status notices. This
tells you what is happening in the system at the present time. This figure shows you how to change the status.

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

Notes:

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

**Figure 2 (Part 1 of 2). Summary of Status Action on Display Terminals**
3 If you are running with TERMINAL MODE CP (the default for the primary system operator), an attention return also occurs that cancels the function. You can use this to terminate certain QUERY or DISPLAY functions on consoles that do not have a PA1 key.

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

5 The status remains VM READ if you have SET AUTOREAD ON.

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

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

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

- Logon with a valid user identification and password. The user identification should have a virtual machine defined for its use.

- Initial program load (IPL) the CMS system by specifying the name of the CMS system or the device address of the CMS system disk.

- Have disk space available that is formatted for use by CMS.

The logging on procedure is discussed in "Terminal Operating Procedures" on page 3. The IPL procedure is described in this section. Disk formatting is detailed in the VM/SP CMS User’s Guide.
How To Initial Program Load (IPL) CMS

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

\texttt{ipl cms}

or

\texttt{ipl 190}

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

\texttt{Ready; T=7.36/19.87 09:26:12}

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

System Product Editor

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

- Full screen editing
- Multiple views of the same file
- Multiple files viewed on the same screen
- Addition of color and extended highlighting
- Ability to issue selected commands directly from the displayed line
• Ability to define screen formats with flexibility, including splitting the screen vertically

• Extended use of the prefix area

• Extended string search to facilitate text processing

• Column pointer for intraline-editing

• Line splitting and/or joining functions

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

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

**EXEC 2 Interpreter**

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

• Acceptance of 255-character words

• Issuance of commands to either CMS or to specified other subcommand environments

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

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

### System Product Interpreter

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

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

- Uses a general-purpose high-level language, REXX.
- Supports structured programming concepts.
- Has many built-in functions.
- Accepts programs written in mixed case (which makes them easier to read).
- Has extensive mathematical capabilities, such as decimal, exponential, and scientific arithmetic.
• Uses a full set of arithmetic, character, and logical operators with algebraic precedence and parentheses.

• Accepts programs in free format.

• Makes XEDIT macros and system EXECs easier to maintain. It easily handles subroutine calls to other EXECs, modules, or internal routines.

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


---

<table>
<thead>
<tr>
<th>CMS Session Services</th>
</tr>
</thead>
</table>
| This support improves the usability of VM/SP on 3270-type display terminals. It includes:

1. Window functions for the CMS user

2. Full-screen CMS

The support also provides functions for:

• Defining and deleting windows

• Positioning windows anywhere on the screen

• Overlaying one window with another

• Scrolling backward and forward through data in a window
• Changing color, highlighting, and other characteristics of virtual screens

• Writing data into virtual screens

• Logging data in a CMS file

Full-screen CMS has the following characteristics:

• Routing VM output and messages to the appropriate windows

• Using extended highlighting for CMS output

• Defining CMS PF keys

• Controlling the display of messages

You can find complete information on this support and its functions in the VM/SP CMS User’s Guide and the VM/SP CMS Command Reference.
Enhanced Connectivity Facilities on VM/SP

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

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

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

- The Server-Requester Programming Interface (SRPI).

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

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

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

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

Creating an Assembler Language Source File

The sample program that follows is an Assembler language program that reads data from one CMS file and writes it to another CMS file. After you have logged on the system and issued IPL CMS, you can create the program using the CMS XEDIT facility.
xedit manip assemble
CREATING NEW FILE:
input
INPUT:
MANIP    CSECT
PRINT NOGEN
SAVE (14,12),,*
BALR 12,0
USING *,12    ESTABLISH ADDRESSABILITY
LA 2,8(,1)  R2=ADDR OF INPUT FILE IN PLIST
LA 3,32(,1) R3=ADDR OF OUTPUT FILE IN PLIST
* DETERMINE IF INPUT FILE EXISTS
FSSTATE (2),ERROR=ERR1
* READ A RECORD FROM INPUT FILE AND WRITE ON OUTPUT FILE
RD    FSREAD (2),ERROR=EOF,BUFFER=BUFF1,BSIZE=80
      FSWRITE (3),ERROR=ERR2,BUFFER=BUFF1,BSIZE=80
      B    RD    LOOP BACK FOR NEXT RECORD
* COME HERE IF ERROR READING INPUT FILE
EOF     EQU *
      LA 15,7    TEST CODE FOR READ ERROR
      C 15,F'12'  END OF FILE?
      BNE ERR3    ERROR IF NOT
      RETURN (14,12),RC=0
* IF INPUT FILE DOES NOT EXIST
ERR1    WRTTERM 'FILE NOT FOUND',EIDT=YES
      B  ERRET
* IF ERROR WRITING FILE
ERR2    LINEDIT TEXT='ERROR CODE ..... IN WRITING FILE', $
        SUB=(DEC,(15))
      B  ERRT
* IF READING ERROR WAS NOT NORMAL END OF FILE
ERR3    LINEDIT TEXT='ERROR CODE ..... IN READING FILE', $
        SUB=(DEC,(15))
ERRET    RETURN (14,12),RC=1 RETURN TO CALLER
BUFF1    DS CL80
END MANIP
(Press the ENTER/RETURN key to leave Input mode.)
(Remember that the continuation character 's' must be in in column 72)

XEDIT:
file
Ready;

Figure 3. Sample Assembler Language Program Used for Creating a Source File
The Editor (the term applied to the edit program that is used by the XEDIT command) did not find a file with the filename and filetype of MANIP ASSEMBLE, so it created the file for you. Enter the INPUT subcommand so that you can enter your program code into the file. You must issue the FILE subcommand to save your program.

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

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

---

**Assembling a Source File**

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

```
assemble manip
```

```
ASSEMBLER (XF) DONE

MAN00331       B       ERRT
IFO024 NEAR OPERAND COLUMN 1-UNDEF SYMBOL

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

Ready (00008);

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

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

1. The MANIP ASSEMBLE file contains the source statements of your program. This file was the input used by the Assembler
Language program. The output from the assembler is two permanent files, MANIP TEXT and MANIP LISTING.

2. The MANIP TEXT file contains the object module.

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

Correcting Errors

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

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

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

```
assemble manip
```
ASSEMBLER (XF) DONE

NO STATEMENTS FLAGGED IN THIS ASSEMBLY
Ready;

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

Creating a Load Module

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

load manip
Ready;

genmod manip
Ready;

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

Testing and Correcting a Program

Once the MODULE file has been created, you can begin testing. To execute the MANIP MODULE file, issue the MANIP command name, plus the file identifiers for the input and output files. The input file (MANIP ASSEMBLE A1) is to be copied and the resulting file is to be called MANIP1 ASSEMBLE A1. The first test should take the branch on the FSREAD error. The following error message appears on the terminal:
You should then use the Editor to correct the program so that this branch is no longer taken.

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

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

```
assemble manip

ASSEMBLER (XF) DONE

NO STATEMENTS FLAGGED IN THIS ASSEMBLY
Ready;
```

```
load manip
Ready;
```

```
genmod manip
Ready;
```

This command produces a file called "MANIP MODULE."

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

```
manip file1 assemble al file2 assemble al
FILE NOT FOUND
Ready (00001);
```
Then, attempt to copy a file to itself. Your program is not equipped to do this; an error occurs.

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

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

```
manip manip assemble al manip1 assemble al
Ready;
```

---

**Erasing Unwanted Files**

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

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

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

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

```
listfile * assemble
MANIP   ASSEMBLE Al
Ready;
```
Printing, Sending, and Reading Files

Printing Files

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

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

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

```plaintext
print manip listing
Ready;
```

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

```plaintext
erase manip listing
Ready;

erase manip text
Ready;
```
Sending Files

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

Reading Files

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

FILES: 001 RDR, NO PRT, NO PUN

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

query reader all

<table>
<thead>
<tr>
<th>ORIGINID</th>
<th>FILE</th>
<th>CLASS</th>
<th>RECORDS</th>
<th>CPY</th>
<th>HOLD</th>
<th>DATE</th>
<th>TIME</th>
<th>NAME</th>
<th>TYPE</th>
<th>DIST</th>
</tr>
</thead>
<tbody>
<tr>
<td>NET4</td>
<td>1014</td>
<td>A PUN</td>
<td>0000051</td>
<td>001</td>
<td>NONE</td>
<td>08/17</td>
<td>12:36:03</td>
<td>MANIP</td>
<td>TEXT</td>
<td>G41/</td>
</tr>
</tbody>
</table>

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

RECEIVE

Or issue

READ *

READ CONTROL CARD MISSING. FOLLOWING ASSUMED:. READ READCARD CMSUT1 A1
CMS reads the first spool reader file in the queue and if there are
READ control cards in the input stream, it names the files as
indicated on the control cards.

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

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

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

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

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

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

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

CMS can be used for many other purposes. Those functions
illustrated in the previous discussion are intended to help VM/SP
users become acquainted with the system and its capabilities. Once
you are familiar with these commands and functions, you have a sound base upon which to build a more thorough understanding of the VM/SP system.

---

CMS Fileids

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

A **fileid** consists of:

filename filetype filemode.

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

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

A valid **filemode** is a 2-character CMS file identifier field comprising the filemode letter (A through Z) followed by the filemode number (0 through 6). The filemode letter indicates the CMS file directory on which the file resides and whether or not the disk is a user virtual disk or a CMS system disk. The filemode number indicates the access mode of the disk.
Filemode 0 Makes that file private. No other users may access it unless they have read/write access to your disk.

Filemode 1 Most common for reading and writing

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

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

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

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

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

For more information see the VM/SP CMS User’s Guide.
Reserved Filetype Descriptions

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

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

<table>
<thead>
<tr>
<th>Filetype</th>
<th>Command</th>
<th>Usage</th>
<th>Filename</th>
<th>Format</th>
<th>LRECL</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMSERV</td>
<td>AMSERV</td>
<td>Input</td>
<td>fn</td>
<td>F</td>
<td>80</td>
<td>Input control statements for Access Method Services</td>
</tr>
<tr>
<td>ASM3705</td>
<td>ASM3705</td>
<td>Input</td>
<td>fn fn(nn)</td>
<td>F</td>
<td>80</td>
<td>3704/3705 assembler source statements</td>
</tr>
<tr>
<td>ASSEMBLE</td>
<td>ASSEMBLE</td>
<td>Input</td>
<td>fn</td>
<td>F</td>
<td>80</td>
<td>Assembler language source statements</td>
</tr>
<tr>
<td>AUXxxxx</td>
<td>UPDATE</td>
<td>Input</td>
<td>fn</td>
<td>F</td>
<td>80</td>
<td>Auxiliary update file</td>
</tr>
<tr>
<td>BASDATA</td>
<td>BASIC</td>
<td>Execution time files</td>
<td>fn</td>
<td>V</td>
<td>256</td>
<td>User input and output files</td>
</tr>
<tr>
<td>BASIC</td>
<td>BASIC</td>
<td>Input</td>
<td>fn</td>
<td>V</td>
<td>156</td>
<td>BASIC language source statements</td>
</tr>
<tr>
<td>CMSUT1</td>
<td>READCARDCOPYFILE_DISK LOAD TAPE_LOAD UPDATE INCLUDE LOAD MACLIB EDIT XEDIT</td>
<td>Intermediate work files</td>
<td>READCARD COPYFILE_DISK TAPE fn DMSLDR DMSLDR DMSLBM EDIT XEDIT</td>
<td>F</td>
<td>80</td>
<td></td>
</tr>
<tr>
<td>CNTRL</td>
<td>UPDATE</td>
<td>Input</td>
<td>fn</td>
<td>F</td>
<td>80</td>
<td>Control file update</td>
</tr>
<tr>
<td>COBOL</td>
<td>COBOL</td>
<td>Input</td>
<td>fn</td>
<td>F</td>
<td>80</td>
<td>COBOL source statements</td>
</tr>
<tr>
<td>COPY</td>
<td>MACLIB</td>
<td>Input</td>
<td>fn</td>
<td>F</td>
<td>80</td>
<td>COPY control cards and macro definitions</td>
</tr>
<tr>
<td></td>
<td>SSERV</td>
<td>Output</td>
<td>fn</td>
<td>F</td>
<td>80</td>
<td>A book from a DOS/VS source library</td>
</tr>
<tr>
<td>DIRECT</td>
<td>DIRECT</td>
<td>Input</td>
<td>fn</td>
<td>F</td>
<td>80</td>
<td>User directory entries</td>
</tr>
</tbody>
</table>

Figure 4 (Part 1 of 5). Reserved Filetypes
<table>
<thead>
<tr>
<th>Filetype</th>
<th>Command</th>
<th>Usage</th>
<th>Filename</th>
<th>Format</th>
<th>RECFM</th>
<th>LRECL</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>DOSLIB</td>
<td>DOSLIB</td>
<td>Input</td>
<td>fn</td>
<td>V</td>
<td>1024</td>
<td></td>
<td>CMS/DOS phase library</td>
</tr>
<tr>
<td></td>
<td>DOSLKED</td>
<td>Input</td>
<td>fn</td>
<td>F</td>
<td>80</td>
<td></td>
<td>Linkage editor control statements for input to CMS/DOS linkage editor</td>
</tr>
<tr>
<td></td>
<td>FETCH</td>
<td>Input</td>
<td>fn</td>
<td>F</td>
<td>80</td>
<td></td>
<td>Input control statements for ESERV program</td>
</tr>
<tr>
<td>ESERV</td>
<td>ESERV</td>
<td>Input</td>
<td>fn</td>
<td>F</td>
<td>80</td>
<td></td>
<td>EXEC statements</td>
</tr>
<tr>
<td>EXEC</td>
<td>EXEC</td>
<td>Input</td>
<td>fn</td>
<td>V</td>
<td>130</td>
<td></td>
<td>FORTTRAN source statements</td>
</tr>
<tr>
<td></td>
<td>EXEC2</td>
<td>Input</td>
<td>fn</td>
<td>V</td>
<td>256</td>
<td></td>
<td>FREEFORM FORTRAN source statements</td>
</tr>
<tr>
<td></td>
<td>REXX</td>
<td>Input</td>
<td>fn</td>
<td>V</td>
<td>none</td>
<td></td>
<td>GCS EXEC statements</td>
</tr>
<tr>
<td></td>
<td>LISTFILE</td>
<td>Output</td>
<td>fn</td>
<td>F/V</td>
<td>520</td>
<td>520</td>
<td>Collection of named variables</td>
</tr>
<tr>
<td>GCS</td>
<td>EXEC</td>
<td>Input</td>
<td>fn</td>
<td>V</td>
<td>130</td>
<td></td>
<td>HELPCMS HELP files for HELP facility</td>
</tr>
<tr>
<td>GLOBALV</td>
<td>GLOBALV Defaults</td>
<td>Input/Output</td>
<td>fn Initial Session Lasting</td>
<td>F/V</td>
<td>520</td>
<td>520</td>
<td>HELPEDIT HELP files for HELP facility</td>
</tr>
<tr>
<td>GROUP</td>
<td>GROUP</td>
<td>Output</td>
<td>fn</td>
<td>F</td>
<td>80</td>
<td></td>
<td>HELPDEBU HELP files for HELP facility</td>
</tr>
<tr>
<td>HELPABBR</td>
<td>HELP</td>
<td>Input</td>
<td>fn</td>
<td>V/F</td>
<td>79</td>
<td></td>
<td>HELPFEXC HELP files for HELP facility</td>
</tr>
<tr>
<td>HELPCMS</td>
<td>HELP</td>
<td>Input</td>
<td>fn</td>
<td>V</td>
<td>79</td>
<td></td>
<td>HELPFEXC2 HELP files for HELP facility</td>
</tr>
<tr>
<td>HELPCMQ</td>
<td>HELP</td>
<td>Input</td>
<td>fn</td>
<td>V</td>
<td>79</td>
<td></td>
<td>HELPGROUP HELP files for HELP facility</td>
</tr>
<tr>
<td>HELPCMSS</td>
<td>HELP</td>
<td>Input</td>
<td>fn</td>
<td>V</td>
<td>79</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HELPCP</td>
<td>HELP</td>
<td>Input</td>
<td>fn</td>
<td>V</td>
<td>79</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HELPCPQU</td>
<td>HELP</td>
<td>Input</td>
<td>fn</td>
<td>V</td>
<td>79</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HELPCPSE</td>
<td>HELP</td>
<td>Input</td>
<td>fn</td>
<td>V</td>
<td>79</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HELPDEBU</td>
<td>HELP</td>
<td>Input</td>
<td>fn</td>
<td>V</td>
<td>79</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HELPEDIT</td>
<td>HELP</td>
<td>Input</td>
<td>fn</td>
<td>V</td>
<td>79</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HELPEXC</td>
<td>HELP</td>
<td>Input</td>
<td>fn</td>
<td>V</td>
<td>79</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HELPEXC2</td>
<td>HELP</td>
<td>Input</td>
<td>fn</td>
<td>V</td>
<td>79</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 4 (Part 2 of 5). Reserved Filetypes
<table>
<thead>
<tr>
<th>Filetype</th>
<th>Command</th>
<th>Usage</th>
<th>Filename</th>
<th>Format</th>
<th>RECFM</th>
<th>LRECL</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>HELPHELP</td>
<td>HELP</td>
<td>Input</td>
<td>fn</td>
<td></td>
<td>V</td>
<td>79</td>
<td>Input files for HELP facility</td>
</tr>
<tr>
<td>HELPPCMS</td>
<td>HELP</td>
<td>Input</td>
<td>fn</td>
<td></td>
<td>V</td>
<td>79</td>
<td>Input files for HELP facility</td>
</tr>
<tr>
<td>HELPMENU</td>
<td>HELP</td>
<td>Input</td>
<td>fn</td>
<td></td>
<td>V</td>
<td>77</td>
<td>Input files for HELP facility</td>
</tr>
<tr>
<td>HELPMMSG</td>
<td>HELP</td>
<td>Input</td>
<td>fn</td>
<td></td>
<td>V</td>
<td>79</td>
<td>Input files for HELP facility</td>
</tr>
<tr>
<td>HELPPF</td>
<td>HELP</td>
<td>Input</td>
<td>fn</td>
<td></td>
<td>V</td>
<td>79</td>
<td>Input files for HELP facility</td>
</tr>
<tr>
<td>HELPREF</td>
<td>HELP</td>
<td>Input</td>
<td>fn</td>
<td></td>
<td>V</td>
<td>79</td>
<td>Input files for HELP facility</td>
</tr>
<tr>
<td>HELPSET</td>
<td>HELP</td>
<td>Input</td>
<td>fn</td>
<td></td>
<td>V</td>
<td>79</td>
<td>Input files for HELP facility</td>
</tr>
<tr>
<td>HELPSQLD</td>
<td>HELP</td>
<td>Input</td>
<td>fn</td>
<td></td>
<td>V</td>
<td>79</td>
<td>Input files for HELP facility</td>
</tr>
<tr>
<td>HELPSRPI</td>
<td>HELP</td>
<td>Input</td>
<td>fn</td>
<td></td>
<td>V</td>
<td>79</td>
<td>Input files for HELP facility</td>
</tr>
<tr>
<td>HELPTASK</td>
<td>HELP</td>
<td>Input</td>
<td>fn</td>
<td></td>
<td>V</td>
<td>105</td>
<td>Input files for HELP facility</td>
</tr>
<tr>
<td>HELPTSAF</td>
<td>HELP</td>
<td>Input</td>
<td>fn</td>
<td></td>
<td>V</td>
<td>79</td>
<td>Input files for HELP facility</td>
</tr>
<tr>
<td>HELPXEDI</td>
<td>HELP</td>
<td>Input</td>
<td>fn</td>
<td></td>
<td>V</td>
<td>79</td>
<td>Input files for HELP facility</td>
</tr>
<tr>
<td>$HELPxxxx</td>
<td>HELPCNV</td>
<td>Output</td>
<td>fn</td>
<td></td>
<td>same as input</td>
<td>Converted file for HELP facility</td>
<td></td>
</tr>
<tr>
<td>LISTING</td>
<td>ASSEMBLE</td>
<td>Output</td>
<td>fn</td>
<td></td>
<td>F</td>
<td>121</td>
<td>COBOL processor output used as input to SOURCE subcommand of TESTCOB</td>
</tr>
<tr>
<td></td>
<td>ASM3705</td>
<td>Output</td>
<td>fn</td>
<td></td>
<td>F</td>
<td>121</td>
<td>COBOL processor output used as input to SOURCE subcommand of TESTCOB</td>
</tr>
<tr>
<td></td>
<td>ESERV</td>
<td>Output</td>
<td>fn</td>
<td></td>
<td>F</td>
<td>121</td>
<td>COBOL processor output used as input to SOURCE subcommand of TESTCOB</td>
</tr>
<tr>
<td></td>
<td>GOFORT</td>
<td>Output</td>
<td>fn</td>
<td></td>
<td>F</td>
<td>121</td>
<td>COBOL processor output used as input to SOURCE subcommand of TESTCOB</td>
</tr>
<tr>
<td></td>
<td>FORTHG</td>
<td>Output</td>
<td>fn</td>
<td></td>
<td>F</td>
<td>121</td>
<td>COBOL processor output used as input to SOURCE subcommand of TESTCOB</td>
</tr>
<tr>
<td></td>
<td>FORTHX</td>
<td>Output</td>
<td>fn</td>
<td></td>
<td>F</td>
<td>121</td>
<td>COBOL processor output used as input to SOURCE subcommand of TESTCOB</td>
</tr>
<tr>
<td></td>
<td>COBOL</td>
<td>Output</td>
<td>fn</td>
<td></td>
<td>F</td>
<td>121</td>
<td>COBOL processor output used as input to SOURCE subcommand of TESTCOB</td>
</tr>
<tr>
<td></td>
<td>PLIC</td>
<td>Output</td>
<td>fn</td>
<td></td>
<td>F</td>
<td>121</td>
<td>COBOL processor output used as input to SOURCE subcommand of TESTCOB</td>
</tr>
<tr>
<td></td>
<td>PLICCR</td>
<td>Output</td>
<td>fn</td>
<td></td>
<td>F</td>
<td>121</td>
<td>COBOL processor output used as input to SOURCE subcommand of TESTCOB</td>
</tr>
<tr>
<td></td>
<td>PLIOPT</td>
<td>Output</td>
<td>fn</td>
<td></td>
<td>F</td>
<td>121</td>
<td>COBOL processor output used as input to SOURCE subcommand of TESTCOB</td>
</tr>
<tr>
<td></td>
<td>TESTCOB</td>
<td>Output</td>
<td>fn</td>
<td></td>
<td>F</td>
<td>121</td>
<td>COBOL processor output used as input to SOURCE subcommand of TESTCOB</td>
</tr>
<tr>
<td>LKEDIT</td>
<td>LKED</td>
<td>Output</td>
<td>fn</td>
<td></td>
<td>F</td>
<td>121</td>
<td>Listing</td>
</tr>
<tr>
<td>LOGFILE</td>
<td>SET</td>
<td>Output</td>
<td>fn</td>
<td></td>
<td>V</td>
<td>none</td>
<td>Log of data written to virtual screen.</td>
</tr>
<tr>
<td></td>
<td>LOGFILE</td>
<td>Output</td>
<td>fn</td>
<td></td>
<td>V</td>
<td>none</td>
<td>Log of data written to virtual screen.</td>
</tr>
<tr>
<td>LOADLIB</td>
<td>LKED</td>
<td>Output</td>
<td>fn</td>
<td></td>
<td>F</td>
<td>≤260</td>
<td>3704/3705 control program load modules</td>
</tr>
<tr>
<td></td>
<td>ZAP</td>
<td>Output</td>
<td>fn</td>
<td></td>
<td>F</td>
<td>≤260</td>
<td>3704/3705 control program load modules</td>
</tr>
<tr>
<td>MACLIB</td>
<td>GLOBAL</td>
<td>Library</td>
<td>fn</td>
<td></td>
<td>fn</td>
<td></td>
<td>Macro definitions (dictionary and members)</td>
</tr>
<tr>
<td></td>
<td>MACLIB</td>
<td>Library</td>
<td>fn</td>
<td></td>
<td>fn</td>
<td></td>
<td>Macro definitions (dictionary and members)</td>
</tr>
<tr>
<td></td>
<td>MACLIST</td>
<td>Library</td>
<td>fn</td>
<td></td>
<td>fn</td>
<td></td>
<td>Macro definitions (dictionary and members)</td>
</tr>
<tr>
<td>MACRO</td>
<td>ESERV</td>
<td>Input</td>
<td>fn</td>
<td></td>
<td>F</td>
<td>80</td>
<td>Macro definitions</td>
</tr>
<tr>
<td></td>
<td>MACLIB</td>
<td>Input</td>
<td>fn</td>
<td></td>
<td>F</td>
<td>80</td>
<td>Macro definitions</td>
</tr>
</tbody>
</table>

Figure 4 (Part 3 of 5). Reserved Filetypes
<table>
<thead>
<tr>
<th>Filetype</th>
<th>Command</th>
<th>Usage</th>
<th>Filename</th>
<th>Format</th>
<th>RECFM</th>
<th>LRECL</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAP</td>
<td>DOSLIB</td>
<td>Output</td>
<td>libname</td>
<td>F</td>
<td>80</td>
<td></td>
<td>Library map</td>
</tr>
<tr>
<td></td>
<td>DOSLKED</td>
<td>Output</td>
<td>fn</td>
<td></td>
<td></td>
<td></td>
<td>DOS/VS linkage editor map</td>
</tr>
<tr>
<td></td>
<td>DSVR</td>
<td>Output</td>
<td>DSVR</td>
<td></td>
<td></td>
<td></td>
<td>Directory information from DOS/VS private or system libraries</td>
</tr>
<tr>
<td></td>
<td>INCLUDE</td>
<td>Output</td>
<td>LOAD</td>
<td></td>
<td></td>
<td></td>
<td>Module map</td>
</tr>
<tr>
<td></td>
<td>LOAD</td>
<td>Output</td>
<td>LOAD</td>
<td></td>
<td></td>
<td></td>
<td>Module map</td>
</tr>
<tr>
<td></td>
<td>MACLIB</td>
<td>Output</td>
<td>fn</td>
<td></td>
<td></td>
<td></td>
<td>Library Map</td>
</tr>
<tr>
<td></td>
<td>TXTLIB</td>
<td>Output</td>
<td>fn</td>
<td></td>
<td></td>
<td></td>
<td>Library Map</td>
</tr>
<tr>
<td></td>
<td>TAPE</td>
<td>Output</td>
<td>fn</td>
<td></td>
<td></td>
<td></td>
<td>Library Map</td>
</tr>
<tr>
<td>MEMO</td>
<td></td>
<td></td>
<td></td>
<td>F</td>
<td>80</td>
<td></td>
<td>Nonrelocatable object file</td>
</tr>
<tr>
<td>MODULE</td>
<td>GENMOD</td>
<td>Output</td>
<td>fn</td>
<td></td>
<td>V</td>
<td></td>
<td>Nonrelocatable object file</td>
</tr>
<tr>
<td></td>
<td>LOADMOD</td>
<td>Input</td>
<td>fn</td>
<td></td>
<td></td>
<td></td>
<td>Nonrelocatable object file</td>
</tr>
<tr>
<td></td>
<td>MODMAP</td>
<td>Input</td>
<td>fn</td>
<td></td>
<td></td>
<td></td>
<td>Nonrelocatable object file</td>
</tr>
<tr>
<td>NAMES</td>
<td>NAMEFIND</td>
<td>Input/Output</td>
<td>userid</td>
<td>V</td>
<td>255</td>
<td></td>
<td>Information about users in communication</td>
</tr>
<tr>
<td></td>
<td>NAMES</td>
<td>Input/Output</td>
<td>userid</td>
<td>V</td>
<td>132</td>
<td></td>
<td>Creates a note to be sent to others.</td>
</tr>
<tr>
<td>NETLOG</td>
<td>RECEIVE</td>
<td>Logging</td>
<td>userid</td>
<td>V</td>
<td>255</td>
<td></td>
<td>Records logging transmission of files sent or received</td>
</tr>
<tr>
<td>NOTE</td>
<td>NOTE</td>
<td>Input/Output</td>
<td>userid</td>
<td>V</td>
<td>132</td>
<td></td>
<td>Creates a note to be sent to others.</td>
</tr>
<tr>
<td>NOTEBOOK</td>
<td>RECEIVE</td>
<td>Input</td>
<td>userid</td>
<td>V</td>
<td>132</td>
<td></td>
<td>Notes sent to or received by you</td>
</tr>
<tr>
<td>PLI or</td>
<td>PLOPT</td>
<td>Input</td>
<td>fn</td>
<td></td>
<td></td>
<td></td>
<td>PL/I source statements</td>
</tr>
<tr>
<td>PLOPT</td>
<td>PLIC</td>
<td>Input</td>
<td>fn</td>
<td></td>
<td></td>
<td></td>
<td>PL/I source statements</td>
</tr>
<tr>
<td>PLOPT</td>
<td>PLICR</td>
<td>Input</td>
<td>fn</td>
<td></td>
<td></td>
<td></td>
<td>PL/I source statements</td>
</tr>
<tr>
<td>PROC</td>
<td>PSERV</td>
<td>Output</td>
<td>fn</td>
<td></td>
<td>F</td>
<td>80</td>
<td>A procedure from the DOS/VS procedure library</td>
</tr>
<tr>
<td>REPOS</td>
<td>GENMSG</td>
<td>Input</td>
<td>DMKINES</td>
<td>F</td>
<td>80</td>
<td></td>
<td>Source statements for message repositories.</td>
</tr>
<tr>
<td>RTABLE</td>
<td>PROP</td>
<td>Input</td>
<td>fn</td>
<td></td>
<td>V</td>
<td>72</td>
<td>Routing table for Programmable Operating Facility.</td>
</tr>
<tr>
<td>SCRIPT</td>
<td>SCRIPT1</td>
<td>Input</td>
<td>fn</td>
<td></td>
<td>V</td>
<td>132</td>
<td>Input to SCRIPT processor</td>
</tr>
<tr>
<td>SYMDMP</td>
<td>FCOBOL</td>
<td>Output</td>
<td>fn</td>
<td></td>
<td>V</td>
<td>512</td>
<td>DOS/VS COBOL DEBUG file for SYMDMP option</td>
</tr>
<tr>
<td>SYNONYM</td>
<td>SYNONYM</td>
<td>Reference</td>
<td>fn</td>
<td>F</td>
<td>80</td>
<td></td>
<td>Command name synonyms</td>
</tr>
<tr>
<td>SYSUT1</td>
<td>ASM3705</td>
<td>Work</td>
<td>fn</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SYSUT2</td>
<td>ASSEMBLE</td>
<td>Work</td>
<td>fn</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SYSUT3</td>
<td>COBOL</td>
<td>Work</td>
<td>fn</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>LKED</td>
<td>Work</td>
<td>fn</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>PLOPT</td>
<td>Work</td>
<td>fn</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SYSUT4</td>
<td>COBOL</td>
<td>Work</td>
<td>fn</td>
<td></td>
<td>F</td>
<td>80</td>
<td>Used as input to TESTCOB</td>
</tr>
<tr>
<td></td>
<td>LKED</td>
<td>Work</td>
<td>fn</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>PLIC</td>
<td>Work</td>
<td>fn</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>PLICR</td>
<td>Work</td>
<td>fn</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>TESTCOB</td>
<td>Input</td>
<td>fn</td>
<td></td>
<td></td>
<td>512</td>
<td></td>
</tr>
<tr>
<td>TESTFORT</td>
<td>TESTFORT</td>
<td>Output</td>
<td>fn</td>
<td>VB</td>
<td>125</td>
<td></td>
<td>Processor printed output</td>
</tr>
</tbody>
</table>

Figure 4 (Part 4 of 5). Reserved Filetypes
<table>
<thead>
<tr>
<th>Filetype</th>
<th>Command</th>
<th>Usage</th>
<th>Filename</th>
<th>Format</th>
<th>RECFM</th>
<th>LRECL</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>TEXT</td>
<td>ASSEMBLE ASM3705</td>
<td>Output</td>
<td>fn</td>
<td>F</td>
<td>80</td>
<td>Object code 3704/3705 source code and job control language statements</td>
<td></td>
</tr>
<tr>
<td></td>
<td>COBOL DOSLKED GEN3705</td>
<td>Output</td>
<td>fn</td>
<td>fn</td>
<td>fn(Ln)</td>
<td>Object code</td>
<td></td>
</tr>
<tr>
<td></td>
<td>INCLUDE LKED</td>
<td>Input</td>
<td>fn</td>
<td>fn</td>
<td>fn</td>
<td>Object code</td>
<td></td>
</tr>
<tr>
<td></td>
<td>LOAD PLOPT TXTLIB GOFORT FORTGI FORTHX RSERV TEXTFORT</td>
<td>Input</td>
<td>fn</td>
<td>fn</td>
<td>fn</td>
<td>Object code</td>
<td></td>
</tr>
<tr>
<td></td>
<td>TXTlangid GENMSG</td>
<td>Output</td>
<td>fn</td>
<td>F</td>
<td>80</td>
<td>Object code for language files.</td>
<td></td>
</tr>
<tr>
<td>TXTLIB</td>
<td>GLOBAL TXTLIB</td>
<td>Library</td>
<td>fn</td>
<td>fn</td>
<td>fn</td>
<td>Object decks (dictionary and members)</td>
<td></td>
</tr>
<tr>
<td>UPDATE</td>
<td>UPDATE</td>
<td>Input</td>
<td>fn</td>
<td>F</td>
<td>80</td>
<td>UPDATE control cards</td>
<td></td>
</tr>
<tr>
<td>UPDLOG</td>
<td>UPDATE</td>
<td>Input</td>
<td>fn</td>
<td>F</td>
<td></td>
<td>UPDATE log</td>
<td></td>
</tr>
<tr>
<td>VSBASIC</td>
<td>VSBASIC</td>
<td>Input</td>
<td>fn</td>
<td>F</td>
<td>≤256</td>
<td>VSBASIC language source statements</td>
<td></td>
</tr>
<tr>
<td>VSBDATA</td>
<td>VSBDATA</td>
<td>Execution time files</td>
<td>fn</td>
<td>V</td>
<td>≤140</td>
<td>VSBASIC user input/output files</td>
<td></td>
</tr>
<tr>
<td>UPDTxxxx</td>
<td>UPDATE</td>
<td>Input</td>
<td>fn</td>
<td>F</td>
<td>80</td>
<td>UPDATE control statements</td>
<td></td>
</tr>
<tr>
<td>XEDIT</td>
<td>XEDIT</td>
<td>Input</td>
<td>fn</td>
<td>V</td>
<td>255</td>
<td>EXEC/XEDIT statements</td>
<td></td>
</tr>
<tr>
<td>ZAP</td>
<td>ZAP ZAPTEXT</td>
<td>Input</td>
<td>fn</td>
<td>F</td>
<td>80</td>
<td>Control records that modify or dump files</td>
<td></td>
</tr>
</tbody>
</table>

Figure 4 (Part 5 of 5). Reserved Filetypes

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

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

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

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

CMS Return Codes

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

<table>
<thead>
<tr>
<th>Return Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-0001</td>
<td>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.)</td>
</tr>
<tr>
<td>-0002</td>
<td>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).</td>
</tr>
<tr>
<td>-0003</td>
<td>No CMS command with this name was found.</td>
</tr>
<tr>
<td>-0004</td>
<td>The LOADMOD failed (for example, there was an error on the module).</td>
</tr>
<tr>
<td>-0005</td>
<td>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).</td>
</tr>
<tr>
<td>0</td>
<td>Normal.</td>
</tr>
<tr>
<td>1</td>
<td>Device disconnected.</td>
</tr>
<tr>
<td>1</td>
<td>Top or bottom of virtual screen reached.</td>
</tr>
<tr>
<td>3</td>
<td>Virtual screen, window, or queue already exists.</td>
</tr>
</tbody>
</table>

Figure 5 (Part 1 of 2). Return Codes Produced by CMS
<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>Data, field, or scroll amount is truncated.</td>
</tr>
<tr>
<td>4</td>
<td>List or queue is empty.</td>
</tr>
<tr>
<td>4</td>
<td>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.</td>
</tr>
<tr>
<td>8</td>
<td>Device errors occurred for which a warning message is issued, or errors have been introduced into the output file.</td>
</tr>
<tr>
<td>12</td>
<td>Errors were found in the input file.</td>
</tr>
<tr>
<td>13</td>
<td>No space available.</td>
</tr>
<tr>
<td>14</td>
<td>No reserved or data area.</td>
</tr>
<tr>
<td>20</td>
<td>Window name of &quot;*&quot; or &quot;=&quot; not allowed.</td>
</tr>
<tr>
<td>20</td>
<td>An invalid character is in the fileid. Valid characters are: 0-9, A-Z, a-z, $, @, #, +, -(hyphen),:(colon),_: (underscore).</td>
</tr>
<tr>
<td>24</td>
<td>CMS virtual screen or window cannot be deleted.</td>
</tr>
<tr>
<td>24</td>
<td>The user did not specify the command line correctly.</td>
</tr>
<tr>
<td>28</td>
<td>Virtual screen, window, or queue not defined.</td>
</tr>
<tr>
<td>28</td>
<td>An error occurred while trying to access, or manipulate, a user's files; for example, file not found.</td>
</tr>
<tr>
<td>32</td>
<td>Invalid position specified.</td>
</tr>
<tr>
<td>32</td>
<td>The user's file is not in the expected format, or the user's file does not contain the expected information.</td>
</tr>
<tr>
<td>36</td>
<td>Window not connected or displaying virtual screen.</td>
</tr>
<tr>
<td>36</td>
<td>No field to write data/color/exthi/PSset.</td>
</tr>
<tr>
<td>36</td>
<td>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.</td>
</tr>
<tr>
<td>40</td>
<td>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).</td>
</tr>
<tr>
<td>41</td>
<td>Insufficient storage was available to execute the command.</td>
</tr>
<tr>
<td>88</td>
<td>TTY device.</td>
</tr>
<tr>
<td>88</td>
<td>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.</td>
</tr>
<tr>
<td>100</td>
<td>I/O error.</td>
</tr>
<tr>
<td>100</td>
<td>Input/output device errors.</td>
</tr>
<tr>
<td>104</td>
<td>Insufficient storage.</td>
</tr>
<tr>
<td>104</td>
<td>A functional error for which the system is responsible occurred during execution of the command.</td>
</tr>
<tr>
<td>256</td>
<td>Request rejected by IUCV.</td>
</tr>
<tr>
<td>256</td>
<td>All unexpected errors for which the system is responsible; that is, terminal error messages.</td>
</tr>
</tbody>
</table>

**Figure 5 (Part 2 of 2). Return Codes Produced by CMS**
CP DIRECT Command Return Codes

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

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Invalid filename, or file not found.</td>
</tr>
<tr>
<td>2</td>
<td>Error loading the directory.</td>
</tr>
<tr>
<td>3</td>
<td>Invalid option from CMS.</td>
</tr>
<tr>
<td>4</td>
<td>Directory not swapped, user not privilege class A, B, or C.</td>
</tr>
<tr>
<td>5</td>
<td>Directory not swapped, system (old) directory locked.</td>
</tr>
<tr>
<td>6</td>
<td>Directory not swapped, the directory in use by the system is not the updated directory.</td>
</tr>
<tr>
<td>1xx</td>
<td>Error in the CMS RDBUF routine.</td>
</tr>
<tr>
<td>2xx</td>
<td>Error in the CMS TYPLIN routine.</td>
</tr>
</tbody>
</table>

Figure 6. Return Codes Produced by the CP DIRECT Command

1 xx is the CMS routine return code.
CMS DDR Command Return Codes

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

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Invalid filename, or file not found.</td>
</tr>
<tr>
<td>2</td>
<td>Error in executing the program.</td>
</tr>
<tr>
<td>3</td>
<td>Flagged DASD (Direct Access Storage Device) track.</td>
</tr>
<tr>
<td>4</td>
<td>Permanent tape or DASD I/O error.</td>
</tr>
<tr>
<td>1xx</td>
<td>Error in the PRINTIO routine.</td>
</tr>
<tr>
<td>2xx</td>
<td>Error in the CONREAD routine.</td>
</tr>
<tr>
<td>3xx</td>
<td>Error in the RDBUF routine.</td>
</tr>
<tr>
<td>4xx</td>
<td>Error in the TYPLIN routine.</td>
</tr>
</tbody>
</table>

Figure 7. Return Codes Produced by the CMS DDR Command

1 xx is the CMS routine return code.

System Product Editor (XEDIT) Command Return Codes

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

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-3</td>
<td>Unknown command; Invalid from environment other than EXEC 2 or REXX</td>
</tr>
<tr>
<td>-2</td>
<td>Invalid subset command</td>
</tr>
<tr>
<td>-1</td>
<td>Incorrect operands specified in the PARSE macro</td>
</tr>
<tr>
<td>0</td>
<td>Normal; Parsing was successful; &quot;n&quot; lines were inserted</td>
</tr>
</tbody>
</table>

Figure 8 (Part 1 of 3). Return Codes Produced by the XEDIT Command
<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>TOF or EOF reached (and displayed) during execution or change; No change (SPLITJOIN issued at TOF or EOF); Duplicate name defined; Valid only in display mode; Column pointer outside restored zone settings; Only one file edited; Parsing incomplete - scanned line does not match PARSE macro format; No action taken - cursor will be set outside screen; Out of zone definition during execution; Partial delete because EOF or TOF reached reached during execution; File has been filed, was only one edited; Overlapping groups of lines; Target line within lines to move; No line(s) changed or cursor not on valid data field; Total number of lines or columns exceeds physical screen size</td>
</tr>
<tr>
<td>2</td>
<td>Target line not found; Name does not exist for OFF function</td>
</tr>
<tr>
<td>3</td>
<td>Terminal is not a display terminal; Truncated or spilled; Invalid placement of cursor or subcommand; SORT cannot be used when a file is edited in UPDATE or extended mode; Operand or subcommand is valid only for display terminal. File already exists; RGLEFT valid in display mode only; No PRESERVE has been issued; Pool of deleted lines is empty; &quot;QUERY POINT&quot; issued, but no symbolic names defined; Macro not currently in storage; LOAD has already been issued; Subcommand is not valid in extended mode or records truncated</td>
</tr>
<tr>
<td>4</td>
<td>Insufficient storage available; No lines changed; Each logical screen must contain at least 5 lines and 20 columns; Line is not reserved; Lrecl must be lower than 65536 for recfm V; File already in storage. Too many control characters defined; Invalid when issued from prefix macro; Macro is in use - do not purge; No change occurred (string1 not found):</td>
</tr>
<tr>
<td>5</td>
<td>Invalid or missing operand, string, or (line) number</td>
</tr>
<tr>
<td>6</td>
<td>Subcommand rejected in the profile because of LOAD error or QUIT issued in macro</td>
</tr>
<tr>
<td>7</td>
<td>Error building the update file</td>
</tr>
<tr>
<td>8</td>
<td>Prefix area contains pending subcommand or macro; I/O error or modifications lost because PA key pressed when message pending</td>
</tr>
<tr>
<td>12</td>
<td>Disk defined in filemode is read-only; File has been changed; Use QQUIT to QUIT anyway</td>
</tr>
<tr>
<td>13</td>
<td>Disk is full</td>
</tr>
<tr>
<td>16</td>
<td>EXEC 2 variable greater than 256 characters</td>
</tr>
<tr>
<td>20</td>
<td>Invalid character in filename or filetype</td>
</tr>
<tr>
<td>24</td>
<td>Invalid filemode, Invalid parameters, or options; Invalid columns defined</td>
</tr>
<tr>
<td>28</td>
<td>Source file not found (UPDATE MODE). or library not found (MEMBER option), or specified profile macro does not exist. or file XEDTEMP CMSUTI already exists; Filename already exists</td>
</tr>
<tr>
<td>32</td>
<td>Error during updating process; Record &quot;firstrec&quot; is beyond end of file; File is not a library; library has no entries; File is not fixed, 80 character records</td>
</tr>
<tr>
<td>36</td>
<td>Disk not accessed yet</td>
</tr>
<tr>
<td>40</td>
<td>No list given</td>
</tr>
<tr>
<td>80</td>
<td>Unsupported OS data set</td>
</tr>
<tr>
<td>81</td>
<td>Unsupported OS data set</td>
</tr>
<tr>
<td>82</td>
<td>Unsupported OS data set</td>
</tr>
<tr>
<td>83</td>
<td>Unsupported OS data set</td>
</tr>
<tr>
<td>88</td>
<td>File is too large. cannot fit into storage; or previous MACLIB function not found</td>
</tr>
</tbody>
</table>

Figure 8 (Part 2 of 3). Return Codes Produced by the XEDIT Command
<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>Error reading/writing file to disk; Error from rdbuf; Error occurred while creating the file</td>
</tr>
<tr>
<td>104</td>
<td>No storage is available</td>
</tr>
<tr>
<td>nn</td>
<td>Command's RC specified as operand; Same as repeated subcommand's return codes; Return code of CMS or CP command; Return code of subcommand or macro or from subcommand following LOCATE</td>
</tr>
<tr>
<td>any number &gt;10</td>
<td>Standard CMS HELP command return codes</td>
</tr>
</tbody>
</table>

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

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

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>An incorrectly entered parameter.</td>
</tr>
<tr>
<td>8</td>
<td>System failure; a read/write error or an invalid internal parameter.</td>
</tr>
</tbody>
</table>

Figure 9. Return Codes Produced by the IPCS Commands

Example of a Return Code from a CP Command

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

```
ipl cms

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

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

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

or

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

The return code may be used by a system programmer in the DEBUG subcommand and also in EXEC procedures. See the VM/SP CMS Command Reference or EXEC 2 Reference for a description of the ‘RC’ special variable.
Chapter 3. Summary of VM/SP Commands and Service Aids

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

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

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

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

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

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

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

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

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

brackets [ ]  
indicate optional choices, one of which may be selected.

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

vertical bar |  
separates the operand alternatives within the brackets and braces.

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

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

Figure 10. Syntax Abbreviations Used

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

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

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

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

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

**Class B  System Resource Operator**
The class B user controls allocation and deallocation of all the real resources of the VM/SP system, except those
controlled by the primary system operator and spooling operator.

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

Class D  Spooling Operator
The class D user controls spool data files and specific functions of the system’s unit record equipment.

Class E  System Analyst
The class E user displays the contents of real storage, performs the functions required to generate saved systems and 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.

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

Class G  General User
The class G user controls functions associated with the execution of his virtual machine. A general user cannot display or modify real storage.
**Class Any** The Any classification is given to certain CP commands that are available to any user. These are primarily for the purpose of gaining and relinquishing access to the VM/SP system.

**Class H** Reserved for IBM use.
Chapter 4. CP, CMS, GCS, TSAF, RSCS, and IPCS Commands

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

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

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

* CMS, CP CLASS ANY, GCS, RSCS

Permits comments

* anycomment

#CP CP Class Any

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

#CP [commandline1 [#commandline2 #...]]
ACCESS

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...[]]]
```

options:

- [NOPROF]
- [ERASE]
- [SAVEONLY]
- [NODISK]
- [NOSAVE]

ACCESS

GCS

Identifies CMS or VSAM disks that an application will use.

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

```
ACCESS [CUU mode [/ext [fn [ft [fm]]]]]
```

ACNT

CP Class A

Creates accounting records.

```
ACNT {userid1 [userid2...]
    [ALL [CLOSE]
    [CLOSE]
    }
```

54 VM/SP Quick Reference
ADD LINK

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

ADD LINK vdev

ADSTOP
Halts the virtual machine’s execution.

ADSTOP \{hexloc\}
  \{OFF\}

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

ALARM VSCREEN vname

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

AMserv fn1 [fn2|fnl][[options...]]

options:

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

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

ASMGEND

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

ASM3705

Invokes the 3705 assembler.

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

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

options:

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

Invokes the system assembler.

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

listing control options:

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

output control options:

- [DECK|NODECK] [OBJECT|NOBJECT]
- [TEST|NOTEST]

SYSTEM options:

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

other options:

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

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

ASSGN SYSxxx

\[
\begin{align*}
\text{Reader} & \quad \{(\text{options...[ ]})\} \\
\text{PUnch} \\
\text{PRinter} \\
\text{Terminal} \\
\{ \text{TAP } [n] \} \\
\text{mode} \\
\text{IGN} \\
\text{UA}
\end{align*}
\]

options:

\[
[\text{UPCASE}|\text{LOWCASE}] [7\text{TRACK}|9\text{TRACK}] [\text{RTCH } a] \quad [\text{DEN den}]
\]

ATTACH

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

ATTach

\[
\begin{align*}
\{ \text{raddr TO} \} \{ \text{userid [AS] vaddr [R/O]} \} \{ 3330V \} \{ \text{VOLID valid} \} \\
\{ \text{SYSTEM [AS] valid} \} \\
\{ \text{raddr...} \} \\
\{ \text{raddr-raddr TO} \} \{ \text{userid [R/O]} \} \{ 3330V \} \\
\{ \text{SYSTEM [AS] valid} \} \\
\{ \text{Laddr TO} \} \{ \text{userid [AS] vaddr} \} \\
\{ \text{CHANnel c [PROC nn] TO} \} \{ \text{userid} \}
\end{align*}
\]

ATTN

Makes attention interruption pending.

ATTN
AUTOLOG

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

AUTOLOG userid password [variable data]

B

CMS Border Command

Scrolls the window backward.

BACKSPAC

Repeats or repositions a current spool file.

Printer Format: Punch Format:

BACKSPAC raddr File

BACKSPAC lprt pages [EOF] BBackspac raddr [File]

BACKSPAC

RSCS

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

BACKSPAC [linkid] [File|nnn]

BEGIN

Starts the execution of a virtual machine.

BEGIN [hexloc]
C (CMS Border Command)

C

Clears the window of scrollable data.

CATCHECK CMS

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

CATCHECK [catname]

60 VM/SP Quick Reference
CHANGE

Alters the attributes of a closed spool file.

```
CHANGE [userid] [Reader] [Class cl] [Class c2] [Form form1] [form2] [Chars name0] [Chars name1] [Chars name2] [Chars name3] [Class c2] [Copy[*]nnn] [Dist distcode] [FCB name] [Flash name nnn] [FORM form1] [form2] [DEST dest1] [dest2] [HOLD] [NOHOLD] [MODify name[n]] [SYS] [NOSYS] [UNCONV] [NAmes{fn[ft]} dsname]
```

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

2 The CHars, FCB, and MODify options are valid for only the 3800 printer.
CHANGE

Alters the attributes of a closed spool file.

CHANGE

{ Reader { Class cl spoolid Printer } }
{ Punch } FORM form1
{ DEST dest1 } HOLD
{ ALL } NOHold

DIst dist
COpy [*]nnn
FLash name nnn

NAME { fn [ft] }

dsname

MOdify name [n]
CHars name1
{ [CHars]name2 }
{ [CHars]name3 }
{ [CHars]name4 }

FCB name
FORM form2
DEST dest2
UNCONV

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

Alters one or more attributes of an inactive spool file.

General User Format:

CHange [*]spoolid options...

Operator Format:

CHange [linkid] spoolid options...

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

- [PRIority nn ]
- [CLass c ]
- [COpy [*]nnn ]
- [D1st distcode ]
- [FLash name nnn ]
- [HOld |NOHold ]
- [MODify name [trc] ]
- [CHars name1 [CHars name2 ...] ]
- [FCB name ]
- [FOrm cccccccc ]
- [DEST [cccccccc|OFF]]

Name {fn [ft]|dsname}

CLEAR VSCREEN

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

CLEAR VSCREEN  vname

CLEAR WINDOW

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

CLEAR WINDOW [wname]
CLOSE

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

Close

Cmd nodeid [command text]

CMDCALL

Converts EXEC 2 extended PLIST function calls to CMS extended or standard PLIST command calls.

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

CMSBATCH

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

CMSBATCH [sysname]
CMSGEND

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

\[
\text{CMSGEND fn [CTLCMS CTLLLALL NOCLEAR MAP NOINV] [MODE fn]}
\]

CMSSERV

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

COMMANDS

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

COMParare

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

\[
\text{COMpare fileid1 fileid2 [(option[])]}
\]

option:

\[
\text{COL [mmm[-]nnn] [1 lrecl]}
\]
**CONVERT**

**CONVERT**  
IPCS

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

`CONVERT fn [ft [fm]]`

---

**CONVERT COMMANDS**  
CMS

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

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

**options:**

- `[SYMptom|USER|ALL]`
- `[CHeck|OUTmode [*|fm]]`
- `[STACK [FIFO|LIFO]`
  - `FIFO`
  - `LIFO`

---

**CONVIPCS EXEC**  
IPCS

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

`CONVIPCS`

---

**CONWAIT**  
CMS

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

`CONWAIT`
COPYFILE

Copies files according to operand specifications.

COPYfile  fileid1 [fileid2...] [fileido] [[options...[]]]

options:

[Type] [NEWDate] [NEWFile] [Prompt]
[NOType] [OLDDate] [REPlace] [NORrompt]

[FROM recno] [FOR numrec] [SPecs]
[FRLable xxxxxxxx] [TOLlabel xxxxxxxx] [OVly]
[NOSpecs] [APpend]

[RECfm {F} [LRecl nnnnn] [TRUnc]
[V] [NOTRunc] [PAck]
[TRUnc] [UNPack] [Fill c]
[Fill hh] [Fill 40]

[EBcdic] [UPcase] [TRAns] [SIngle]
[LOwcase]

COUPLE

Connects virtual channel to channel adapters.

COUPLE  vaddr1 [TO] userid vaddr2

CP

Permits entry of CP commands without leaving the CMS environment.

CP [commandline]

CP

Permits execution of CP commands within your privilege class.

CP [commandline1 [#commandline2 #...]]
**CP**

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

**CPQUERY**

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

```plaintext
CPQuery
   { INDicate
     LOGmsg
     Names
     Time
     Users [userid]
     CPUid
     CPLEVEL }
```
CPTRAP

CP Class C

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

```
CPTrap typenum [VmblOk DEVaddr cuu CODE code-value OFF]
ALL [ON OFF]
ALLOWid userid
GROUPid group-name
STArt [TO] {userid} [WRAP wrap-size]
CLOSE
STOP
```

CURSOR VSCREEN

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

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

```
options: [ Reserved Data ]
```

D

CMS Border Command

Drops the window.

```
D
```
DCP

Displays real processor storage on the terminal.

DCP 
MLhexloc1 { - }\{hexloc2\} \{ END \} \{ NLhexloc1 \} \{ MThexloc1 \} \{ NThexloc1 \} \{ Mhexloc1 \} \{ Nhexloc1 \} \{ hexloc1 \} \{ END \} \{ bytecount \} \{ hexloc1 \} \{ END \} \{ Thexloc1 \} \{ Lhexloc1 \} \{ Thexloc1 \} \{ Lhexloc1 \}

DDR

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

DDR [fn ft [fm*]]

I/O definition statements:

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

options:

\{ REWind \} \{ MMode 800 \} \{ SKip nn \} \{ Compact \}
\{ LEave \} \{ MMode 1600 \} \{ SKip 0 \}
\{ UNload \} \{ MMode 6250 \} \{ MMode 38K \}
SYSPRINT control statement:

SYsprint [cuu|CONS]

function control statements:

\[
\begin{align*}
\{ \text{DUmp} \} & \quad [\text{FTr}] \quad \{ \text{cy11 [To] [cy12 [Reorder] [To] [cy13]]} \} \\
\{ \text{Copy} \} & \quad \{ \text{block1[To][block2[Reorder[]To][block3]} \} \\
\{ \text{REstore} \} & \\
\text{CPvol} & \\
\text{NUcleus} & \\
\text{ALL} & \\
\end{align*}
\]

PRINT/TYPe function statements:

\[
\begin{align*}
\{ \text{PRINT} \} & \quad \{ \text{cy11 [hh1 [rr1]]To cy12 [hh2[rr2]][(options...[]]} \} \\
\{ \text{TYpe} \} & \quad \{ \text{block1[To block2]} \} \\
\end{align*}
\]

options:

[Hex] [Graphic] [Count]

1 The FTr option is valid only with the DUMP control statement.
Enters the DEBUG environment to test and debug a program.

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

**BReak**
\[
\text{BReak} \{ \text{id symbol} \}
\]
Segue program execution at a specific instruction location.

**CAW**
Displays the Channel Address Word (CAW).

**CSW**
Displays the Channel Status Word (CSW).

**DEFine**
\[
\text{DEFine symbol hexloc [bytecount]}
\]
Assigns a symbolic name to a specific storage address.

**DUmp**
\[
\text{DUmp} \begin{bmatrix}
\text{symbol1} & \text{symbol2} \\
\text{hexloc1} & \text{hexloc2} \\
0 & 32
\end{bmatrix}
\]
Dumps the contents of storage locations to the virtual printer.

**GO**
\[
\text{GO} \{ \text{symbol} \}
\]
Exits from the DEBUG environment and begins program execution.
GPR reg1 [reg2]
   Displays the contents of the specified general registers.

HX
   Returns to CMS environment.

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

PSW
   Displays old PSW.

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

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

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

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

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

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

Redefines the status of a 3330V volume.

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

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

DEFine
  Reader [AS] VADDR
  Printer 1403, 1443 ... 4248

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

Printer
  PUnch
  CONsole
  CTCa
  TIMER
  1403
  1443
  2501
  2540P
  2540R
  3088
  3203
  3211
  3262
  3289E
  3505
  3525
  4245
  4248

Chapter 4. CP, CMS, GCS, TSAF, RSCS, and IPCS Commands 75
DEFINE (continued)

CP CLASS G

\[
\begin{align*}
\{3800 & \} \quad [As] \quad vaddr \quad [Size \quad ww \quad 11] \quad [2WCGM] \quad [BTS] \quad [DATck] \\
3800-1 & \} \quad [4WCGM] \quad [CFS] \quad [NODatck] \\
3800-3 & \\
\{CHANNels \} \quad [As] \quad \{SEL\} \\
BMX & \\
\{LIne \} \quad [As] \quad vaddr \quad [TELE[2]] \\
IBM[1] & \\
vaddr1 & \quad [As] \quad vaddr2 \\
\{GRAF \} \quad cuu \quad \{3033\} \\
\quad \quad \quad 3036 \\
\quad \quad \quad 3138 \\
\quad \quad \quad 3148 \\
\quad \quad \quad 3158 \\
\quad \quad \quad 3270 \\
\{TFB-512\} \\
{T3310} & \quad [As] \quad vaddr \quad [BLK] \quad [nnnnn] \\
T3370 & \\
\{T2305\} \\
T2314 & \\
T2319 & \\
{T3330} & \quad [As] \quad vaddr \quad [CYL] \quad [nnn] \\
T3340 & \\
T3350 & \\
T3375 & \\
T3380 & \\
\{STORage \} \quad [As] \quad \{nnnnK\} \\
\quad \quad \quad nnM \\
\end{align*}
\]
DEFINE RSCS

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

```
DEFINE 1nkid [ASTart|NOASTart]
   [Class  c]
   [DP    dpriority]
   [KEEP  holdslot]
   [LINE  vaddr]
   [LOGMode logmodename]
   [LUName luname]
   [Queue (Priority|Fifo|Size)]
   TYPE
      {NJE
       SNANJE
       RJE
       MRJE
       3270P
       SNA3270P}
   [Parm [parameters...]]
```

DEFINE VSCREEN CMS

Creates a virtual screen.

```
DEFINE VSCREEN  vname lines cols rtop rbot [(optionA optionB optionC optionD[])]
```

- **optionA:**
  - **TYPE**
  - **NOType**

- **optionB:**
  - **PProtect**
  - **NOPProtect**
  - **NHgh**
  - **NOHgh**

- **optionC:**
  - **color**
  - **exthi**
  - **PSset**

- **optionD:**
  - **USER**
  - **SYstem**
DEFINE WINDOW

DEFINE WINDOW

Creates a window.

DEFINE WINDOW wname lines cols pslines pscol [(options...[])]

options: [VARIABLE] [BORDER] [POP]

[FIXED] [NO BORDER] [NOPop]

[TOP] [USER]

[NOTop] [SYSTEM]

DELETE

DELETE

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

DELETE linkid

DELETE LINK

DELETE LINK

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

DELETE LINK vdev

DELETE VSCREEN

DELETE VSCREEN

Removes a virtual screen definition.

DELETE VSCREEN vname
DELETE WINDOW

Removes a window definition.

DELETE WINDOW

DELete WINdow  wname

DESBUF

Clears the console and program stack input and output buffers.

DESBUF

DETACH

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

DETach

DETach

DETach {vaddr [vaddr...]}

DETach {vaddr-vaddr}

DETACH

Removes a virtual device from the virtual machine.

DETach

DETach ([vaddr [vaddr...]])

DETach ([vaddr-vaddr])
DIAL

Attaches a terminal device to a multiple access system.

DIAL userid [vaddr]

DIRECT

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

DIRECT [fn [ft [fm]]] [(EDIT)]

Control statements:

Account number [distribution]
CLASS classes
Console vaddr devtype [class] [userid]
Dedicate {NETWORK vaddr resource
vaddr rdev }
{ [VOLID] [volser] [3330V] [R/O] }
{ [VOLID] [volser] [3330V] [R/O] }
DIRectory cuu devtype volser [alt - cuu]
Ipl ipIsys [PARM data]

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

Link userid vaddr1 [vaddr2 [mode]]
DIRECT (continued)

Mdisk cuu devtype {cylr  cyls volser [mode pr pw pm]}]
{ T-DISK cyls
blkr  blks

Option {Realtimer} [Ecmode] [Virt=Real] [Acct] [Svcoff] [BMX]
[CPUID bbbbbb] [AFFinity nn] [Isam]
[VMsave] [STFirst] [370E] [Maxconn nnnnn] [MIH]
[DIAG98]
[VCUNOSHR]

SCReen area {color [hilight]} {hilight [color]} ...

SPEcial vaddr devtype [IBM Tele]

Spool cuu devtype [class [ww[l1 [2WCGM [CFS
4WCGM [BTS [DATCK]]]]]]

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

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

DISABLE

CP Class A or B

Prevents communication lines from accessing VM/SP.

DISAble {raddr...}
   Laddr...
   SNA [userid]
   ALL

DISCONN

CP Class Any

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

DISConn [H0ld]
DISCONN

DISCONN

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

DISConn [[LOG|NOLog] [userid]]

DISK

Disk dumps disk files to punched cards and restores disk files.

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

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

DISKMAP

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

DISKMAP filename [filetype]

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

Display

\[
\begin{align*}
\text{hexloc1} & \{;\} \text{ hexloc2} \\
\text{Khexloc1} & \\
\text{Lhexloc1} & \\
\text{Thexloc1} & .
\end{align*}
\]

\[
\text{bytecount} \\
\text{END}
\]

\[
\begin{align*}
\text{Gregl} & \{;\} \text{ reg2} \\
\text{Yregl} & \\
\text{Xregl} & .
\end{align*}
\]

\[
\text{regcount} \\
\text{END}
\]

\[
\begin{align*}
\text{Psw} & \\
\text{CAW} & \\
\text{CSW} &
\end{align*}
\]
In CMS/DOS, defines DOS and CMS sequential disk files for program input/output; identifies DOS files and libraries; defines and identifies VSAM catalogs, clusters, and data spaces; identifies VSAM, DOS, or CMS files uses for VSAM program input/output and access method service functions.

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

```
DLBL [ddname {mode} {CMS fn ft} [CMS FILE ddname]]

DDNAME {mode} {CMS FILE ddname}

DDNAME {mode} {DNS qual1 [.qual2...qualn]}

DDNAME {mode} {DNS qual1 [qual2...qualn]}

DDNAME {mode} {DNS ?}

DUMMY CMS FILE ddname

DUMMY CMS FILE ddname

DUMMY CMS FILE ddname

{ddname CLEAR [optionA optionB optionC ()]}

optionA:  optionB:  optionC:

[SYSxxx]  [PERM]  [VSAM]

[CHANGE]  [EXTENT]  [MULT]

[CAT catdd]

[BUFSP nnnnnn]
```
DLBL

Defines VSAM files used for program input/output.

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

optionB:  optionC:

[PERM]       [VSAM ]
[MULT ]      [CAT catdd ]
[CHANGE ]    [BUFSP nnnnn ]
[NOCHANGE]   

DMCP

CP Class C or E

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

```
DMCP [MLhexloc1] [{:} [hexloc2 END]] [{*dumpid}]
      NLhexloc1
      MThexloc1
      NThexloc1
      Mhexloc1
      Nhhexloc1
      Lhexloc1
      Thexloc1
      hexloc1
      0
```

Chapter 4. CP, CMS, GCS, TSAF, RSCS, and IPCS Commands  85
DOSLIB

DOSLIB

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

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

options:

[ TERM ]
[ DISK ]
[ PRINT ]

DOSLKED

DOSLKED fn [ libname ] [ (options ... []) ]

options:

[ DISK ]
[ PRINT ]
[ TERM ]
DRAIN

**DRAIN**

**CP Class D**

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

```
DRAIN [ Reader
       Printer
       PUnch
       radrr...
       lprr
       ALL
]
```

**DRAIN**

**RSCS**

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

```
DRAIN [linkid]
```

**DROP WINDOW**

**CMS**

Moves a window down in the order of displayed windows.

```
DROP WINDOW { wname
               = WM
               [n]
[+] }
```

**DROPBUF**

**CMS**

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

```
DROPBUF n
```
DSERV

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

DSERV

\[
\begin{align*}
\text{CD} & \left[ \text{PHASE \{name[12]\}} \right] \left[ \text{d2 ... dn} \right] \text{[(options...[])]} \\
\text{RD} & \\
\text{SD} \\
\text{PD} \\
\text{TD} \\
\text{ALL} & \\
\end{align*}
\]

options:

- \text{DISK}
- \text{TERM}
- \text{PRINT}
- \text{SORT}

DUMP

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

DUMP

\[
\begin{align*}
\text{Lhexloc1} & \\
\text{Thexloc1} & \\
\text{hexloc1} & \\
\{ - \} & \left[ \text{hexloc2} \right] \left[ \text{END} \right] \text{[*dumpid]} \\
\{ . \} & \left[ \text{bytecount} \right] \left[ \text{END} \right] \\
\end{align*}
\]
This is an IPCS command that enables you to interactively examine a dump of a Group Control System (GCS) virtual machine existing as a CMS file.

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

- **IUcv**
  - Displays all entries in the IUCV path table.

- **TActive**
  - **[taskid]**
  - **ALL**
  - Displays the task’s active program list.

- **TLoadl**
  - **[taskid]**
  - **ALL**
  - Displays the task load list.

- **TSab**
  - **[taskid]**
  - **ALL**
  - Displays the subpool map and chain header of a task.

- **VMLoadl**
  - Displays information about all NUCCBLKs on a virtual machine’s load list.
DUMPSCAN

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

DUMPSCAN

The DUMPSCAN subcommands, followed by their descriptions, are:

- (null line)  (Common)
  Reissues the previous CHAIN, LOCATE, or SCROLL subcommand.

- ?  (Common)
  Displays last subcommand entered.

- + (increment)  (Common)
- - (decrement)  (Common)
  Adjusts the address pointer and reissues the DISPLAY command.

- &name [subcommand]  (Common)
  Creates a synonym for frequently used subcommands.

- Aregs  (CP dump only)
  Displays the registers, clocks, PSW, CSW, and CAW for the attached or the non-IPL processor (AP).

- ARIOBLOK cuu  (CP dump only)
  Displays the RCHBLOK, RCUBLOK, and RDEVBLOK for the specified device attached to the non-IPL processor in an MP configuration.

- C  (CP dump only)
  Displays the control registers for the failing processor.
DUMPSCAN (continued)

CHain fromhexloc increment endval
(Continued) (Common)
Verifies the chain of homogeneous control blocks that start at the specified location.

CMS
(Common)
Enteres the CMS subset environment.

CMSPoint
(CMS dump only)
Displays the formatted contents of 17 pointers from CMS NUCON.

CORtable hexloc
(CP dump only)
Displays page status and the formatted contents of the CORTABLE entry for the hexadecimal location specified.

Display {hexloc [nnnn]}
{hexloc%
These loc%
}
(Common)
Displays areas in the dump. The actual address or an indirect address may be specified.

DOSPoint
(CMS dump only)
Displays the formatted contents of five pointers used by DOS simulation.

DUMPID
(Common)
Displays a dump identification message and the dumpid information from the VMDUMP command line (if any).

END
(Common)
Ends the session and returns to CMS.
DUMPSCAN

FDISPlay (TSAF dump only)

    [PATH
    SERVice
    CLUSter
    RESOuRCe
    NEIGhbor
    ROUTing
    LINKDef
    LINKCt1 { BSC
               CTCa}
    ALL

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

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

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

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

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

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

MAPA hexloc
    Locates the module that contains the address specified.

MAPN mmmmmmmm
    Searches the load map for an entry point.

Mregs
    Displays the registers, clocks, PSW, CSW, CAW and timers for the main or the IPL processor.

MRIoblok raddr
    Displays the RCHBLOK, RCUBLOK, and RDEVBLOK for the specified device attached to the IPL processor in an MP configuration.

OSPoint
    Displays the formatted contents of three pointers used in OS simulation.

Print
    Prints the displayed data.

PRT
    
    ON
    OFF
    CLOSE

    Prints the displayed data.

QUIT
    Ends the session and returns to CMS.

Regs
    Displays the registers, clocks, PSWs, timers, CSW, and CAW.

RIOblok raddr
    Displays the RCHBLOK, RCUBLOK, and RDEVBLOK for the specified cuu.
### DUMPSCAN (continued)

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Scroll U [HEX]</strong></td>
<td>Repeats the most recent DISPLAY or TRACE subcommand with an adjusted address. SCROLLU displays the preceding screen of data. SCROLL displays the next full screen of data. The HEX/FORMAT subcommands are only valid following trace entry displays.</td>
</tr>
<tr>
<td><strong>SCROLLU</strong></td>
<td>Formats and displays the summary of the symptom record.</td>
</tr>
<tr>
<td><strong>Trace [[FOR]count][FROM fromloc] [HEX]</strong></td>
<td>Displays trace table entries in hexadecimal or formatted display. After the first successful invocation of the TRACE subcommand, you can specify either SCROLL operand to move forward or back through the trace table.</td>
</tr>
<tr>
<td><strong>USERMAP</strong></td>
<td>Adds a user load map to the dump being viewed.</td>
</tr>
<tr>
<td><strong>VIOblok cuu [userid] [OPERATOR]</strong></td>
<td>Displays the VCHBLOK, VCUBLOK, and VDEVBLOK for the specified cuu and userid.</td>
</tr>
<tr>
<td><strong>Vmblok [userid] [SYSSPOOL]</strong></td>
<td>Scans the VMBLOK chain and displays userid, VMBLOK address and status of all logged-on users. Displays formatted information from a specified user’s VMBLOK by specifying the userid operand.</td>
</tr>
<tr>
<td><strong>ECHO</strong></td>
<td>Returns data directly to the terminal a specified number of times.</td>
</tr>
</tbody>
</table>

---

**ECHO**

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

**ECHO**

| nn |
EDIT CMS

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

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

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

options:

[LRECL nn] [NODISP]

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

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

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

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

ALTER char1 char2 \[n \ \{S\]
Scans n records, altering the specified character.
EDIT

EDIT (continued)

AUTOSave [n|OFF]
   Saves the file after the indicated number of changes or displays the current setting.

BACKward [1|n]
   Repositions the current line pointer toward the top of the file.

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

CASE [U|M]
   Translates to uppercase.

Change [/string1[/string2[/
   [n  [G]]] ]]
   Changes string1 to string2.

CMS
   Enters CMS subset command mode.

DELete [n|l|*]
   Deletes n lines or to EOF.

DOWN [n|l]
   Moves the current line pointer to the nth line down from the current line.

DString /[string /]
   Deletes lines from the current line down to (but not including) the line that contains the designated string.
EDIT (continued)

FILE [fn [ft [fm ]]]
    Saves the file edited on disk and returns to CMS mode.

Find [line]
    Searches the file for the specified line.

FMode [fn]
    Resets or displays the filemode.

FName [fn]
    Resets or displays the filename.

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

FOrward [1|n]
    Moves the current line pointer forward n lines.

Getfile {fn} [ft [fm [firstrec] [numrec]]] ¹
    Inserts some or all of the specified file.

¹ Operands are positional.

IMAGE [ON|OFF|CANON]²
    Expands text (including backspace and tab) into line images or displays current settings.

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

EDIT (continued)

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

LINE\text{mode} \quad \begin{bmatrix} \text{LEFT} & \text{RIGHT} & \text{OFF} \end{bmatrix}

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

   \text{3} \quad \text{Line-number editing is the default for USBASIC and FREEFORT files.}

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

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

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

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

PREServe
   Saves current mode settings.

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

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

RE Cf m  [F|V]
   Sets or displays record format.

R En um  [strtno [incrno]]
   Recomputes line numbers.

RE Pe at  [n|\*]
   Executes the following OVERLAY request n times or to EOF.

Re plac e  [line]
   Replaces the current line with "line" or deletes the line and enters input mode.

RE Sto re
   Restores mode settings.

RE T urn
   Returns to EDIT environment.

RE USE   [subcommand]
   Stacks (LIFO) the last EDIT subcommand and executes the stacked subcommands.

SA VE   [fn [ft [fm ]]]
   Saves the file on disk.
EDIT

EDIT (continued)

\{Scroll \{S\{c\{r\}oll\}U[p]\}\} \[*|n|1\]
Displays a number of lines above or below the current line.

SERial \{OFF \ON \ALL \[incr \10 \seq\}\}
Turns serialization on or off in columns 73-80.

SHORT
Enters SHORT error message mode.

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

TABSet \(n1 \[n2 \ldots \ nn\]
Sets the specified tabs.

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

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

Type \[\begin{array}{c}
m \\
* \\
[|n|] \\
1 \\
\end{array}\]
Displays the specified number of lines beginning with the current line.
EDIT (continued)

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

Verify  [ON]  [OFF]  [startcol]  [endcol]
Sets, displays, or resets verify mode.

{X}  [subcommand|n|l]
Assigns to X or Y the given EDIT subcommand.

Zone  [firstcol]  [lastcol]
Sets or displays the columns to be edited.

{n}  [text]
Locates the line when using line-number editing.

ENABLE

Activates communication lines.

Enable  {raddr...  Laddr  SNA [userid]  ALL

Chapter 4. CP, CMS, GCS, TSAF, RSCS, and IPCS Commands  101
ERASE

ERASE

Deletes files from a user's disk.

ERASE \{fn\} \{ft\} \{fm\} [{options...[]}]

options:

[Type|Notype]

ESERV

ESERV

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

ESERV fn

ETRACE

ETRACE

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

ETrace

\[ [\{\{DSP\ EX\ T\ F\ R\ G\ T\ G\ E\ T\ I/O\ P\ R\ G\ S\ I/O\ S\ S\ S\ S\ V\ C\ G\ T\ a\ c\ e\]\{\{OFF\}\}\{\{GRoup\}\}\{\{ALL\}\}\{\{END\}\} ]\]
EXEC

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

[EXEC] fn [args...]

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

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

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

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

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

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

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

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

&CONTINUE
   Provides a branching address for EXEC branch statements.
EXEC

EXEC (continued)

&CONTROL [OFF ERROR ALL CMS] [TIME NOTIME] [NOPACK PACK] [NOMSG MSG]

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

&EMSG mmmnnns [tok1...[tokn]]
Displays a line of tokens to be edited as an error message.

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

&ERROR [executable-statement|&CONTINUE]
Provides error processing.

&EXIT [return-code|0]
Exits from the EXEC file with a specified return code.

&GOTO {TOP|line-number|-label}
Transfers control to a specified location.

&HEX {ON|OFF}
Initiates or terminates hexadecimal conversion in an EXEC procedure.

&IF {tok1} operator {tok2} executable-statement
{&$ &$ &* &*}
 Allows statement execution if the comparison is satisfied. If the comparison is invalid, execution continues with the statement following the &IF statement.

&LOOP {n -label} {m condition}
Describes a loop in an EXEC, and conditions for exit from the loop.
&PUNCH [tok1 tok2 ... tokn]]
  Punches a card with the specified tokens.

&READ [n ARGS VARS[&var1 &var2 ... [&varn]]]

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

&SKIP [n|1]
  Skips subsequent statements.

&SPACE [n|1]
  Displays blank lines at the terminal.

&STACK [FIFO|LIFO|HT|RT]

  Places a line of tokens in the console input stack.

&TIME [ON|OFF|RESET|TYPE]

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

&TYPE [tok1 tok2 ... tokn]]

  Displays the specified tokens at the terminal.
EXEC

EXEC (continued)

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

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

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

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

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

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

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

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

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

\&0
Contains the filename of the EXEC file.

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

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

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

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

\&EXEC
Indicates the filename of the EXEC file.
EXEC (continued)

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

&GLOBALn
Represents the numeric variables &GLOBAL0 through &GLOBAL9.

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

&LINENUM
Indicates the current line number in the EXEC file.

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

&RETCODE
Indicates the return code from the most recently executed command.

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

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

EXec filename [arguments...]
EXEC 2

EXEC 2

Invokes EXEC 2 files.

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

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

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

&0
   Initializes to the first word of the command string passed to the EXEC 2 interpreter. This variable is initialized or maintained automatically.

&1, &2, ...
   Initializes to the arguments arg1, arg2, ..., argn since they themselves are arguments and then are passed to the EXEC 2 file.

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

&BLANK
   Assigns the value of a blank.

&CMDSTRING
   Initialized to the untranslated command string that is passed to the EXEC 2 file.

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

&DATE
   Evaluates true date (primary meridian -- GMT) in the form: YY/MM/DD. See also &TIME, below.)
EXEC 2

EXEC 2 (continued)

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

&FILEMODE
  Initializes to third qualifier of EXEC 2 file.

&FILENAME
  Initializes to first qualifier of EXEC 2 file.

&FILETYPE
  Initializes to second qualifier EXEC 2 file.

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

&LINE,&LINENUM
  Keeps current line number of EXEC 2 file.

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

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

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

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

Note: An asterisk (*), a hyphen (-), or an ampersand (&) starting a command must be given as an argument.
The formats of the EXEC 2 control statements, followed by their formats are:

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

&BEGPRINT
&BEGTYPE

$line1$
$line2$

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

&BEGSTACK

$line1$
$line2$

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

&BUFFER  n  [comment]

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

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

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

&CASE    [U [comment]]
         [M]

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

&COMMAND word1 [word2 ...]

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

&CRASH   [text]

Aids debugging the EXEC 2 interpreter DMSEXE. Intended to to be used by system support people only.

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

Prints a line for each &1, &2, ..., &n argument or variables var1, var2, ..., varn.
&ERROR action
Sets the action to be automatically invoked on return from any command(s) or subcommand(s) that has a nonzero return code.

&EXIT [return-code [comment]]
0

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

&GOTO line-number [comment]
label

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

&IF word1 =|EQ |NE |LT |LE |NG |GT |GE |NL

[word2 executable statement]

Executes the given executable statement if the condition is satisfied; otherwise, proceeds to next statement.
EXEC 2 (continued)

&LOOP  n  m
  label  *
  WHILE  condition
  UNTIL  condition
Loops through the designated operands until specified condition is satisfied.

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

&PRESUME  [&COMMAND
          &SUBCOMMAND  environment]
Presumes that any statement without a beginning ampersand is to be issued to CMS or to the
designated subcommand environment.

&PRINT  [word1  [word2  ...]]
&TYPE
Prints or types a line containing the operand(s) each separated by one blank, or prints or
types a blank line if no operand appears.

&READ
  [n
  1  *
  ARGS
  VAR[S]  [var1  [var2...]]
  STRING  var
Reads from the console stack (if stack is not empty); otherwise, reads from the console the
number of lines indicated, or assigns values as designated.
&RETURN [word] [comment]
   Returns control to the most recently invoked subroutine to which no return has as yet been made.

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

&STACK [[FIFO] [word1 [word2 ...]]]
   Places a line in the program stack that contains word(s) that are separated by one space or stacks a null line if no words are given.

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

&TRACE [ON ER]r [output-action]
   [ALL OFF *]
   Traces commands and subcommands as indicated by the trace setting. Information obtained is passed to the destination determined by output action.

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

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

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

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

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

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

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

&DATATYPE OF [word]
&TYPE OF

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

&DIVISION OF dividend divisor
&DIV OF

Yields a numeric value representing the integral part of the division of the dividend by the divisor.
&LEFT OF word j
Left-justifies word of length j. Truncates or pads with blanks on the right-hand side.

&LENGTH OF [word]
Gives either the number of characters in word or zero if word is not given.

&LITERAL OF [string]
Gives the literal string beginning with character after blank following OF and ending with the last non-blank character.

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

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

&MULTIPLICATION OF i j [k...]
&MULT OF
Yields numeric value that results from the multiplying of given numeric signed or unsigned words.

Note: There must be at least two of these.

&PIECE OF word i [j]
&SUBSTR OF
Extracts part of word starting at character i for length j or to end of word.

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

&POSITION OF word [word1 [word2 ...]]
Compares and tries to match word with word1, word2, ... If match occurs, gives numeric value of position of matching word. If no match is made or if there is no word(s) with which to compare, the result is zero.
EXEC 2 (continued)

&\text{RANGE} \text{ OF} \ stem \ i \ j
Yields a string comprising of words made by appending numbers to the stem ranging from \( i \) to \( j \) with one blank between each or, if \( i \) is greater than \( j \), yields a null string.

\textbf{Note:} Appended numbers are stripped of any plus sign or redundant leading zeros.

&\text{RIGHT} \text{ OF} \ word \ j
Right-justifies word of length \( j \). Truncates or pads with blanks on left-hand side.

&\text{STRING} \text{ OF} \ [\text{string}]
Gives the string beginning with character after blank following \text{OF} and ending with last non-blank character.

&\text{TRANSLATION} \text{ OF} \ \text{word1} \ [\text{word2} \ [\text{word3}\]]
&\text{TRANS OF}
Compares each character in \text{word1} with \text{word2}. If a match is found, the position of that matching character in \text{word1} is replaced with the character in the same position from \text{word3}.

&\text{TRIM} \text{ OF} \ [\text{word}]
If word has trailing blanks, removes them. If word is not given, result is a null line.

&\text{WORD} \text{ OF} \ [\text{word1} \ [\text{word2} \ldots \ ]] \ i
Gives the \( i \)th word in the list of words unless the number given is zero or exceeds the number of words in the list.

The format of \textbf{EXEC 2 User-Defined function}, followed by its description, is:

\begin{verbatim}
line-number \text{ OF} \ [\text{arg1} \ [\text{arg2} \ldots\ ]]
label \text{ OF}
\end{verbatim}

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

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

EXECDrop
EXDdrop {exectype} {execname} [options...[]]

options:
[User]
[SYstem]
[SHared]
Do I/O operations between a device and the program stack.

EXECIO {\textit{lines}}
\begin{align*}
\text{DISKR} \ fn \ ft \ [fm \ [linenum]] \ [[\text{FINIs}]\
\quad \text{options} \ [a] \ [b] \ [\)]
\end{align*}
\begin{align*}
\text{CARD} \ [(\text{options} \ [a] \ [b]) \ [)]
\end{align*}
\begin{align*}
\text{CP} \ [(\text{options} \ [a] \ [b] \ [d]) \ [)]
\end{align*}
\begin{align*}
\text{DISKW} \ fn \ ft \ fm \ [linenum] \ [\text{recfm} \ [\text{lreci}]] \ [[\text{FINIs}]\
\quad \text{options} \ [b] \ [c] \ [d] \ [e]) \ [)]
\end{align*}
\begin{align*}
\text{PUNCH} [(\text{options} \ [b] \ [c] \ [d]) \ [)]
\end{align*}
\begin{align*}
\text{PRINT} [(\text{CC code} \ \text{options} \ [b] \ [c] \ [d])
\end{align*}
\begin{align*}
\text{DATA}
\end{align*}
\begin{align*}
\text{EMSG} [(\text{options} \ [b] \ [c] \ [d]) \ [)]
\end{align*}

Option formats:
\begin{itemize}
\item[(a)] \text{FInd /chars/}
\item[(b)] \text{LOcate /chars/}
\item[(c)] \text{Avoid /chars/}
\item[(d)] \text{Zone[\text{n1 n2}]}\quad [\text{STRIP}]
\item[(e)] [\text{N}OTYPE]
\end{itemize}

\text{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 parenthesis is used to mark the end of an EXECIO option, it must be preceded by at least one blank character. A right parenthesis cannot be used to mark the end of the STRING option.
EXECLOAD

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

EXECLOAD [fn ft] [fm [execname [exectype]]] [[options...[]]]

EXLoad

options:

[User]
[SYstem]
[Push]

EXECMAP

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

EXECMap [execname [exectype]] [[options...[]]]

EXMap

options:

[User]
[SYstem]
[SHared]

[STACK FIFO LIFO]
[FIFO]
[LIFO]
EXECOS

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

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

EXECSTAT

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

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

EXECStat

EXStat

EXECUPDT

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

EXECUPDT fn [ft [fm]] [options...[]]
EXPAND

CMS

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

EXPAND fn1 [ft1 [fm1 [fn2 [ft2 [fm2]]]]]] [[options...[]]]

options:

\[
\begin{align*}
\text{INPUT} & \quad \text{filename} \\
\text{CSECT} & \quad \text{csect} \quad \text{SIZE} \quad \text{size} \\
\text{PRINT} & \quad \text{NOPRINT}
\end{align*}
\]\n
Control Statements
EXPAND csect size [, csect size...]

EXTERNAL

CP Class G

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

EXTernal \[
\begin{align*}
\text{code} \quad 40
\end{align*}
\]

F

CMS Border Command

Scrolls the window forward.

F
FETCH

Loads an executable phase into storage for execution.

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

options:

[START]
[COMP]
[ORIGIN hexloc]
Simulates OS JCL (Job Control Language) data definition (DD) statements. Displays current definitions when entered without operands.

FILEDEF Terminal [(optionA optionD[])]

PRinter [(optionA OPTCD J [])]
PUnch [(optionA [])]
Reader [(optionA [])]

DISK [fn ft fm] [(optionA optionB [])]

FILE ddname Al

or

DISK fn ft fm

FILE ddname Al

DSN ?
DSN qual1 qual2...
DSN qual1.qual2...

[option A optionB()]]

or

DISK vaddr
DUMMY [(optionA[])]

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

[(optionA optionC optionE optionF[])]

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

CLEAR
FILEDEF (continued)  

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 nnnn|50 ] [CONCAT]
[LIMCT nnn] [OPTCD a]
[DISP MOD] [MEMBER membername]
[DSORG {PS|PO|DA}]

optionC:

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

optionD:

[UPCASE|LOWCASE]

optionE:

[LEAVE]
[NOEOV]

optionF:

\[
\begin{array}{c}
\text{ALT} \\
\{ \text{TAP}_n \\
\text{c}_u \\
\text{v}_u \\
\} \\
\end{array}
\]

optionG:

\[
\begin{array}{c}
\text{SYSPARM} \\
\{ \text{(string)} \\
\text{(?)} \\
\} \\
\end{array}
\]
FILEDEF

Defines CMS format files and spool files.

FILEDEF

optionA:

[PERM]

[CHANGE]

[NOCHANGE]

[RECFM a]

[LRECL nnnnn]

[BLOCK nnnnn]

[BLKSIZE nnnnn]

optionB:

[DISP MOD]

[DSORG PS]

FILELIST CMS

Lists information about CMS disk files, with the ability to edit and issue commands from the list.

FILELIST [fn [ft [fm]]] [(options...[])]

options:

[Append]

[Filelist | Nofilelist]

[PROFile fn]

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

Closes one or more files.

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

---

**FLUSH**

CP Class D

Halts and immediately purges or holds the current spool file.

```
Flush [raddr] [ALL] [HOId]  
```  

---

**FLUSH**

RSCS

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

**General User Format:**

```
Flush  [*]spoolid  [ ALL ] [HOId ]  
```  

**Operator Format:**

```
Flush [linkid]  {spoolid | *}  [ALL | HOId ] 
```  

---

**FORCE**

CP Class A

Forces logoff of the named user.

```
FORCE  userid 
```
FORCE

FORCE linkid

FORMAT

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 [nocy 1] [options...] 

options:

<table>
<thead>
<tr>
<th>Blksizes</th>
</tr>
</thead>
<tbody>
<tr>
<td>512</td>
</tr>
<tr>
<td>800</td>
</tr>
<tr>
<td>1024</td>
</tr>
<tr>
<td>2048</td>
</tr>
<tr>
<td>4096</td>
</tr>
<tr>
<td>1K</td>
</tr>
<tr>
<td>2K</td>
</tr>
<tr>
<td>4K</td>
</tr>
</tbody>
</table>

[Noerase]
[Label]
[Recomp]
FORMAT-ALLOCATE

FORMAT-ALLOCATE Service Aid

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

Format Service Aid Control Statements

- Format Function
  FORMAT, devadr, devtype, startadr, endadr, volser

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

- Label Function
  FORMAT, devadr, devtype, volser, LABEL

FREE CP Class D

Releases previously held user spool files.

FREE userid [Printer
  PUnch
  ALL]

FREE RSCS

Resumes transmission on a communication link previously in HOLD status.

FREE [linkid]
FWDSPACE

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

FWdspace [linkid] [nnn]

GDUMP

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

GDUMP

\[
\begin{align*}
\text{hexloc1} & \quad \{ - \} \quad \text{hexloc2} \quad \text{TO} \star \quad \text{TO userid} \quad [DSS] \quad \text{FORMAT type} \\
\text{bytecount} & \\
\end{align*}
\]

GENDIRT

Creates auxiliary module directories.

GENDIRT directoryname [targetmode[sourcemode]]

GENIMAGE

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] sfm A1
GENMOD

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

Genmod [fn [MODULE [fm|Al]]] [(options...[])]

options:

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

GENMSG

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

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

options:

[CP] [Dbscs List Xref]
[ Nobcs NOList NOXref]
[Object Margin nn]
[ NOObject Margin 72]

GEN3705

Invokes 3705 Stage 2 service aid.

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

options:

[RUN SAVE]
[NORUN NOSAVE]
GET VSCREEN

GET VSCREEN

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

GET VSCREEN vname fn ft [fm [fromrec [numrec]]]

GLOBAL

GLOBAL

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

GLOBAL {MACLIB [libnamel ... libname63]}

GLOBAL {TXTLIB}

GLOBAL {DOSLIB}

GLOBAL {LOADLIB}

GLOBAL

GLOBAL

defines the CMS load libraries you want searched for modules.

GLOBAL LOADLIB [libnamel ... libname63]
Set, maintain, and retrieve a collection of named variables.

GLOBALV INIT
SELECT {group
  UNNAMED}

GLOBALV

GLOBALV

GLOBALV

GLOBALV

GLOBALV

GLOBALV

GLOBALV

GLOBALV

GLOBALV

GLOBALV

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

GTRACE

Provides additional tracing through VM/GCS.

**GTrace** [ON] [OFF]

---

**H**

CMS Border Command

Hides the window.

**H**

---

**HALT**

CP Class A

Stops any active channel program on the real device specified.

**HALT raddr**

---

**HB**

CMS Immediate Command

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

**HB**
HELP

Displays information about commands, instructions, and messages.

HELP

TASKS

Help

taskname TASKs

menuname MENU

component-name command-name

MESSAGE

MSG

message-id

options:

optionA:

BRIef

DETail

RELated

optionB:

[ALL]

[DESCript]

[FORMAT]

[PARMs]

[OPTIONS]

[NOTEs]

[ERRORs]

optionC:

SCreen

NOType

[EXTend]

HELPCONV

Converts the specified file into a formatted HELP file.

HELPCONV filename filetype [filemode] *

HI

CMS Immediate Command

Halt interpretation command causes all currently executing System Product interpreter or EXEC 2 programs or macros to terminate execution without destroying the environment (as HX would).

HI
HIDE WINDOW

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

HIDE WINDOW (wname [ON vname [line col]])

HO
CMS Immediate Command
Halts the current CMS tracing operation.

HO

HOLD
CP Class D
Defers processing of specified spool output.

HOLD userid [Printer]

Hold PUnch

ALL

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

Hold [linkid] [IMMED]

HT
CMS Immediate Command
Halts displaying at the terminal.

HT
**HX CMS Immediate Command**

Halts execution of the current CMS command or program.

**HX**

**HX**

**GCS**

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

**HX**

**IDENTIFY CMS**

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

Allows changes to the 3800 named systems.

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

options:

[ TERM ]
[ PRINT ]
[ DISK ]

IMMCMOD

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

IMMCMOD { SET } name
{ CLEAR }  
{ QUERY }  
{ STATUS }
INCLUDE

Brings additional TEXT files into virtual storage.

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

options:

CLEAR [RESET {entry}] [ORIGIN {hexloc} [...Trans} [...START]

NOCLEAR

TYPE [MAP] [INV] [REP] [AUTO] [LIBE]

NOTYPE NOMAP NOINV NOREP NOAUTO NOLIBE

DUP [HIST] [SAME] [RLDSAVE]

NODUP NOHIST

INDICATE

Displays the use of and contention for major system resources.

INDicate FAVORed

INDICATE

Displays the use of and contention for major system resources.

INDicate FAVORed

I/O LOAD

PAGing [WAIT] ALL

Queues

USER [* userid]
INDICATE

Displays the use of and contention for major system resources.

**INDicate [LOAD USER]**

INIT

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

**INIT**

IPCSDUMP

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

**IPCSDUMP**

IPL

Simulates an initial program load function for a virtual machine.

**Ipl**

```
    vaddr [cylno] [CLEAR] [STOP] [ATTN] [PARAM p1[p2...]]
```

```
    nnnnnn
    NOCLEAR
```

```
    systemname
```

optional CMS PARMS

- AUTOCR
- BATCH
- INSTSEG [YES|NO|name]
- NOSPROF
- SAVESYS sysname (Class E only)
Invokes other EXECs and commands to do most of the steps in the installation procedure.

**ITASK**

**LOAD**
- CP
- CMS
- GCS
- HELP
- HPO
- IPCS
- TSAF
- CMSSRC
- CPSRC
- GCSSRC
- IPCSSRC
- TSAFSRC
- CMSFTSRC

**LANG**
- ALL
- ALLOBJ
- CP
- CMS
- TSAF
- HELP
- CMSSRC

**BUILD**
- CP [noassem]
- CMS
- GCS [groupname]

**ASSEMBLE**
- ALLOCP
- BMKBOX
- DMKFCB
- DMKRIO
- DMKSNT
- DMKSYS
- DMSNGP

**ALLOCATE**

**BASEIDS**
ITRACE

ITRACE

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

ITrace

{ [GTrace
  [SUP]
  [OFF]
  [ALL]
  [END]
  [GGroup]

L

CMS Border Command

Scrolls the window to the left.

L

LABELDEF

CMS

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

LABELDEF

{ [fn]
  [CLEAR
    [FID(?|fid)] [VOLID (?|valid|SCRATCH)]
    [VOLSEQ volseq] [FSEQ fseq]
    [GENN genn] [GENV genv]
    [CRDTE yyddd] [EXDTE yyddd]
    [SEQ [Q|1|3]]
    [options...{]}

options:

[PERM] [CHANGE
  [NOCCHANGE]
LANGGEN

LANGGEN
Combines all the text files created by LANGMERG for a language and saves them in a DCSS named
NLSxy, where x is the levelid and y is the langid. LANGGEN also saves CP’s message
repository for CP to use.

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

LANGMERG

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

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

LINK

LINK CP Class G
Permits one user to access minidisks belonging to another user.

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

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

LISTDS

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

LISTDS [ dsname ] { fm } [ ( options...[] ) ]

options:

[ FREE ] [ FORMAT ]
[ EXTENT ] [ PDS ]
LISTFILE

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
- XEDIT
- FName
- FType
- FMode
- Format
- Alloc
- Date
- Label
- Blocks
- %x

LISTIO

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

Used to create a CMS LOADLIB or LOADLIB member.

**LKED fn [(options...[])]**

**options:**

- [NCAL] [LET] [ALIGN2] [NE] [OL] [RENT]
- [REUS] [REFR] [OVLY] [XCAL]
- [NAME membername] [LIBE libraryname]

```plaintext
[XREF] [PRINT] [TERM]
[MAP] [DISK] [NOTERM]
[List] [NOPRINT]

[size value1 value2]
  value1
  ,value2
```

LOAD

Brings TEXT files into storage and establishes links.

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

**options:**

- [CLEAR] [RESET] [MAP] [INV] [REP] [START]
- [NCLEAR] [NOMAP] [NOINV] [NOREP]
- [TYPE] [AUTO] [LIBE] [ORIGIN] [DUP] [RLDsave]
- [NOTYPE] [NOAUTO] [NOLIBE] [TRANS] [NODUP]
- [HIST] [NOHIST] [NODUP]
LOADBUF

LOADBUF

CP Class D

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

LOADBUF
{ raddr UCS name [Fold] [Ver] }
{ raddr FCB name [Index [nn]] }
LOADVFCB

CP Class G

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

LOADVFCB vaddr Fcb name [Index [nn]]

LOCATE

CP Class C or E

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

LOCate \{userid \{vaddr\}\}

\{raddr\}

LOCK

CP Class A

Locks specified pages in processor storage.

LOCK \{userid \{firstpage lastpage [MAP]\}\}

\{SYSTEM\}

LOGOFF

CP Class Any

Terminates a terminal session.

LOGoff \[H0ld\]

LOGout

LOGON

CP Class Any

Initiates all virtual machine operation.

Logon \userid \{password\} \[Noipl\]

Login
**M (CMS Border Command)**

**M**

Changes the location of the window.

**MACLIB**

Creates and updates macro libraries.

MAClib

\[
\begin{align*}
\text{GEN} & \{ \text{libname fn}\{\text{fn2...}\} \\
\text{ADD} & \{ \text{libname membername1[membername2...]} \\
\text{REP} & \{ \text{libname} \\
\text{DEL} & \{ \text{libname membername1[membername2...]} \\
\text{COMP} & \{ \text{libname} \\
\text{MAP} & \{ \text{libname [membername1[membername2...]][(options...[])]} \\
\end{align*}
\]

options:

\[
\begin{align*}
\text{DISK} & \\
\text{PRINT} & \\
\text{TERM} & \\
\text{XEDIT} & \\
\text{STACK} & \{ \text{FIFO} \\
\text{LIFO} & \}
\end{align*}
\]

**MACLIST**

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

MACLIST

MList \( \{ \text{libname [(options [])]} \)
MAKEBUF

CREATES A NEW BUFFER WITHIN THE PROGRAM STACK.

MAP

CONVERTS VARIOUS TYPES OF LOAD MAPS INTO THE PROPER FORMAT FOR USE BY IPCS.

MAP type [Prompt]

MAXIMIZE WINDOW

EXPANDS A WINDOW TO THE PHYSICAL SCREEN SIZE.

MESSAGE

SEND TEXT MESSAGES TO OTHER USERS, SYSTEM OPERATOR OR SELF.

MESSAGE

SEND TEXT MESSAGES TO OTHER USERS, SYSTEM OPERATOR OR SELF.
MIGRATE

Activates normal page/swap table migration routines or forces the pages of the specified user to a secondary device when the user is currently active.

MIGrate [userid]

MINIMIZE WINDOW

Reduces the size of the window to one line.

MINimize WINdow [ wname ]

MODMAP

Displays a MODULE file load map.

MODmap fn

MONITOR

Starts or stops the recording of interruptions and other events that occur in the real machine.

MONitor

AUTOdisk { ON, OFF }
CLOSE
Display { SPOOL, TAPE, ALL }
MONITOR (continued)  

```
CP Class A or E

{ENable {PERForm 1
RESPonse
SChedule
USER
INSTsim
DAStap
SEEKs
SYSprof }

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

{INTerval nnnnn [SEC]mm
MIN ]

LIMIT n [NOSTOP
STOP
SAMPLE ]

SEEKS {INclude raddr raddr... }
EXclude raddr raddr...
DElete
DIplay }

STArt [SPOOL [TO userid] [BUFFS n]
CPTRACE

TAPE raddr [MODE 800
1600
6250
38K ][BUFFS n] }
```
MONITOR

MONITOR (continued)

\[
\begin{align*}
\text{STOP} & \quad \text{SPOOL} \\
\phantom{\text{STOP}} & \quad \text{CPTRACE} \\
\phantom{\text{STOP}} & \quad \text{TAPE} \\
\text{TIME} & \quad \{\text{FROM}\ h1:m1\ \text{TO}\ h2:m2\} \\
\phantom{\text{TIME}} & \quad \{\text{FOR}\ \text{hh:mm}\} \\
\phantom{\text{TIME}} & \quad \{\text{ALL} \quad \text{NONE}\}
\end{align*}
\]

\(^2\) See the operand description for the default values.

MOREHELP

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

MOREhelp \(\{(\text{optionA} \ \text{optionB} [)])\}

optionA: \(\{\text{BRIef} \quad \text{DETail} \quad \text{RELated}\}\)

optionB: \(\{\text{ALL} \quad \text{DESCript} \quad \text{FORMat} \quad \text{PARMs} \quad \text{OPTions} \quad \text{NOTEs} \quad \text{ERRors}\}\)

MOVEFILE

Moves data from one device to another device.

MOVEfile \(\text{in_ddname} \quad \{\text{inMOVE} \quad \text{OUIMOVE}\} \quad \{\text{PDS()}\}\)
MSG

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

\[ \text{Msg nodeid \{userid|SYSTEM\} \[msgtext \]} \]

MSGNOH

CP Class A or B

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

\[ \text{MSGNOH \{userid\} \[msgtext \]} \]

N

CMS Border Command

Minimizes the window.

N

NAMEFIND

CMS

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

\[ \text{NAMEFind :tag value \{:tag [value]\}... [options...[]]} \]

options:

\[ \begin{align*}
\text{STACK} & [n|*| 1] [\text{FIFO}|\text{LIFO}] & [\text{File fn}] \\
\text{FIFO} & [n|*| 1] & [\text{LINenum}] \\
\text{LIFO} & [n|*| 1] & [\text{START recnum}] \\
\text{TYPE} & [n|*| 1] & [\text{Size[n|*| 8]}] \\
\text{XEDIT} & & 
\end{align*} \]
NAMES

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

Processes CP spool reader files created by 3705 dumping operations.

NCPDUMP [DUMPxx] [([ERASE] [NOFORM] [NCPBUFF]])]
Controls communications to 370x controllers or resources or 3270 remote equipment.

```
ATTach resid [To] userid [As] cuu

DETach resid [From] userid

DISAble [ALL]
    resid [resid...]  

Display raddr hexloc1
    { : } hexloc2
    END

DUMP raddr [IMMED]
    [OFF]
    [AUTO]

ENable [ALL]
    resid [resid...]  

LOAD raddr ncpname

POLLdlay nnnn [ALL]
    raddr

Query [ACTIVE]
    OFFline
    FREe
    ALL
    resid [resid...]  

SHUTDOWN [raddr]
    [ALL]

VARY {ONline}
    resid [resid...]
    OFFline
```
Controls the 370x control program and its resources. Also provides a means of altering binary synchronous line poll delay interval.

```
NETWORK ATTach resid [To] userid [As] cuu

DETach resid [From] userid

DISAble [ALL resid [resid...]]

{ Display raddr hexloc1 [::hexloc2] END
  [.bytecount] END
}

DUMP raddr [IMMED OFF AUTO]

ENABLE [ALL resid [resid...]]

LOAD raddr ncpname

POLLdelay nnnn [ALL raddr]

Query [ACTIVE OFFline FREE ALL resid [resid...]]

VARY [ONline OFFline resid [resid...]]
```
NETWORK

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

```
NETWORK
  { START [APPLid name]
    [Pass password]
    [RETry nn]
  }
  HALT [QUICK]
```

NOTE

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

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

```
options:
  [ACK] [NOAck]
  [NOTEbook fn] [LOG] [LONG]
  [NOTebook *] [NOLog] [Short]
```

```
[ADD] [Cancel] [Replace] [PROFile fn]
```

NOTREADY

Simulates loss of ready status on virtual device.

```
NOTReady  vaddr
```

NUCXDROP

Deletes specified nucleus extensions.

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

Loads a nucleus extension.

```
NUCXLOAD { name [fn]
    name member ddname } [(SYstem) (SErvice)
    [NEndcmd] [IMmcmd] [Push] ]
```

NUCXMAP

Identifies existing nucleus extensions.

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

O

CMS Border Command

Restores the window.

```
O
```

OPTION

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

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

```
options:
    [DUMP] [NODUMP]
    [DECK] [NODECK]
    [LIST] [NOLIST]
    [LISTX] [NOLISTX]
    [SYM] [NOSYM]
    [XREF] [NXREF]
    [ERRS] [NOERRS]
    [48C] [60C]
    [TERM] [NOTERM]
```
ORDER

ORDER CP Class D

Places closed spool files in a specified order by device type. (A combination of CLASS and spoolid specifications may be entered.)

ORDER [userid] {Reader|RDR} {CLASS c1 CLASS c2...}
SYSTEM {Printer|PRT} {spoolid1 spoolid2...}
Punch {PUnch|PCH} {FORM form1 FORM form2...}
Dest {DEST dest1 DEST dest2...}

Note: Sequencing may be done with the ORDER command using a combination of "CLASS c" FORM, and spoolid specifications.

ORDER CP Class G

Places closed spool files in a specified order by device type.

ORDER {Reader} {CLASS c1 CLASS c2...}
Printer {FORM form1 FORM form2...}
Punch {spoolid1 spoolid2...}
Dest {DEST dest1 DEST dest2...}

ORDER RSCS

Reorders files enqueued on a specific link.

ORDER [linkid] spoolid1 [spoolid2 ...]

ORDER CMS

OSRUN

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

OSRUN member [PARM=parameters]

Chapter 4. CP, CMS, GCS, TSAF, RSCS, and IPCS Commands 159
OSRUN

OSRUN
OSRUN GCS
Starts a GCS application program.

OSRUN member [PARM=parameters]

OVERRIDE

OVERRIDE
OVERRIDE fn [ft OVERRIDE] [fm] [EDIT]

OVERRIDE CMS
Implements changes to the class structure.

Control Statements:

DESTination cuu devtype volser altcuu

command [Type=c] Class = {classes}

P

P CMS Border Command

Pops the window.

P

PARSECMD

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

PARSECMD CMS
Use PARSECMD to invoke the parsing facility from an exec.

options:

[TYPE [APPLID applid] [STRING cmdstring]]

[NOTYPE]
PEEK

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

PEEK  [spoolid] [(options...[])]

options:

[FRom recno]
[FOR numrec]
[PROFile fn]

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

Monitors certain events as they occur during program execution in the user's virtual machine, such as: the fetching and execution of an instruction, the execution of a successful branch instruction, the execution of an instruction that alters a specific general purpose register, and the execution of an instruction in the virtual machine that alters storage.

event types:

PER

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


PER (continued)

options:

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

PAss \[ \frac{0}{n} \]

CMd \{text\}

\[
\begin{bmatrix}
\text{Printer} & [\text{RUN}] \\
\text{TErminal} & \text{NORun} \\
\text{BOTH} & \text{RUN} \\
& \text{STEP}\left(\frac{1}{n}\right)
\end{bmatrix}
\]

GUESTR
GUESTV
DATOFF
DAT

sub-commands:

COUNT
TABLE
SAVE traceset-name [APPend]
GET traceset-name [APPend]
END \{ ALL \\
COUNT \\
CURRENT \\
element-number \\
event-type \\
traceset name \}
POP WINDOW

Moves a window up in the order of displayed windows.

POP WINdow  \{ wname \} \[ \] * 

POSITION WINDOW

Changes the location of a window on the physical screen.

POSition WINdow  \{ wname \} psline pscol 

PRB

Updates the STATUS, FUNCTN, SEV, or DUP/APAR/PTF field in a symptom record for a specific problem number or displays a specific problem report.

PRB  \{ \begin{align*} 
\text{APAR} & \text{ aparnumber} \\
\text{CLOSE} & \\
\text{DSPLY} & \\
\text{DUPOF} & \{ \text{nnnnn} | \text{aparnumber} \} \\
\text{IBM} & \\
\text{NEEDINFO} & \\
\text{PTFIS} & \{ \text{filename} \} \text{ filetype} \\
\text{PTFON} & \\
\text{SEV} & \{ 1 \ 2 \ 3 \ 4 \} \\
\text{USER} & \\
\text{HELP} & \\
\end{align*} \} 

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

PRELOAD

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

PRELOAD loadlist [control]

PRINT

Directs a specified spool file to the virtual printer.

PRINT fn ft [fm] [[options...]]

options:

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

PROB

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

PROB
Anyone, authorized by the active routing table, may execute the programmable operator commands. To execute a programmable operator command you must send a message to the programmable operator facility virtual machine. The text of the message is the command to be issued. Use the CMS EXEC, PROPST EXEC, to invoke the programmable operator facility.

The format of the invocation EXEC is:

\[
\text{PROPST} \left[ \text{rtable-name} \right] \text{PROP} \left[ \text{DISConn} \right]
\]

The local format of the message sent to the programmable operator facility is:

\[
\text{Message} \quad \text{userid} \quad \text{propcmd} \quad \text{[parameters]}\]

\[
\text{MSG}
\]

The distributed (network) format of the message sent to the programmable operator facility is:

\[
\text{SMsg} \quad \text{netid} \quad \text{Msg} \quad \text{nodeid} \quad \text{userid} \quad \text{propcmd} \quad \text{[parameters]}
\]

The CMS TELL EXEC may be used by the logical operator instead of either the local or the distributed format.

The format of the TELL EXEC is:

\[
\text{TELL} \left\{ \text{nickname} \right\} \quad \text{message} \quad \text{userid} \quad \text{[AT node]}
\]

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

CMD \quad \text{vmcmd}

Executes selected CP or CMS commands in the programmable operator’s virtual machine.
PROGRAMMABLE OPERATOR

PROGRAMMABLE OPERATOR (continued) CMS

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

GET { FEEDBACK
     FB
   } LOG [ymmd]
Retrieves one of the programmable operator files: the feedback file (FB) or the log file (LOG).

LOADTBL [filename] [(RPL[])]
Loads a new routing table to control the operation of the programmable operator facility.

LGLOPR { ASN
           RLS
           RPL
       }
Changes the assignment of logical operator of the programmable operator facility.

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

QUERY HOSTCHK
Indicates node-checking status.

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

QUERY LOGGING
Indicates logging status.

QUERY PROPCHECK
Indicates node-checking status.

QUERY RTABLE
Indicates the name of the programmable operator’s active routing table.
PROGRAMMABLE OPERATOR

PROGRAMMABLE OPERATOR (continued)

SET DEBUG {ON OFF}
Enter and exit from programmable operator DEBUG mode.

SET HOSTCHK {ON OFF}
Restarts or halts checking of the host system by the distributed system.

SET LOGGING {ON OFF ALL}
Causes the programmable operator facility to stop writing any messages to the log file.

SET PROPCHK {ON OFF [nodeid]}
Restarts or halts checking of the programmable operators on the distributed systems.

STOP
Stops operation of the programmable operator.

Note: The SET DEBUG command may be entered only at the programmable operator virtual console. The SET LOGGING, SET HOSTCHK, and SET PROPCHK commands may be entered at the programmable operator virtual console, and also from the logical operator’s console.

PRTDUMP

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

PRTDUMP PRBnnnnn [option...]

CP DUMP options:

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

CMS DUMP options:

not available; standard print routine is used.

partial DUMP options:

none; standard print routine is used.
PSERV

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

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

options:

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

PUNCH

Directs a specified spool file to the virtual punch.

PUnch fn ft [fm!*] [(options...())]

options:

[HEADER] [NOHEADER] [MEMBER {* membername}]

PURGE

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

PURge [FORCE] [userid] [SYSTEM] [READER|RDR] [PRINTER|PRT] [PUNCH|PCH] [ALL CLass c1 CLassc2 ...1 spoolid1 spoolid2 ... FORM form1 FORM form2 ... DEST dest1 DEST dest2 ...]

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

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

```
PURge Reader  Class c1 Class c2 ...
   Printer  FORM form1 FORM form2 ...
   PUNch   spoolid1 spoolid2 ...
   ALL     ALL
   DEST    DEST dest1 DEST dest2...
```

PURGE

Removes and discards all or specified inactive files from a link.

**General User Format:**

```
PURge [*]spoolid
```

**Operator Format:**

```
PURge [linkid]  [spoolid [spoolid...] | ALL]
```

PUT SCREEN

Makes a copy of the physical screen and writes the image to a CMS file.

```
PUT SCREEN  fn ft \[ fm \]
```

PUT VSCREEN

Writes the data from the scrollable data area of a virtual screen to a CMS file.

```
PUT VSCREEN  vname fn ft \[ fm \  \[ fromlin \ numlin \] \]
```

Chapter 4. CP, CMS, GCS, TSAF, RSCS, and IPCS Commands 169
In CMS or CMS/DOS mode, provides information about the virtual machine environment.

Query

```
| ABBREV |
| APL    | wname [ALL]|
| AUTOREAD |
| BLIP   |
| BORDER | mode |
| CHARMODE |
| CMSLEVEL |
| CMSPF  |
| CMSLEVEL |
| CMSTYPE |
| CURSOR | wname |
| DISK   |
| DISK   |
| Disk   |
| Display |
| DLBL   |
| DOS    |
| DOSLIB |
| DOSLNCNT |
| DOSPART |
| EXECTRAC |
| FILEDEF |
| FULLREAD |
| FULLSCREEN |
| HIDE   |
| IMESCAPE |
| IMPCP  |
| IMPEX  |
| INPUT  |
| INSTSEG |
| KEY    |
| LABELDEF |
| LANGLIST |
| LANGUAGE |
| LDRTBLS |
```

[(options ... [])]
QUERY (continued)

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

Chapter 4. CP, CMS, GCS, TSAF, RSCS, and IPCS Commands 171
QUERY

Displays system software maintenance information, log messages, the number of logged-on users; lists logged-on users. Provides the paging activity index or specified user priority or status of the Virtual Machine Assist feature.

Query

```
AFFinity [userid]
CPAssist1
JOURNAL2
LOGmsg
Names
PAGING
PRIORity userid
PROCESSr
QDROP
SASSist1
SPMODE
SRM
  APAGES
  DSPSlice
  IB
  MAXDrum
  MAXWss
  MHFULL
  PB
  PCI
  PGMStat
  PGMTlim
users [userid]
```

1 The collective use of both QUERY CPASSIST and QUERY SASSIST is used to determine the current status of the expanded virtual machine assist portion of the Extended Control-Program Support: VM/370.

2 The JOURNAL operand is valid only if STQUERY = YES is specified in the SYSJRL macro instruction in DMKSYS.
CP Class B

Displays system status, paging, scheduling, machine configuration information, system software maintenance information, log messages, the number of logged-on users; lists logged-on users.

Query

```
    [ DA.sd [ Sysvirt ] ]
    [ GRAf ]
    [ LINES ]
    [ UR ]
    [ TApes ]
    [ ALL ]

    DAsd valid
    DUMP
    Lnnn
    LOGmsg
    MITime
    Names
    PROCessr
    raddr [ -raddr2 ]
    lpert
    STATUS raddr
    STORage
    SYStem raddr
    TDsk
    Users [ userid ]
```

CP Class C

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

Query

```
    [ CPTrap [ STATUS ] ]
    [ SELECT ALL typenum1 [ ... typenumn ] ]

    LOGmsg
    Names
    PROCessr
    Users [ userid ]
```
QUERY

QUERY

CP Class D

Provides data on spooling operations.

Query

\[
\begin{align*}
\text{Files } & [\text{Class c}] [\text{FORM form}] [\text{DEST dest}] [\text{userid}]^2 \{\text{Hold NOHold SYShold USERhold}\}^1 \\
\text{Hold} & \\
\text{Printer} & [\text{Class c}] [\text{FORM form}] [\text{DEST dest}] [\text{userid}]^2 \{\text{Hold NOHold SYShold USERhold}\}^1 \\
\text{PUnch} & \\
\text{Reader} & \\
\text{spoolid} \\
\text{CPELEVEL} & \\
\text{LOGmsg} & \\
\text{Names} & \\
\{\text{ACTIVE ATTach}\} & \\
\text{UR} & \{\text{FREE OFFline}\} [\text{PATHS}] \\
\text{Users} & [\text{userid}] \\
\end{align*}
\]

1 The options may be specified in any order.

2 Using a 1-4 digit all numeric userid will cause unpredictable results for the QUERY command, which also has a 1-4 digit all numeric spoolid parameter.
QUERY

CP Class E

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

Query

\[
\begin{align*}
\text{AFFinity [userid]} \\
\text{CPAssist} \_1 \\
\text{Journal2} \\
\text{LOGmsg} \\
\text{Names} \\
\text{PAGing} \\
\text{PRIORity userid} \\
\text{PROCessr} \\
\text{QDROP} \\
\text{SASSist} \_1 \\
\text{SRM} \quad \text{APAGES} \\
\text{\quad\quad DSPSlice} \\
\text{\quad\quad IB} \\
\text{\quad\quad MAXDrum} \\
\text{\quad\quad MAXWss} \\
\text{\quad\quad MHFULL} \\
\text{\quad PB} \\
\text{\quad PCI} \\
\text{\quad PGMStat} \\
\text{\quad PGMTlim} \\
\text{Users [userid]} 
\end{align*}
\]

1 The collective use of both QUERY CPASSIST and QUERY SASSIST is used to determine the current status of the expanded virtual machine assist portion of the Extended Control-Program Support: VM/370.

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

QUERY

CP Class F

Displays log messages, number of logged-on users; lists logged-on users.

Query

\[
\begin{align*}
\text{LOGmsg} \\
\text{Names} \\
\text{Users [userid]} 
\end{align*}
\]
QUERY

Provides system status and machine configuration information.

Query

<table>
<thead>
<tr>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Set</td>
</tr>
<tr>
<td>TERMINal</td>
</tr>
</tbody>
</table>

Files [CLass c] [FORM form] [NOHold Hold USERHold SYShold]

LOGmsg

CHANnels
GRAF
CONsole
DAsd
TApes
LINES
UR
STORage
ALL
[vaddr[-vaddr]]

Links vaddr
CPLANG
CPELEVEL
CPUid
SECuser
Names
S370E

Reader [CLass c ] [NOHold Hold USERHold SYShold] [ALL]

Printer [FORM form] [NOHold Hold USERHold PSF]

PUnch [DEST dest] spoolid [SYShold]

PF [nn] 
SCReen
PROCessr
VMSAVE
SPMODE
QUERY (continued)

REQUESTS INFORMATION ABOUT YOUR GCS VIRTUAL MACHINE.

The DISK, DLBL, LOADLIB, FILEDEF, SEARCH, and SYSNAMES operands work the same as for the CMS QUERY command with the exception that no options are allowed.
Requests information about the TSAF configuration when the TSAF virtual machine is running. Only the TSAF virtual console or the secondary user of the TSAF virtual machine can issue this command.

QUERY

\[
\begin{align*}
\text{COLLECT} & \quad \text{ETRACE} \\
\text{LINK} & \quad \left\{ vdev \right\} \\
\text{RESOURCE} & 
\end{align*}
\]

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

Query nodeid

\[
\begin{align*}
\text{linkid} & \quad \text{Active} \\
& \quad \text{Def} \\
& \quad \text{Files} \\
& \quad \text{Queue} \\
& \quad \text{Sum} \\
\text{File} & \quad \{ \text{spoolid} \} \\
& \quad \{ \text{*spoolid} \} \\
& \quad \{ \text{*} \} \\
\text{System} & \quad \text{NETwork} \\
& \quad \text{Active} \\
& \quad \text{Dest} \\
& \quad \text{EXits} \\
& \quad \text{LEVEL} \\
& \quad \text{LOCAL} \\
& \quad \text{Links} \\
& \quad \text{Ports} \\
& \quad \text{Queue} \\
& \quad \text{Routes} \\
& \quad \text{REroutes} \\
\text{Status} & \quad \text{Rcs} \\
& \quad \text{S} \\
& \quad \text{Vm} \\
\end{align*}
\]
QVM

Requests the transition from VM/370 to a particular virtual machine, running in native mode.

QVM userid [NORETURN]

---

R

CMS Border Command

Scrolls the window to the right.

R

---

RDR

CMS

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

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

options:

[NOTYPE
STACK [FIFO]
FIFO
LIFO]

---

RDRList

CMS

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

RDRlist [ (options...[])]

RList

options:

[PROFILE fn]
[Append ]
READCARD

READCARD

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

\[
\text{READcard} \quad \left\{ \begin{array}{c}
\text{fn} \\
\text{ft} \\
\text{[fm]} \\
\text{[A]} \\
\end{array} \right\} \quad \text{[}(\text{options}...\text{[)])]}
\]

\[
\begin{array}{c}
\text{options:} \\
\text{[Fullprompt]} \\
\text{[Replace]} \\
\text{[Minprompt]} \\
\text{[NORedate]} \\
\text{[NOPrompt]} \\
\text{[NOReplace]} \\
\end{array}
\]

READY

CP Class G

Makes a device-end interruption pending for the specified device.

READY vaddr

READY

RSCS

Notifies RSCS that a forms mount has been satisfied, or that a setup page is wanted. This command is for RJE, 3270P, SNA3270P, and MRJE type links.

Ready [linkid]

RECEIVE

CMS

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

RECEIVE \quad [\text{spoolid} \ [\text{fn} \ [\text{ft} \ [\text{fm}]])]] \quad \text{[}(\text{options}...\text{[)])]}

\text{options:}

\[
\begin{array}{c}
\text{[NOTEbook fn]} \\
\text{[Log]} \\
\text{[Olddate]} \\
\text{[Fullprompt]} \\
\text{[Replace]} \\
\text{[NOLog]} \\
\text{[NEwdate]} \\
\text{[Minprompt]} \\
\text{[NORedate]} \\
\text{[NOPrompt]} \\
\text{[NOReplace]} \\
\text{[Purge]} \\
\text{[Stack]} \\
\end{array}
\]

180 VM/SP Quick Reference
RECONN

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

REFRESH

Updates virtual screen and their associated windows.

RELEASE

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

RELEASE { cuu } { (DET [ ])}

RELEASE

Releases a disk.

RELEASE { cuu } { (DET[])}

{ mode }
RENAME

Changes the name of one or more CMS disk files.

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

options:

[TYPEno] [NOUPDIRT]

REPEAT

Holds or increases the copies of an output spool file.

REPeat [raddr] [nnn] [l] [Hold]

REPLY

Replies to messages sent to the GCS operator.

REPLY id [text]

REQUEST

Makes an attention interruption pending.

REQUEST
REROUTE

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

\[
\text{REROUTE} \\
\{ \text{Files} \} \quad \{ \text{Msgs} \} \quad \{ \text{nodeid} \} \quad \{ \text{userid} \} \\
\{ \text{ALL} \} \quad \{ \text{FOR} \} \quad \{ \text{*} \} \quad \{ \text{SYSTEM} \} \quad \{ \text{ANY} \} \\
\} \\
\{ \text{NOTrcvg} \} \quad \{ \text{FOR} \} \quad \{ \text{userid} \} \quad \{ \text{ANY} \} \\
\{ \text{OFF} \} \\
\]

RESERVE

Allocates all available blocks of a 512-, 1K-, 2K-, or 4K-byte block formatted minidisk to a unique CMS file.

\[
\text{RESERVE} \quad \{ \text{fn ft fm} \} \\
\]

RESET

Clears all pending interruptions; resets error conditions on the device specified.

\[
\text{RESET} \quad \text{vaddr} \\
\]

RESTORE WINDOW

Returns a maximized or minimized window to its size and location prior to the maximize or minimize.

\[
\text{RESTORE WINDOW} \quad \{ \text{wname} \} \\
\]

Chapter 4. CP, CMS, GCS, TSAF, RSCS, and IPCS Commands 183
REWIND

Rewinds a real tape drive.

REWind vaddr

REXX

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

REXX INSTRUCTIONS

The formats of the REXX instructions, followed by their descriptions, are:

ADDRESS
environment [expression] ;
[VALUE] expression

Effects a temporary or permanent change to the destination of command(s).

ARG [template];
Retrieves the argument strings provided to a program or internal routine and assigns them to variables. It is just a short form of the instruction PARSE UPPER ARG [template];.

CALL name [expression][,[expression]]...;
Invokes an internal routine, an external routine or program, or a built-in function. The invoked routine may optionally return a result upon its completion.
DO [name=expri [TO exprt] [BY exprb] [FOR exprf]]; 
FOREVER 
exrrr 
| instruction |
| " |
| " |
| " |
END [symbol];
Or, to present the instruction more generally:

DO [repetitor] [conditional];

| instruction |
| " |
| " |
| " |
END [symbol];
Groups instructions together and optionally executes them repetitively.

DROP name [name] [name]...;
"Unassigns" variables; that is, restores them to their original uninitialized state.

EXIT [expression];
Unconditionally leaves a program, and optionally returns a data string to the caller. The program is terminated immediately.
REXX (continued)

IF expression [;] THEN[;] instruction
    [ELSE[;] instruction]
Conditionally executes an instruction or group of instructions.

INTERPRET expression;
Executes instructions that have been built dynamically by evaluating an expression (rather than that exist permanently in the program).

ITERATE [name];
Alters the flow within a repetitive DO loop (that is, any DO construct other than that with a plain DO).

LEAVE [name];
Causes immediate exit from one or more repetitive DO loops (that is, any DO construct other than that with a plain DO).

NOP;
NOP is a dummy instruction that has no effect. It can be useful as the target of a THEN or ELSE clause.

NUMERIC (DIGITS [expression] ;
    { FORM [SCIENTIFIC ]
        [ENGINEERING ]
    } FUZZ [expression]
Changes the way in which arithmetic operations are carried out.
NUMERIC DIGITS controls the precision to which arithmetic operations will be carried out.
NUMERIC FORM sets the form of exponential notation to be used.
NUMERIC FUZZ controls how many digits, at full precision, will be ignored during a comparison operation.

OPTIONS [expression]
Specifies whether double byte character set (DBCS) strings can be manipulated.

    OPTIONS ETMODE DBCS strings can be manipulated
    OPTIONS NOETMODE DBCS strings cannot be manipulated
REXX (continued)

PARSE [UPPER] [template];

ARGS
EXTERNAL
NUMERIC
PULL
SOURCE
VALUE [expression] WITH
VAR name

Assigns data (from various sources) to one or more variables according to the rules of parsing.

PROCEDURE [EXPOSE name [name] [name]...];

Used within an internal routine (subroutine or function) to protect all the existing variables by making them unknown to following instructions.

PULL [template];

Reads a string from the program stack (system-provided data queue). It is just a short form of the instruction: PARSE UPPER PULL [template];

PUSH [expression];

The string resulting from expression will be stacked LIFO -- onto the most recently created buffer of the program stack (system-provided data queue), limited to 255 characters per entry. If no expression is specified, a null string is stacked.

QUEUE [expression];

The string resulting from expression will be appended to the most recently created buffer of the program stack (system-provided data queue) limited to 255 characters per entry. That is, it will be stacked FIFO. If no expression is specified, a null string is stacked.

RETURN [expression];

Returns control (and possibly a result from a REXX program or internal routine to the point of its invocation.

SAY [expression];

The result of evaluating the expression is displayed (or typed, etc.) on the user's console. The result of the expression may be of any length.
REXX (continued)

SELECT;
  WHEN expression [;] THEN [;] instruction
  [WHEN expression [;] THEN [;] instruction
  ["" "" "$" "$"
  ["" "" "$" "$"
  [OTHERWISE [;] [instruction]]
  [""
  ["
END;
  Conditionally executes one of several alternative instructions.

SIGNAL (labelname [VALUE] expression
  \{ON \{ERROR \{HALT \{OFF \{NOVALUE \{SYNTAX \}

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

```
REXX

TRACE

? [?...]
! [!...]

All
Commands
Errors
Intermediates
Labels
Normal
Negative
Off
Results
Scan

[number]

Or, alternatively:

TRACE

[string

[VALUE] expression

; symbol

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

UPPER variable [variable] [variable]...

Used to translate the contents of one or more variables to uppercase. The variables are translated in sequence from left to right.
```
REXX BUILT-IN FUNCTIONS

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

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

ABBREV(information, info[, length])
Tests whether info is a true abbreviation of information, with minimum length.

ABS(number)
Returns the absolute value of number.

ADDRESS()
Returns the current environment for commands.

ARG([n[, option]])
Returns the number of arguments, the nth argument, or test if the nth argument exists or is omitted.

BITAND(string1[, string2][, pad])
Returns a string composed of the two input strings logically AND’ed together, bit by bit.

BITOR(string1[, string2][, pad])
Returns a string composed of the two input strings logically OR’ed together bit by bit.

BITXOR(string1[, string2][, pad])
Returns a string composed of the two input strings logically exclusive OR’ed together, bit by bit.

CENTER(string, length[, pad])

CENTRE(string, length[, pad])
Returns a string of length length with string centered in it, with pad characters added as necessary to make up length.

COMPARE(string1, string2[, pad])
Returns 0 if the strings are identical. If they are not, the returned number is non-zero and is the position of the first character that does not match.

COPIES(string, n)
Returns n concatenated copies of string.
REXX (continued)

C2D(string[,n])
Character to Decimal. Returns the decimal value of the binary representation of string.

C2X(string)
Character to Hexadecimal. Converts a character string to its hexadecimal representation (unpacks).

DATATYPE(string[,type])
If only string is specified, the returned result is NUM if string is a valid REXX number (any format), or CHAR otherwise. If type is specified, returns 1, string matches the type, otherwise 0.

DATE([option])
Returns the local date in the format: dd Mmm yyyy or in the format according to option.

DELSTR(string,n[,length])
Deletes the substring of string that begins at the nth character, and is of length length.

DELWORD(string,n[,length])
Deletes the substring of string that starts at the nth word, and is of length length blank-delimited words.

D2C(whole-number[,n])
Decimal to Character. Returns a character string of length as needed, or of length n, which is the binary representation of the decimal number.

D2X(whole-number[,n])
Decimal to Hexadecimal. Returns a string of hexadecimal characters of length as needed or of length n, which is the hexadecimal (unpacked) representation of the decimal number.

ERRORTEXT(n)
Returns the error message associated with error number n.

EXTERNALS()
Returns the number of lines in the terminal input buffer (system external event queue).

FIND(string,phrase)
Searches string for the first occurrence of the sequence of blank-delimited words phrase, and returns the word number of the first word of phrase in string.

FORMAT(number[,before][,[after]])
Rounds and formats number to specified integer (before) and (after) decimal places.
REXX

REXX (continued)

INDEX(haystack,needle[,start])
    Returns the character position of one string, needle, in another, haystack, beginning at start.

INSERT(new,target[,n][,length][,pad]])
    Inserts the string new, padded to length length, into the string target after the nth character.

JUSTIFY(string,length[,pad])
    Formats blank-delimited words in string, by adding pad characters between words to justify to both margins.

LASTPOS(needle,haystack[,start])
    Returns the position of the last occurrence of one string, needle, in another, haystack, beginning at start.

LEFT(string,length[,pad])
    Returns a string of length length containing the left-most length characters of string.

LENGTH(string)
    Returns the length of string.

LINESIZE()
    Returns the current terminal line width (the point at which the interpreter will break lines displayed using the SAY instruction).

MAX(number[,number]...)
    Returns the largest number out of the list specified.

MIN(number[,number]...)
    Returns the smallest number out of the list specified.

OVERLAY(new,target[,n][,length][,pad]])
    Overlays the string new, padded or truncated to length length, onto the string target starting at the nth character.

POS(needle,haystack[,start])
    Returns the position of one string, needle, in another, haystack, beginning at start.

QUEUED()
    Returns the number of lines in the program stack (system-provided data queue).

RANDOM([min][,[max][,seed]])
    Returns a pseudo-random non-negative whole number in the range of 0-999 or min to max inclusive. The generator seed may be specified.

REVERSE(string)
    Returns string, swapped end for end.
REXX (continued)

RIGHT(string, length[, pad])
Returns a string of length length right justified.

SIGN(number)
Number is rounded and returns the sign of number (-1, 0, or 1).

SOURCELIN([n])
Returns the line number of the final line in the source file or the nth line.

SPACE(string[, n][, pad])
Formats the blank-delimited words in string with n pad characters between each word. If it is 0, all blanks are removed.

STRIP(string[, option][, char])
Removes Leading, Trailing, or Both leading and trailing characters from string when the first character of option is L, T, or B respectively. The default is B.

SUBSTR(string, n[, length][, pad])
Returns the substring of string that begins at the nth character, and is of length length.

SUBWORD(string, n[, length])
Returns the substring of string that starts at the nth word, and is of length length blank-delimited words.

SYMBOL(name)
If name is not a valid REXX symbol, BAD is returned. If it is the name of a variable, VAR is returned. Otherwise LIT is returned.

TIME([option])
Returns the local time in the 24-hour clock format: hh:mm:ss (hours, minutes, and seconds). All calls in one expression are synchronized.

TRACE([option])
Returns current trace setting, and sets new trace option.

TRANSLATE(string[, tableo][, tablei][, pad])
Translates characters in string to be other characters, or may be used to reorder characters in a string. If neither translate table is given, string is simply translated to uppercase.

TRUNC(number[, n])
Returns the integer part of the number, and n decimal places. The default n is zero.

USERID()
Returns the system-defined User Identifier.

VALUE(name)
Returns the value of the symbol name.
REXX (continued)

VERIFY(string,reference[,['Match'][,start]])
   Verifies that the string is composed only of characters from reference, by returning the position of the first character in string that is not also in reference. If all the characters were found in reference, 0 is returned.

WORD(string,n)
   Returns the nth blank-delimited word in string.

WORDINDEX(string,n)
   Returns the position of the nth blank-delimited word in string.

WORDLENGTH(string,n)
   Returns the length of the nth blank-delimited word in string.

WORDS(string)
   Returns the number of blank-delimited words in string.

XRANGE([start][,end])
   Returns a string of all one-byte codes between and including the values start and end.

X2C(hex-string)
   Converts hex-string (a string of hexadecimal characters) to Character (packs).

X2D(hex-string[,n])
   Converts hex-string (a string of hexadecimal characters) to decimal.
RXSYSFN PACKAGE OF CP/CMS FUNCTIONS

These all provide useful CP or CMS functions. The package is loaded automatically when needed. The formats are followed by their descriptions.

CMSFLAG(flag)
Returns the setting of one of the specified CMS flags:

<table>
<thead>
<tr>
<th>ABBREV</th>
<th>AUTOREAD</th>
<th>CMSTYPE</th>
<th>DOS</th>
<th>EXECTRAC</th>
</tr>
</thead>
<tbody>
<tr>
<td>IMCP</td>
<td>IMPEX</td>
<td>PROTECT</td>
<td>RELPAGE</td>
<td>SUBSET</td>
</tr>
</tbody>
</table>

DIAG(n[?] [,data][,data]...)
Communicates with CP via a dummy DIAGNOSE instruction and returns data as a character string.

DIAGRC(n[?] [,data][,data...])
Is identical to the DIAG function, except that CP return code and condition code are prefixed to the result.

STORAGE([address[,length][,data]])
Returns the current virtual machine size if no arguments are specified; else returns length bytes from user’s memory starting at address...

RO

CMS Immediate Command

Resumes recording of trace information previously suspended by the SO Immediate command.

RO

ROUTE

Directs data of a particular message class to a virtual screen.

ROUTE msgclass TO vname [[options...[]]]

OPTIONS:

[ALARM] [NOTify]
[NOALARM] [NONotify]
ROUTE

ROUTE  
Temporarily adds, deletes, or alters an RSCS routing table entry. (For RSCS operator only)
ROUte nodeid   [OFF|TO linkid]

RSERV  
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  
Resumes terminal displaying.
RT

RUN  
Initiates a series of functions for a file depending on the file type. Selects or combines the procedures to compile, load, or start execution of the specified file.
RUN   fn [ft [fm]] [(args...())]
RUNTSAF

Starts the TSAF virtual machine. Only the TSAF virtual console or the secondary user of the TSAF virtual machine can issue this command.

RUNTSAF [nnn] [ETRACE]

S

CMS Border Command

Changes the size of the window.

S

SAVENCP

Reads/Loads 3705 control program load module.

Note: CP command privilege class A, B, or C is required to use SAVENCMP.

SAVENCP fn [(options...[)])

options:

ENTRY symbol [NAME ncpname] [LIBE libname] [CAMOD {0}]

SAVESYS

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

SAVESYS systemname
SCREEN

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

```plaintext
area [{ extcolor } { ext hil ight }] ]
```

Note: Each time you enter the command, you must specify at least one screen “area” operand with at least one “extcolor” and/or “exthilight” value. You may specify more than one ‘area’ operand on the same command line.
Moves a window to a new location on the virtual screen to which it is connected.

<table>
<thead>
<tr>
<th>SCROLL</th>
<th>B. Ackward [ wname = [ n ] ]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Bottom [ wname = ]</td>
</tr>
<tr>
<td></td>
<td>Down [ wname = [ n ] ]</td>
</tr>
<tr>
<td></td>
<td>Forward [ wname = [ n ] ]</td>
</tr>
<tr>
<td></td>
<td>Left [ wname = [ n ] ]</td>
</tr>
<tr>
<td></td>
<td>Next [ wname = [ n ] ]</td>
</tr>
<tr>
<td></td>
<td>Right [ wname = [ n ] ]</td>
</tr>
<tr>
<td></td>
<td>Top [ wname = ]</td>
</tr>
<tr>
<td></td>
<td>Up [ wname = [ n ] ]</td>
</tr>
</tbody>
</table>

SCROLL CMS

SEND CP

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

SEND [CP] { userid } [text] [lprt]
**SendFile**

Send files or notes to one or more computer users, attached locally or remotely, by issuing the command or by using a menu (display terminal only).

SendFile [fn ft[fm][TO] name ... ][options...[ ]]}

Options:

- [Ack] [Filelist] [Log] [New] [Type] [NOTE]
- [NOAck] [Nofilelist] [Nolog] [Old] [Notype]

**Sentries**

Determines the number of lines currently in the program stack.

Sentries

**Set**

Controls various functions within your virtual machine. (Only one function may be specified per Set command.)

The options available with SET are listed below. A complete description of each option follows.

- ABBREV
- APL
- AUTOREAD
- BLIP
- BORDER
- CHARMODE
- CMSPF
- CMSTYPE
- DOS
- Doslncnt
- Dospart
- Exetrac
- Fullread
- Fullscreen
- Imescape
- Impcp
- Impex
- Input
- Instseg
- Language
- LDRTBS
- Linend
- Location
- Logfile
- Nondisp
- Nonshare
- Output
- Protect
- Rdymsg
- REDTYPE
- RELPAGE
- Remote
- Reserved
- Sysname
- Text
- Translate
- Upsi
- Vscreen
- Window
- Wmpf
SET ABBREV

Controls whether the system ignores user abbreviations of system commands and EXECs or accepts only the full system command name or the full user synonym (if one is available) for system commands.

```
SET ABBREV {ON OFF}
```

SET APL

Activates character code conversion to APL characters for windows.

```
SET APL {ON OFF}
```

SET AUTOREAD

Specifies whether console read is to be issued immediately after command execution or pressing the ENTER key is its equivalent.

```
SET AUTOREAD {ON OFF}
```

SET BLIP

Turns ON or OFF the BLIP character string displayed at the terminal to indicate every two seconds of virtual interval timer time.

```
SET BLIP string [(count)]
  {ON OFF}
```
SET BORDER

Defines borders around windows.

```
SET BORDER wname {ON} [{optionA optionB[]}]
```

optionA: [TOP char] [BOTTOM char] [LEFT char] [RIGHT char] [ALL char]

optionB: [High]

[NOHigh]

[color]
[exthi]
[PSset]

SET CHARMODE

Specifies whether character attributes should be used when displaying virtual screen data on the physical screen.

```
SET CHARMODE {ON}
```

SET CMSPF

Defines a command that should be executed when a specified PF key is pressed in CMS full-screen mode.

```
SET CMSPF nn {pseudonym} {keyword} string
```

[NOWRITE] [DELAYED]
SET CMSTYPE

Specifies suppression of a CMS terminal display within an EXEC.

\[ \text{SET CMSTYPE } \{ \text{HT} \}_{\text{RT}} \]

SET DOS

Indicates whether your CMS virtual machine is in CMS/DOS environment, specifies the mode letter at which the VSE system residence is accessed, and specifies that you are going to use the AMSERV command or you are going to execute programs to access VSAM data sets.

\[ \text{SET DOS } \{ \text{ON} \}_{\text{OFF}} \{ \text{mode} \{(\text{VSAM})\} \} \]

SET DOSLNCNT

Specifies the number of SYSLST lines per page.

\[ \text{SET DOSLNCNT } \text{nn} \]

SET DOSPART

Specifies control regarding the size of the virtual partition in which you want a program to execute.

\[ \text{SET DOSPART } \{ \text{nnnnnK} \}_{\text{OFF}} \}

SET EXECTRAC

Specifies whether you want tracing turned on or off for your System Product Interpreter or EXEC2 program.

\[ \text{SET EXECTRAC} \{ \text{ON} \}_{\text{OFF}} \]
SET FULLREAD

Allows 3270 null characters to be recognized in the middle of the physical screen.

SET FULLREAD \{ON \OFF\}

SET FULLSCREEN

Runs CMS in full-screen mode.

SET FULLSCREEN \{ON OFF SUSPEND RESUME\}

SET IMESCAPE

Indicates whether an escape character is required to execute immediate commands.

SET IMESCAPE \{ON OFF char\}

SET IMPCP

Specifies whether command names that are unrecognized by CMS are considered CP commands and are passed on to CP.

SET IMPCP \{ON OFF\}
SET IMPEX

Controls whether EXEC files are treated as commands.

```plaintext
SET IMPEX {ON OFF}
```

SET INPUT

Controls the translation of a specified character "a" to hexadecimal code xx for characters entered from the terminal and the reset of the hexadecimal code xx to the specified hexadecimal code yy in your translate table.

```plaintext
SET INPUT [a xx ]
[xx yy ]
```

SET INSTSEG

Specifies whether the system should search the Installation Discontiguous Shared Segment (DCSS) to locate an EXEC or Editor Macro.

```plaintext
SET INSTSEG {ON [mode|LAST]}
[OFF]
```

SET LANGUAGE

Changes the current language of your CMS session and any application running on CMS that uses national language support.

```plaintext
SET LANGUAGE [langid] [{options...}]
```

**options:**

- `ADD applid`
- `DELETE applid`
- `USER SYSTEM ALL`
- `TYPE NOTYPE`

Chapter 4. CP, CMS, GCS, TSAF, RSCS, and IPCS Commands 205
SET LDRTBLS

Defines the number (nn) of pages of storage to be used for loader tables.

```
SET LDRTBLS [nn]
```

SET LINEND

Activates and/or defines the logical line end for full-screen CMS.

```
SET LINEND { ON } [char] { OFF }
```

SET LOCATION

Specifies whether the location indicator should be displayed in the window when the data in the virtual screen exceeds the size of the window.

```
SET LOCATION wname { ON } { OFF }
```

SET LOGFILE

Indicates whether a log file should be updated with the data being written to the virtual screen.

```
SET LOGFILE vname { ON } { OFF } fn LOGFILE fm
```

SET NONDISP

Defines a character to be used in place of nondisplayable characters.

```
SET NONDISP [char]
```
SET NONSHARE

Specifies a non-shared copy of a normally shared named system.

```
SET NONSHARE

CMS

CMSDOS
CMSVSAM
CMSAMS
CMSBAM
```

SET OUTPUT

Controls the translation and reset of the specified hexadecimal representation xx to the specified character "a" for all xx characters displayed at the terminal.

```
SET OUTPUT [xx a]
```

SET PROTECT

Specifies whether the CMS nucleus is protected against writing in its storage area.

```
SET PROTECT ON
OFF
```

SET RDYMSG

Indicates whether the standard CMS ready message or a shortened form of the CMS ready message is used.

```
SET RDYMSG LMSG
SMSG
```
SET REDTYPE

Controls whether CMS error messages are typed in red for certain terminals equipped with the appropriate terminal feature and a two-color ribbon.

SET REDTYPE {ON}{OFF}

SET RELPAGE

Releases or holds the page frames of storage and sets them to binary zeros, after the following commands complete execution: ASSEMBLE, COPYFILE, COMPARE, EDIT, MACLIB, SORT, TXTLIB, UPDATE, HELP, and the program product language processors supported by VM/SP.

SET RELPAGE {ON}{OFF}

SET REMOTE

Controls the display of data transmissions.

SET REMOTE {ON}{OFF}

SET RESERVED

Specifies whether the number of lines in a window are to be used to display virtual screen reserved lines.

SET RESERVED wname {rtop}{rbot}
SET SYSNAME

Allows for the replacement of a saved system name entry in the SYSNAMES table with the name of an alternative, or backup system.

```
SET SYSNAME [CMSDOS CMSVSAM CMSAMS CMSBAM] entryname
```

SET TEXT

Activates character code conversion of TEXT characters for windows.

```
SET TEXT {ON OFF}
```

SET TRANSLATE

Suppresses translations and translation synonyms of command names for a language.

```
SET TRANSLATE {ON OFF} [USER SYSTEM [TRANslate [APPLID applid]]] [SYNonym [ALL BOTH]]
```

SET UPSI

Controls the setting of the UPSI (User Program Switch Indicator) byte to the specified bit string of 0's and 1's or to binary zeros.

```
SET UPSI {nnnnnnnnn OFF}
```
SET VSCREEN

Indicates what action should take place when the virtual screen is updated with data.

SET VSCREEN vname
  { [Type] [PRotect] [High]
    [NOTYPE] [NOProtect] [NOHigh]
    [color] [exthi] [psset] }

SET WINDOW

Specifies whether the window is to be variable or fixed size.

SET WINDOW wname
  { [VARiable] [POP]
    [FIXed] [NOpop] [NOTop] }

SET WMPF

Defines a WMPF key to execute a windowing command.

SET WMPF nn
  { [pseudonym] [keyword] [string] [NOWRITE] [DEPRECATED] string }
Sets special CP preferred options.

```
SET

  AFFInity [userid] [ON|OFF|nn]
  CPAssist [ON|OFF] [PROC[nn]]

  FAVORed userid [nnn]

  JOurnall {LOGon}[ ON]
  {LINK}[ OFF]

  PRIORity userid nn
  QDROP userid [ON|OFF][USERS][NOQ3]
  REServe userid [nnn|OFF]
  SASSsist [ON|OFF] [PROC[nn]]
  S370E [ON|OFF] [[PROC]addr]
```

Establishes disposition for log messages and dumps.

```
SET

  DUMP {AUTO [CP
        raddr] [ALL]

  LOGmsg [nn [text]]

  MITime {class {mm:ss} [class {mm:ss}...]
          OFF [OFF]...]

  OFF
```
SET

CP Class E

Sets SRM function and the number to be used in the working set size estimate control algorithm.

```
SET
  PAGING nn
  SRM
    APAGES nnnn
    DSPslice nnn
    IB n
    MAXDrum {nnnn|OFF}
    MAXWss {nnnn|OFF}
    MHHFULL {nnnn|OFF}
    PB nn
    PCI {DRUM|DISK}
    PGMTlim
```

SET

CP Class F

Sets recording mode for a device, or enables/disables soft machine check interrupts.

```
SET RECORD
  OFF
  ON raddr LIMIT nn BYTE nn BIT n [ { AND } BYTE nn BIT n ]
  MODE { RETRY } { QUIet } [ cpuid ]
          { MAIN } { Record }
```
The SET command controls various functions within your virtual machine.

<table>
<thead>
<tr>
<th>Command</th>
<th>Options</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACNT</td>
<td></td>
</tr>
<tr>
<td>AUTOPoll</td>
<td></td>
</tr>
<tr>
<td>RUN</td>
<td>{ON}</td>
</tr>
<tr>
<td>LINEDit</td>
<td>{OFF}</td>
</tr>
<tr>
<td>NOTrans</td>
<td></td>
</tr>
<tr>
<td>ECmode</td>
<td></td>
</tr>
<tr>
<td>CONCeal</td>
<td></td>
</tr>
<tr>
<td>ISAM</td>
<td></td>
</tr>
<tr>
<td>PAGEX</td>
<td></td>
</tr>
<tr>
<td>EMSG</td>
<td>{ON</td>
</tr>
<tr>
<td>TIMER</td>
<td>{ON</td>
</tr>
<tr>
<td>CPUid</td>
<td>bbbbbb</td>
</tr>
<tr>
<td>IMSG</td>
<td>{ON}</td>
</tr>
<tr>
<td>MSG</td>
<td>{OFF}</td>
</tr>
<tr>
<td>SMsg</td>
<td>{IUCV}</td>
</tr>
<tr>
<td>WNG</td>
<td></td>
</tr>
<tr>
<td>CPCONIO</td>
<td>{OFF}</td>
</tr>
<tr>
<td>VMCONIO</td>
<td>{IUCV}</td>
</tr>
<tr>
<td>ASSist</td>
<td>{ON</td>
</tr>
<tr>
<td></td>
<td>{OFF}</td>
</tr>
<tr>
<td>AFFInity</td>
<td>{ON</td>
</tr>
<tr>
<td>PFnn</td>
<td>{IMMed}</td>
</tr>
<tr>
<td></td>
<td>{DELAYed}</td>
</tr>
<tr>
<td></td>
<td>{TAB n1 n2...}</td>
</tr>
<tr>
<td></td>
<td>[COPY (resid)]</td>
</tr>
<tr>
<td></td>
<td>[COPY (cUU)]</td>
</tr>
<tr>
<td></td>
<td>[COPY (luname)]</td>
</tr>
<tr>
<td></td>
<td>[COPY (Laddr)]</td>
</tr>
<tr>
<td>PFnn</td>
<td>RETrieve</td>
</tr>
<tr>
<td>VMSAVEl</td>
<td>{ON</td>
</tr>
<tr>
<td>STBypass</td>
<td>{nnnnnnK [NOVERIFY]}</td>
</tr>
<tr>
<td></td>
<td>{nnM}</td>
</tr>
<tr>
<td></td>
<td>{VR}</td>
</tr>
<tr>
<td></td>
<td>{OFF}</td>
</tr>
</tbody>
</table>
SET

SET (continued)  CP Class G

\[
\begin{align*}
\text{STMulti} & \{ [n \text{ \(\left[\right.\) \text{USEG xx} \text{ \(\right]\) \text{CSEG yyy}] \left.\right]} ] \} \\
370E & \{ ON \text{ \(\left[\right.\) OFF} \\
\text{MIH} & \{ ON \left.\right] \text{ \(\right]\) \text{OFF} \} \\
\text{SVCAcc} & \{ ON \text{ \(\left[\right.\) OFF} \\
\end{align*}
\]

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

SET  GCS

Replaces a saved system name entry for VSAM in the SYSNAMES table.

SET

\[
\text{SYSNAME} \left\{ \begin{array}{c}
\text{GCSVSAM} \\
\text{GCSBAM}
\end{array} \right\} \text{ entry name}
\]

SET  RSCS

Requests or disables console message routing. (For authorized alternative operator only)

SET  TSAF

\[
\text{ETRACE} \left\{ \begin{array}{c}
\text{ON} \\
\text{OFF}
\end{array} \right\}
\]

Enables or disables external tracing. Only TSAF virtual console or the secondary user of the TSAF virtual machine can issue this command.

VM/SP Quick Reference
SETKEY

Sets the storage key for a discontiguous saved segment.

SETKEY key systemname [startadr]

SETPRT

Loads a virtual 3800 printer. Command is valid only for the 3800.

SETPRT

CHars[()ccc...[]]
COpies[(]lnnn[])
COPYnr[(]lnnn[])
Fcb[(]fff[])
FLash[(]lid nnn[])
Init
Modify[(]lmmm[n][])

SHOW WINDOW

Places a window on top of all other displayed windows and connects a window to a virtual screen.

SHOW WINdow wname [ON vname [line col]]

SHUTDOWN

Checkpoints and terminates the current VM/370 operation.

SHUTDOWN

REIPL[addr]
POWEROFF
SHUTDOWN

SHUTDOWN

RSCS

Stops RSCS operations in an orderly fashion. Issues DRAIN to all active links, unless faster termination is requested by the QUICK command. Deactivates the RSCS/VTAM interface (if it is active). (For RSCS operator only)

SHUTDOWN [QUICK]

SIZE WINDOW

SIZE WINDOW

CMS

Changes the number of lines and columns for a specified window.

SIZE WINdow { wname } lines [cols]

SLEEP

CP Class Any

Places the virtual machine in a dormant state with the terminal keyboard entry blocked. Allows message display.

SLeeP [nn SEC [MIN [HRs]]]

SMSSG

CP Class G

Sends a "special message" to a virtual machine which is running with SET SMSG ON.

SMssg userid msgtext
SMSG

Delivers the command text to the RSCS virtual machine to be executed. All RSCS commands that are issued by a virtual machine user (including authorized alternative operators) must be included as text in an SMSG command. (The exception is when a local installation has provided an EXEC for each command that automatically puts the "SMSG rcsid" characters in front of the RSCS command expression).

\[
\text{SM} \text{sg rcsid } \{ \text{command-text} \\
\hspace{1cm} \text{Msg nodeidm userid message-text} \\
\hspace{1cm} \text{CMD nodeidc CMD-command-text} \}
\]

SNTMAP

Processes DMKSNT macro definitions and produces two CMS files; a saved segment DASD map and a virtual memory map.

\[
\text{SNTMAP (fn (ft (fm)))}
\]

SO

CMS Immediate Command

Suspends the recording of trace information during the execution command or program.

\[
\text{SO}
\]

SORT

CMS

Sorts records within a file and creates a new file containing the sorted records.

\[
\text{SORT fileid1 fileid2}
\]
SPACE

Forces single spacing on the printer.

SPACeraddr

SPGEN

Does various system generation and maintenance functions, using the parameters contained in SPGEN PROFILE. These functions include:

- Creating, verifying, and displaying system profile parameters.
- Assembling system files.
- Generating CP, CMS, and GCS nuclei.
- Receiving and verifying load maps.

SPGEN

CREATE [optionA[)]

VERIFY [optionA[]]

DISPLAY [ALL compid [optionA[]]

ASSEMBLE fn compid CP [optionA[]]

SETUP compid [optionA[]]

NUCLEUS compid [optionA optionB optionC[]]

MAP compid [optionA[]]

DTYPE vdev

optionA:

PROFILE profname

optionB:

NOIPL

optionC:

NOSETUP
**SPLOAD CMS**

Loads contents of the VM/SP product tapes to appropriate minidisks during initial VM/SP installation.

```plaintext
SPLOAD  group  element  [fn  [ft]]
```

---

**SPMODE CP Class A**

Establishes or resets the single processor mode.

```plaintext
SPMode \{ON \}
```

```plaintext
\{OFF \}
```
SPOOL

Changes spooling control options.

SPOOL {Reader vaddr} \[ \{ \text{Class } \{ * \} \{ \text{CONT} \{ \text{EOF} \{ \text{HOLD} \{ \text{NOCont} \{ \text{NOEof} \{ \text{NOHold} \} \} \} \} \} \} \]^{1}

\{Printer \text{PUnch CONsole vaddr} \}

\{ \text{To For } \{ \text{userid} \{ \text{HOLD} \{ \text{CONT} \{ \text{NOCont} \} \} \} \} \} \]^{1}

\{OFF \{ \text{CLASS c} \{ \text{COPY[*]nnn} \} \} \} \]^{1}

\{ \text{CLOSE PURGE} \}

\{ \text{FLash name nnn} \}^{3}

\{ \text{FORM } \{ \text{form} \{ \text{OFF} \} \} \}

\{ \text{DEST } \{ \text{dest} \{ \text{OFF} \} \} \}

\{ \text{Chars name1} \}

\{ \text{Chars name2} \}

\{ \text{Chars name3} \}

\{ \text{Chars name4} \}

\{ \text{FCB name} \}

\{ \text{Dist } \{ \text{distcode \{OFF \}} \} \}

\{ \text{START } \{ \text{START } \} \}

\{ \text{TERM } \{ \text{TERM } \} \}

^{1} \text{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} \text{These options apply only to a virtual spooled console.}

^{3} \text{These options can only be used to modify a virtual spooling printer. These options only apply to a device type 3800 as a virtual spooling device.}
Dumps output spool files to tape or loads output spool files from tape.

SPTape (STOP raddr)

CANCEL raddr

SCAN raddr option2

LOAD raddr SADump option2

LOAD raddr {Printer spoolid1 [spoolid2] END Class cl [c2[c3[c4]]][FORM form][DEST dest] FORM form [DEST dest] DEST dest ALL}

DUMP raddr {Printer spoolid1 spoolid2 END Class cl[c2[c3[c4]]][FORM form][DEST dest] FORM form [DEST dest] DEST dest ALL}

options:

option1 [MODE 800 1600 6250 38K]

option2 [LEAVE REW ind]

option3 [SYSHOLD USERHOLD NOHOLD]

option4 [PURGE]
SSERV

In CMS/DOS, copies, displays, prints, or punches a book from a DOS/VSE source statement library.

SSERV sublib bookname[ft] [[options...][]]

options:

[DISH] [PRINT] [PUNCH] [TERM]

START

Restarts a drained device or changes its output spooling class.

START [ALL]

Printer
PUnch
Reader

raddr[Class c...][FORM form][DEST dest][NOsep][Auto][NO3800]
lprr [FORM *][DEST OFF][DEST *][SETup][BEG3800]

[Manual][ANY3800]

Flash name
Chars name
FCB plpi
Image imagelib
PUrge

START

Begins program execution.

START [entry [args...]]

* (option[])

option: [NO]
Activates a specified communication link.

```
START [linkid] [Class c ]
[DP dpriority ]
[FOrm name ]
[LINE vaddr ]
[LOGMode logmodename ]
[LUName luname ]
[MANUAL|AUTO|SETup ]
[Queue [Priority|Fifo|Size] ]
[TRACE [ALL|LOG] ]
TYPE {NJE
  SNANJE
  RJE
  MRJE
  3270P
  SNA3270P

  [OParm operation parameters...]
  [Parm [operation parameters...]]
```

**Note:** Any combination of keywords with associated options may be entered in any order, except that Parm keyword must be the rightmost keyword.
STAT

STAT

Lists current status of a problem, a specific subset of problems, or all problems.

STAT

n
nnnn

ALL

\{\text{OPENUSER}\}
\{\text{OPNUSR}\}

\{\text{OPENIBM}\}
\{\text{OPNIBM}\}

\{\text{OPEN}\}
\{\text{OPN}\}

\{\text{APARED}\}

\{\text{NEEDINFO}\}
\{\text{NDINFO}\}

\{\text{PTFRCVD}\}
\{\text{PTFRCV}\}

\{\text{PTFON}\}

\{\text{CLOSED}\}

1 \text{ ABend}

2

\{\text{DOC}\}
\{\text{DD}\}

\text{INcorr}
\text{INF}

\{\text{LOOP}\}
\{\text{LP}\}

\text{MSg}

\{\text{PERFORM}\}
\{\text{PR}\}

\text{WAIT}
\text{WS}

\{\text{PTFERROR}\}
\{\text{PE}\}

1 One of these status keywords may be specified with the ALL operand.

2 One of these failure keywords may be specified with the ALL operand.

STATE/STATEW

STATE verifies the existence of a file on any accessed disk. STATEW verifies the existence of a file on a read/write disk.

STATE \{\text{fn}\} \{\text{ft}\} [\text{fm}]

STATEW \{\text{*}\} \{\text{*}\} [\text{*}]

224 VM/SP Quick Reference
STCP

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

Quickly deactivates a specified link without completing transmission of a file.

```
STOP [linkid]
```

STOP TSAF

Stops the TSAF virtual machine. Only the TSAF virtual console or the secondary user of the TSAF virtual machine can issue this command.

```
STOP TSAF
```
STORE

Alters virtual machine storage, PSW, and registers.

STORE

<table>
<thead>
<tr>
<th>hexloc</th>
<th>hexword1[hexword2...]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lhexloc</td>
<td></td>
</tr>
<tr>
<td>Shexloc</td>
<td>hexdata</td>
</tr>
<tr>
<td>{Greg}</td>
<td>hexword1[hexword2...]</td>
</tr>
<tr>
<td>{Xreg}</td>
<td></td>
</tr>
<tr>
<td>[Yreg]</td>
<td>hexdata1[hexdata2...]</td>
</tr>
<tr>
<td>Psw</td>
<td>[hexword1] hexword2</td>
</tr>
<tr>
<td>STATUS</td>
<td></td>
</tr>
</tbody>
</table>

SVCTRACE

Records information about supervisor call instructions.

SVCTrace

<table>
<thead>
<tr>
<th>ON</th>
</tr>
</thead>
<tbody>
<tr>
<td>OFF</td>
</tr>
</tbody>
</table>

SYNONYM

Specifies alternate names for invoking CMS commands.

SYNonym

<table>
<thead>
<tr>
<th>fn</th>
</tr>
</thead>
<tbody>
<tr>
<td>SYNonym</td>
</tr>
<tr>
<td>fm</td>
</tr>
<tr>
<td>Al</td>
</tr>
<tr>
<td>*</td>
</tr>
</tbody>
</table>

(options...[])

options:

<table>
<thead>
<tr>
<th>STD</th>
</tr>
</thead>
<tbody>
<tr>
<td>NOSTD</td>
</tr>
<tr>
<td>CLEAR</td>
</tr>
</tbody>
</table>
SYSTEM

CP Class G

Simulates virtual machine console functions and clears virtual storage and storage keys to binary zeros.

```
SYstem { CLEAR
             RESET
             RESTART }
```

TAG

CP Class G

Appends or queries the TAG text to a VM/SP spool file utilized by subsystems (such as RSCS).

```
TAg DEv { Printer } [tagtext]
          PUnch
          CONsole
          vaddr

FILE spoolid [tagtext]

QUery { Printer
          PUnch
          DEv { CONsole
                vaddr
          FILE spoolid
```
TAPE

Performs tape to disk or disk to tape operations for CMS files.

TAPE

\[
\begin{align*}
\text{TAPE } \text{DUMP} & \hspace{1em} \{ fn \} \{ ft \} \{ fm \} \{ optionA \} \{ optionB \} \{ optionD \} \{ optionE \} \\
\text{TAPE } \text{LOAD} & \hspace{1em} \{ fn \} \{ ft \} \{ fm \} \{ optionA \} \{ optionC \} \{ optionD \} \\
\text{TAPE } \text{SCAN} & \hspace{1em} \{ fn \} \{ ft \} \{ optionA \} \{ optionB \} \{ optionC \} \{ optionD \} \\
\text{TAPE } \text{SKIP} & \hspace{1em} \{ fn \} \{ ft \} \{ optionA \} \{ optionB \} \{ optionC \} \{ optionD \} \\
\text{TAPE } \text{DVOL1} & \hspace{1em} \{ optionA \} \{ optionD \} \\
\text{TAPE } \text{WVOL1} & \hspace{1em} \{ optionA \} \{ optionD \} \\
\text{TAPE } \text{MODESET} & \hspace{1em} \{ optionD \} \\
\text{tapcmd} & \{ optionD \}
\end{align*}
\]

Note: The tapcmd operand can be one of the following:

\[
[\text{BSF}|\text{BSR}|\text{ERG}|\text{FSF}|\text{FSR}|\text{REW}|\text{RUN}|\text{WTM}]
\]

optionA: \hspace{1em} optionB: \hspace{1em} optionC:

\[
\begin{align*}
\text{WTM} & \hspace{1em} \text{BLKSIZE 4096} & \text{NOPRint} & \text{EOF n} \\
\text{NOWTM} & \hspace{1em} \text{BLKSIZE 800} & \text{PRint} & \text{EOT} \\
 & & \text{DISK} & \text{EOF 1} \\
 & & \text{Term} & \\
\end{align*}
\]

optionD:

\[
\begin{align*}
\text{TAPn} & \hspace{1em} \text{cu} & \text{TRTCH a} & \text{7TRACK} & \text{DEN den} \\
\text{TAP1} & \hspace{1em} \text{181} & & \text{9TRACK} & \\
\end{align*}
\]
**TAPE (continued)**

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LEAVE</td>
<td></td>
</tr>
<tr>
<td>REWIND</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TRANSfer BUFFERed</td>
<td></td>
</tr>
<tr>
<td>TRANSfer IMMEDIATE</td>
<td></td>
</tr>
</tbody>
</table>

**TAPEMAC**

Creates a CMS MACLIB from an unloaded partitioned data set (PDS) from a tape created by the IEHMOVE utility program under OS.

```latex
TAPEMAC fn [SL [labeldefid]] [(options...[])]
   NSL filename [ID=identifier]
```

**Options:**

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TAPn</td>
<td>ITEMCT yyyy</td>
</tr>
<tr>
<td>TAP1</td>
<td>ITEMCT 50000</td>
</tr>
</tbody>
</table>

**TAPPDS**

Creates CMS disk files from tapes which are input to or output from the IEBPUPCH, IEBUPDTE, and IEHMOVE OS utility programs.

```latex
TAPPDS fn [ft [fm]] [SL [labeldefid]]
   NSL filename [ID=identifier]
```

**Options:**

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PDS</td>
<td>COL1</td>
</tr>
<tr>
<td>NOPDS</td>
<td>NOCOL1</td>
</tr>
<tr>
<td>UPDATE</td>
<td>TAPn</td>
</tr>
<tr>
<td>END</td>
<td>NOEND</td>
</tr>
<tr>
<td>MAXTEN</td>
<td>NOMAXTEN</td>
</tr>
</tbody>
</table>
**TE**

**CMS Immediate Command**

Trace end command stops all tracing of your System Product interpreter or EXEC 2 program or macro.

**TE**

---

**TELL**

Send a message to one or more computer users who are logged on to your computer or to one attached to yours via RSCS.

**TELL** name message
Controls virtual console functions.

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

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

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

3 The SCRNSAVE operand is not provided for VM/VTAM and remote terminals.
TRACE

Traces and records program execution.

```
TRACE (SVC
   I/O
   PROgram
   EXTernal
   PRIV
   SIO
   CCW
   BRanch
   INSTRUCT
   ALL
   CSW
   END) 1 PRINTER

TRACE SVC PROGRAM SIO PRINTER.
```

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

```
TRACE SVC PROGRAM SIO PRINTER.
```

---

TRACE

Monitors line activity on a specified link.

```
TRACE [linkid] [ALL] [TO userid [nodeid]
   LOG
   NOLOG
   OFF
```

---

TRANSFER

Transfers closed reader spool files.

```
TRANSFER [userid] [Printer]
   [SYSTEM]
   [Punch]
   [Reader]
   [spoolid]
   [Class c]
   [FORM form]
   [DEST dest]
   [ALL]
   [TO userid]
   [FROM userid]
   [Printer]
   [Punch]
   [Reader]
```

TRANSFER

CP Class G

Transfers closed reader spool files.

TRANSFER

changes the destination address for specified files.

General User Format:

TRANSFER [*]spoolid TO nodeid [userid]

Operator Format:

TRANSFER [linkid] spoolid [spoolid...] TO nodeid [userid]

Chapter 4. CP, CMS, GCS, TSAF, RSCS, and IPCS Commands 233
TRAPRED

TRAPRED

Accesses the CPTRAP reader file and reviews the entries contained in that file. Once TRAPRED is invoked, you may execute TRAPRED subcommands.

TRAPRED filenum

TRAPRED subcommands:

typenum [Vmblok address]
      [DEVaddr cuu]
      [C0de code-value]
      [MACHtype {nn {machname}}]
      [OFF]

ALL [ON]
     [OFF]

Hex

F0rmat

TOP

BOTtom

Up [1/n]  Down [1/n]

Type [1/n]  TYPEBack [1/n]

Printer [1/n]

QUIT

Note: The TRAPRED subcommands must be entered on separate command lines.

---

TS

CMS Immediate Command

Trace start command starts tracing your System Product Interpreter or EXEC 2 program or macro.

TS
TXTLIB

Updates a library of TEXT files (object modules).

```
TXTLib (GEN libname fn1 [fn2...] [(optionA [])])
   ADD libname fn1 [fn2...] [(optionA [])]
   DEL libname membername1 [membername2...]
   MAP libname [(optionB...())]
```

- **optionA:** [FILENAME]
- **optionB:**
  - TERM
  - DISK
  - PRINT

TYPE

Types or displays all or part of a file at a terminal.

```
Type fn ft [fm] [recl[recm]] [(options...[])]
```

- **options:**
  - COL {xxxxx} - {yyyyy} [HEX] [MEMBER{* [MEM {name}]}
  - l
  - mem
  - lrecl

UNLOCK

Releases pages of storage.

```
UNLOCK (userid) firstpage lastpage
```

- **SYSTEM**
  - VIRT=REAL
  - V=R
UPDATE

Make changes in a program source file as defined by control statements in a record file.

```
Update [fn1 ft1] [fn2 [ft2 [fm2]]] [options...[]]
```

Options:

- `REP`
- `SEQ8`
- `INC`
- `CTL`
- `STK`
- `TERM`
- `PRINT`
- `STOR`
- `OUTMODE fm`

Control Statements:

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

UTILITY

Allows you to do some occasionally used utility functions (i.e. print off system definition files, create a stand-alone service utility tape, and write back-up copy of the CP nucleus to tape.

```
UTILITY

PRSAMPLE

UTILITAPE { ALL } { DSF } { FMT } { DDR } { DIR }

IPLDECK { ALL } { DIR } [ctlfile] { DDR } { FMT }

NUCTAPE
```
VALIDATE

Verifies the syntax of a file identifier (filename filetype filemode). If the filemode is specified, VALIDATE verifies whether or not the disk is accessed.

VALIDATE \{fn\} \{ft\} [fm]

VARY

Varies the availability of a device.

VARY \{ONLine\} \{Offline\} \{raddr...\} \{raddr-raddr\} \{lprr\} \{PROCessr\}

Offline PROCessr nn \{VPHY\} \{FORCE\}

VMDUMP

Dumps storage for virtual machine. It enables sending dumps to other users. Used in conjunction with VM/IPCS.

VMDUMP \{hexloc1\} \{:\} \{hexloc2\} \{END\} \{TO \*\} \{TO userid\} \{SYSTEM\} \{FORMAT vmtype\} \{bytecount\} \{DSS\} \{*dumpid\} \{END\}
VMFASM

Updates a specified source file according to entries in a control file and assembles the updated source file.

VMFASM fn ctlfile [(options...)]

options:

<table>
<thead>
<tr>
<th>DISK</th>
<th>TERM</th>
<th>LIST</th>
<th>DECK</th>
<th>RENT</th>
<th>EXP</th>
<th>XREF</th>
<th>MAX</th>
<th>MIN</th>
<th>STD</th>
</tr>
</thead>
<tbody>
<tr>
<td>PRINT</td>
<td>NOTERM</td>
<td>NOLIST</td>
<td>NODECK</td>
<td>NORENT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

VMFDOS

Creates CMS files containing VSE modules for specific installation purposes only. VMFDOS uses either a VSE distribution library tape or VSE SYSIN tape to install only DOS/VS RPG II and VSE/VSAM Program Products.

VMFDOS [LOAD] [SCAN] [181 182] [PRIVATE] [fn] [(options...)]

PRIV or SYST options:

CSL

*  
xxx*  
    {module name}

RL

*  
yyy*  
    {module name}

SL

*  
zzz*  
    {module name}

SYSIN options:

{ALL  
SELECT}
VMFLKED

VMFLKED

Invokes the CMS LKED command to link-edit modules into a loadlib.

VMFLKED fn [ft [fm]] [((options []])]

options:
    PRINT
    MODULE module-name

VMFLOAD

Generate a new CP nucleus, or stand-alone dump (DDR) program. The VMFLOAD program uses two files, a loadlist EXEC file and a control file, to produce a punch file that has several object modules.

VMFLOAD loadlist ctlfile [langid]

VMFMAC

Updates Macro libraries. If you specify a control file, the EXEC invokes the CMS UPDATE command to update specified copy or macro files, according to entries in a control file, and then builds a new macro library from the resulting new versions of those files.

VMFMAC libname [ctlfile]

VMFMERGE

Applies PTFs from the DELTA disk to the Merge disk.

VMFMERGE prodid {{PTF [ptfnum | *]} [EXCLUDE exclist]}
    [PTFLIST applist]
**VMFNLS**

Automatically applies updates to national language-related source files, generates text files, and renames the text files so they can be loaded into the system.

```
VMFNLS fn ft ctlfile [(options...[])]
```

---

**VMFPLC2**

Loads source code from the Product Tape, loads the service installation VMSERV EXEC from the Program Update Tape, dump CMS-formatted files from disk to tape, and loads previously dumped files from tape to disk. The VMFPLC2 command does not process multi-volume files. Files processed by the VMFPLC2 command must be CMS-formatted.

```
VMFPLC2 DUMP {~n} {~t} [~m] [(OptionA,optionB,optionC,optionD)]
LOAD {~n} {~t} {~m} [(OptionB,optionC,optionD,optionE,optionF)]
SCAN {~n} {~t} [(OptionB,optionC,optionD,optionF)]
SKIP {~n} {~t} [(OptionB,optionC,optionD)]
MODESET [(optionD)]
tapcmd [n] [(optionD)]
```

<table>
<thead>
<tr>
<th>optionA:</th>
<th>optionB:</th>
</tr>
</thead>
<tbody>
<tr>
<td>WTM NOWTM</td>
<td>NOPrint</td>
</tr>
<tr>
<td></td>
<td>PRINT</td>
</tr>
<tr>
<td></td>
<td>Term</td>
</tr>
<tr>
<td></td>
<td>DISK</td>
</tr>
<tr>
<td></td>
<td>APPend</td>
</tr>
</tbody>
</table>
VMFPLC2 (continued)

optionC:

```
[ EOT ]
[ EOF n]
[ EOF ]
```

optionD:

```
[
  [7TRACK]
  [9TRACK]
  [18TRACK]
]
```

optionE:

```
[SELect] [STOP]
```

optionF:

```
[DATE]
```

---

VMFREMOV

Removes PTFs applied by the VMFMERGE EXEC procedure.

```
VMFREMOV prodid [PTF {ptfnum | *}] PTFLIST remlist
          [CONVERT [lastfilemode]]
```

---

VMFTXT

Creates text libraries. VMFTXT rebuilds a named TXTLIB file using a member list in an EXEC file with the same name.

```
VMFTXT libname [ctlfile]
```

---

VMFZAP

Applies ZAPs and maintains a record of them in the ZAP Log. This exec uses the Base disk, Merge disk and the ZAP disk as inputs and produces an updated ZAP disk as output.

```
VMFZAP prodid
```
WAITREAD VSCREEN

Updates the virtual screen with data, rebuilds the screen image (updates the 3270 data buffers), and waits for the next attention interrupt.

WAITREAD VSCREEN vname

WAITT VSCREEN

Updates the virtual screen with data.

WAITT VSCREEN [ vname ]

WARNING

Sends high priority messages.

Warning { userid msgtext
    Wng { Operator msgtext
        ALL } 

WRITE VSCREEN

Enters information into a virtual screen.

```
WRITE VSCREEN vname line col length [([REServed] [optionA] [optionB] [optionC] [optionD] [)])
```

- **optionA:** [BLANKs] [NULLs]
- **optionB:** [PROtect] [NOtect] [High] [NOHigh] [Invisible]
- **optionC:** [color] [exthi] [psset]
- **optionD:** [FIELD] [DATA] [COLOR] [EXTTH] [PSS] text

**Note:**
If optionD is used, a right parentheses cannot be used to mark the end of the options.

---

X

CMS Border Command

Maximizes the window.

X
XEDIT

XEDIT

Create, modifies, and manipulates CMS disk files. This is the command used to invoke the VM/SP System Product editor. Once XEDIT is invoked you may execute XEDIT subcommands and use the System Product interpreter or the EXEC 2 macro facility.

Note: In all formats of the XEDIT subcommands and macros, use of the word "subcommand" means an XEDIT subcommand only.

Xedit [fn [ft [fm]]) [[options...][]]

options:

| Width nn | NOSCreen | PROFile macroname | WINDow wname |
| NOClear | NOPROFil | NOMsg | MEMber membername |

options valid only in Update mode:

[Update|NOUpdate] | Seg8|NOSeg8 | Ctl fn1|NOCtl | UNtil filetype | Squares | SIDcode string |

The formats of the XEDIT subcommands and macros, followed by their descriptions, are:

& [subcommand]
    Redisplays the subcommand and allows reexecution by pressing the ENTER key.

? [subcommand]
    Displays the last executed XEDIT subcommand except for the = (equal sign) or the ? (question mark) subcommands.

= [subcommand]
    Reexecutes the last subcommand or macro that was entered. Also executes a specified subcommand and then reexecutes the last one entered.

Add [n|l]
    Inserts blank lines immediately following the current line.

ALL [rtarget]
    Displays a specified collection of lines for editing, while excluding others from the display. This is a macro.
ALTER char1 char2 \[target \[n \[p \]\]\]\]

Changes a single character to another character unavailable on terminal keyboard by referencing its hexadecimal value. This is a macro.

BACKWARD \[n \[* \* \]\]
Scrolls backward the number of screen displays specified.

BOTTOM
Makes the last line of the file or of the range (see SET RANGE) the new current line.

CANCEL
Terminates the editing session for all of the files. This is a macro.

CAPPEND \[text\]
Appends specified text to the end of the current line. This is a macro.

CDELETE \[column-target \]\]
Deletes one or more characters from the current line, starting at the column pointer.

CFIRST
Moves the column pointer to the beginning of the zone (see SET ZONE).

CHANGE /string1/ /string2/ \[target \[p \[q \]\]\]\]

Changes a specified group of characters on one or more lines at one time.
Insert text
Inserts text into the current line immediately ahead of the column pointer.

CLAst
Moves the column pointer to the end of the zone (see SET ZONE).

CLocate column-target
Scans the file for a specified column-target starting at the column following (or preceding) the column pointer in the current line. Also finds successively all occurrences of a character string.

CMS [commandline]
Forces the editor to transmit a command to CMS for execution or causes the editor to enter CMS subset mode.

CMG [text]
Displays a message in the command line; intended for issuance from a macro.

COMMAND [commandline]
Causes the editor to execute a specified XEDIT command without first checking for a synonym or macro with the same name.

COMPress [target|1]
Prepares one or more lines for automatic repositioning of data (see SET TABS).

COPY target1 target2
Copies one or more lines at a specified location in the file.

COUNT /string [/target|1]
Displays the number of times a specified character string appears in one or more lines, beginning at the current line.

COVerlay text
Selectively replaces one or more characters in the current line with the same number of characters keyed in.

CP [commandline]
Transmits commands to the VM/SP control program environment during an editing session.

CReplace text
Replaces one or more characters in the current line.
XEDIT (continued)

CURsor  CMDline [colno] [Priority n]
        Column [Priority n]
        File  lineno [colno] [Priority n]
        Screen lineno [colno] [Priority n]
        Home  [Priority n]

Moves the cursor to a specified position and assigns a priority to the specified position.

DELete [target]

Deletes one or more lines from a file beginning with the current line.

Down [n]*

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

DUPlicat \[n [target]\]

Duplicates one or more lines beginning with the current line.

EMSG [text]

Displays a message at the terminal; or used in macros and modules that interface with XEDIT
and whose messages follow VM/SP message rules. The severity determines whether or not
the alarm sounds.

EXPand [target]

Repositions data in one or more lines that contain tab characters (X'05').
EXTract /operand [/operand [/operand]]...

Used within a macro to get information about internal XEDIT variables or about file data.

operand

may be any one of the keywords listed below.

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACTion</td>
<td>HEX</td>
</tr>
<tr>
<td>ALT</td>
<td>IMage RING VARblank</td>
</tr>
<tr>
<td>APL</td>
<td>IMpcmscp SCALe</td>
</tr>
<tr>
<td>ARBchar</td>
<td>INPmode SCOPE Verify</td>
</tr>
<tr>
<td>AUTosave</td>
<td>LASTLorc SCReen</td>
</tr>
<tr>
<td>BASEft</td>
<td>LASTmsg SELECT</td>
</tr>
<tr>
<td>BRKkey</td>
<td>LENght Seq8 Width</td>
</tr>
<tr>
<td>CASE</td>
<td>LIBName SERIAL WRap</td>
</tr>
<tr>
<td>CMDline</td>
<td>LIBType SHA Dow Zone</td>
</tr>
<tr>
<td>COLOR field</td>
<td>LLine SIDcode =</td>
</tr>
<tr>
<td>COLPtr</td>
<td>LINENd SIZE</td>
</tr>
<tr>
<td>COLUMN</td>
<td>LRecl SPAN</td>
</tr>
<tr>
<td>CTLchar</td>
<td>LScreen SPILL</td>
</tr>
<tr>
<td>CURLIne</td>
<td>MACRO STAY</td>
</tr>
<tr>
<td>CURSor</td>
<td>MASK STREAM</td>
</tr>
<tr>
<td>DISPLAY</td>
<td>MEMBER SYNonym [name</td>
</tr>
<tr>
<td>EFMode</td>
<td>MSGLine SPILL</td>
</tr>
<tr>
<td>EFName</td>
<td>MSGMode STAY</td>
</tr>
<tr>
<td>EFType</td>
<td>NBFile STREAM</td>
</tr>
<tr>
<td>EOF</td>
<td>NONDisp SYNonym [name</td>
</tr>
<tr>
<td>ESCape</td>
<td>NULLs TABLine</td>
</tr>
<tr>
<td>ETARBCH</td>
<td>NUMBER TABS</td>
</tr>
<tr>
<td>ETMODE</td>
<td>PA n[*] TARGET</td>
</tr>
<tr>
<td>Filler</td>
<td>PACK TERMINal</td>
</tr>
<tr>
<td>FILLer</td>
<td>PENDING (see below) TEXT</td>
</tr>
<tr>
<td>FLScreen</td>
<td>PF [n</td>
</tr>
<tr>
<td>FMode</td>
<td>Point [*] TOFEOF</td>
</tr>
<tr>
<td>FName</td>
<td>PREFIX (see below) TOL</td>
</tr>
<tr>
<td>FType</td>
<td>RANGE TRANSLat</td>
</tr>
<tr>
<td>FULLread</td>
<td>RECFm TRunc</td>
</tr>
<tr>
<td></td>
<td>REMote UNIQueid</td>
</tr>
</tbody>
</table>

PENDING [BLOCK] [OLDNAME] name[*] [target1[target2]]

PREFIX [Synonym name[*]]
XEDIT (continued)

FILE

\[
\begin{array}{l}
[fn = \{ft = [fm]\}]
\end{array}
\]

Writes the edited file onto disk and optionally overrides the file identifier originally supplied.

Find text

Searches forward, starting with the current line, for the first line that corresponds to the text specified in the operand.

FINDUp text

Searches backward, starting with the current line, for the first line that corresponds to the text specified in the operand.

FORward [n|*|l]

Scrolls (toward the end of the file) the operand-specified number of screen displays.

GET

\[
\begin{array}{l}
[fn = \{ft = [fm]\} \{firstrec = [numrec]\}]
\end{array}
\]

Inserts all or part of a specified CMS file following the current line of the edit file.

Help [MENU | HELP | name]

Displays a list of all XEDIT subcommands and macros and their descriptions, formats, and parameters, or invokes the CMS HELP command.

HEXType [target|l]

Displays a specified number of lines in both hexadecimal and EBCDIC. This is a macro.

Input [line]

Inserts a single line into a file. Also used to leave edit mode for entry into input mode.
XEDIT

XEDIT (continued)

Join [ALigned] [Column CURSOR]

or

[ALigned] [colno [string] ...]

Combines two or more lines into one replacement line. The first format lets you join two lines at the column pointer or at the cursor. The second format lets you join two or more lines at a specified column number(s) or inserts a specified character string(s) before appending the next line.

LEFT [n|1]

Allows viewing of columns not currently visible on the screen that lie to the left of the first column on the screen.

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

Reads a copy of the file being edited into virtual storage. This subcommand can only be issued from the XEDIT profile. Allows the macro to prompt for edit options or assign default values to edit variables. The LOAD subcommand has the same format and editing options as in the XEDIT command; however, the options specified in the XEDIT command override those specified in the LOAD subcommand.

options:

[Width nn] [NOScreen] [PROFILE macroname] [WINDOW wname]

[NOClear] [NOPROFIL] [NOMsg] [MEMber membername]

options valid only in Update mode:

[Update] [NOUpdate] [Seq8] [NOSeq8] [Ctl fn1] [NOCtl1]

[Merge] [UNTil filetype]

[Incr nn] [SIDcode string]

[Locate] target [subcommand]

Scans file for a specified target, which (when found) becomes the current line.

LOWercase [target|1]

Changes all uppercase letters to lowercase letters in one or more lines.

LPrefix [text]

Simulates writing in the prefix area of the current line.

MACRO [macrol ine]

Causes the specified operand to be executed as a macro.
XEDIT (continued)

MERge target1 target2 [col]
Combines two sets of lines. The first set of lines is deleted and the second set is modified in place.

MODify keyword

keyword operands:

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Operand</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALT</td>
<td>IMPmscp</td>
</tr>
<tr>
<td>APL</td>
<td>LASTlorc</td>
</tr>
<tr>
<td>ARBchar</td>
<td>LNd</td>
</tr>
<tr>
<td>Autosave</td>
<td>LRecl</td>
</tr>
<tr>
<td>BRKkey</td>
<td>MACRO</td>
</tr>
<tr>
<td>CASE</td>
<td>MASK</td>
</tr>
<tr>
<td>CMDline</td>
<td>MSGLine</td>
</tr>
<tr>
<td>COLOR</td>
<td>MSGMode</td>
</tr>
<tr>
<td>COLPtr</td>
<td>NONDisp</td>
</tr>
<tr>
<td>COLUMN</td>
<td>NULLs</td>
</tr>
<tr>
<td>CURLine</td>
<td>PA</td>
</tr>
<tr>
<td>DISPLAY</td>
<td>PACK</td>
</tr>
<tr>
<td>ENTER</td>
<td>PF</td>
</tr>
<tr>
<td>ESCape</td>
<td>PREfix [Synonym name]</td>
</tr>
<tr>
<td>ETARBCH</td>
<td>RANGE</td>
</tr>
<tr>
<td>ETMODE</td>
<td>RECfm</td>
</tr>
<tr>
<td>FILLer</td>
<td>REMote</td>
</tr>
<tr>
<td>FMode</td>
<td>SCALE</td>
</tr>
<tr>
<td>FName</td>
<td>SCOPE</td>
</tr>
<tr>
<td>FType</td>
<td>SCREEN</td>
</tr>
<tr>
<td>FULLread</td>
<td></td>
</tr>
<tr>
<td>HEX</td>
<td></td>
</tr>
<tr>
<td>IMAGE</td>
<td></td>
</tr>
</tbody>
</table>

Displays a subcommand and its current operand values so that new values can be typed over the current ones and the subcommand immediately reentered. This is a macro.

MOve target1 target2
Moves one or more lines, beginning with the current line, to a specified place in the file.

MSG [text]
Displays a message in the message area of the screen.
XEDIT (continued)

Next \[n]\*1
   Advances the line pointer a specified number of lines toward the end of the file.

NFind text
   Searches forward for the first line that does not start with the text specified in the operand.

NFINDUp text
   Searches backward for the first line that does not start with the text specified in the operand.

Overlay text
   Replaces, selectively, one or more characters with non-blank characters starting at the first tab column of the current line.

PARSE startcol Alphaword
   Number
   String ...
   Dblstring
   Target
   Word
   Line
   Helps in writing new macros by scanning the new macro(s) to see if the format-specified-operands match those in the macro. This is a macro.

POWerinp
   Enters an input mode where data can be keyed in as though the screen were one long line.
XEDIT (continued)

PREServe

settings saved include:

<table>
<thead>
<tr>
<th>Variable</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>ARBCHAR</td>
<td>IMPCMSCP</td>
</tr>
<tr>
<td>AUTOSAVE</td>
<td>LASTLORC</td>
</tr>
<tr>
<td>CASE</td>
<td>LINEND</td>
</tr>
<tr>
<td>CMDLINE</td>
<td>LRECL</td>
</tr>
<tr>
<td>COLOR</td>
<td>MACRO</td>
</tr>
<tr>
<td>COLPTR</td>
<td>MASK</td>
</tr>
<tr>
<td>CURLINE</td>
<td>MSGMODE</td>
</tr>
<tr>
<td>DISPLAY</td>
<td>NULLS</td>
</tr>
<tr>
<td>ESCAPE</td>
<td>NUMBER</td>
</tr>
<tr>
<td>ETARBCCH</td>
<td>PACK</td>
</tr>
<tr>
<td>FILLER</td>
<td>PREFIX</td>
</tr>
<tr>
<td>FMODE</td>
<td>RECFM</td>
</tr>
<tr>
<td>FNAME</td>
<td>SCALE</td>
</tr>
<tr>
<td>FTYPE</td>
<td>SCOPE</td>
</tr>
<tr>
<td>HEX</td>
<td>SERIAL</td>
</tr>
<tr>
<td>IMAGE</td>
<td>IMAGE</td>
</tr>
</tbody>
</table>

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

PURge macroname

Removes a copy of a macro in virtual storage.

PUT

\[
\text{PUT} \begin{bmatrix}
\ell & \text{target} & \text{fn} & \text{ft} & \text{fm}
\end{bmatrix}
\]

Inserts one or more lines, starting at the current line, into the end of an existing file or into a new file or into a temporary file created by the editor.

PUTD

\[
\text{PUTD} \begin{bmatrix}
\ell & \text{target} & \text{fn} & \text{ft} & \text{fm}
\end{bmatrix}
\]

Inserts one or more lines, starting with the current line, into the end of an existing file or into a new file or into a temporary file. This command deletes the specified lines from the original file.
Query [option]

options (specify only one each time):

ACTion LASTLorc Seq8
ALT LASTmsg SERial
APL LENgth SHADow
ARBchar LIBName SIDcode
Autosave LIBType SIZE
BASEft Line SPAN
BRKkey LINEND SPILL
CASE LRecl STAY
CMDline LScreen STREAM
COLOR *Ifield MACRO SYNonym [*|name]
COLPtr MASK TABLine
COLUMN MEMBER TABS
CTLchar[char] MSGLine TARGET
CURLine MSGMode TERMINal
CURSor NBFile TEXT
DISPLAY NONDISP TOF
EFMode NULLs TOPEOF
EFName NUMber TOL
EFType PA [n|*] TRANSLat
ENTER PACK Trunc
EOF PENDING (see below) UNIQUEid
EOL PF [n|*] UNTil
EScape Point [*] UPDATE
ETARBCH PREFIX (see below) VARblank
ETMODE RANge Verify
Filler RECFm VERSHift
FMode REMOte Width
FName RESERved WINDOW
FType RING Zone
FULLread SCALE =
HEX SCOPE
IMAGE SCReen
IMPCmscp SELECT

PENDING [BLOCK] [OLDNAME]name|*
PREFIX [SYNonym *|name]

Displays the current setting of various editing options.
XEDIT (continued)

QUIT [n]
Terminates the editing session and leaves the previous copy intact.

READ
   CMDLINE  [Tag  NOTAG]
   All  [Number]
   Nochange  [Number]

Places data from the terminal into the console stack (LIFO). This subcommand generally is issued from a macro.

RECover [n| *| 1]
Replaces a specified number of lines removed by a DELETE or PUTD subcommand or a D (delete) prefix subcommand.

REFRESH
Displays the screen. Issued from a macro, it presents the screen as of that moment in processing, without waiting for input.

RENum [startno  incr]

Renumber the line numbers of VSBASIC and/or FREEFORT files.

REPEat [target| 1]
Advances the line pointer and executes the last subcommand entered.

Replace [text]
Replaces the current line with a specified line or keyed in text, or deletes the current line and enters input mode.

RESet
Removes all prefix subcommands when the screen is in a "pending" or "incomplete" status.
RESTore
Restores the settings of the XEDIT variables to the values in effect when last the PRESERVE subcommand was issued.

RGTLIGHT \[n\]
View columns of data not currently visible on the screen.

RIGHnut [n[l]
Allows viewing of data in columns not currently visible on the screen. These columns are to the right of the right-most column on the screen.

SAVE \[
\]
Enters the file that is currently being edited onto disk without returning control to CMS.

SCHANGE [keynumber]
Locates every occurrence of a string and changes the string only when specified to do so. This is a macro.

SET option
Changes the settings of various editing options while editing is in progress.

options (must specify one):

SET ALT n \[p\]
Changes the number of alterations that have been made to the file since the last AUTOSAVE and/or since the last SAVE.

\[
\]
Shows whether APL keys are available.

\[
\]
Defines an arbitrary character used in a target definition. Note that the initial setting is OFF.
[SET] AUTosave n mode
    OFF A

Sets or resets the automatic save function of the editor. Note that the initial setting is OFF.

[SET] BRKkey ON|OFF

Specifies whether CP should break in when the "BRKKEY" (defined by CP TERMINAL BRKKEY) is pressed.

[SET] CASE Uppercase Respect
    Mixed Ignore

Controls letters entered, and specifies significance in target searches.

[SET] CMDline On
    OFF Top
    Bottom

Specifies the position of the command line on the screen.

[SET] COLOR field [color] [exthi] [High|Nohigh] [PSs]
    *

Associates specific colors with certain areas of the XEDIT screen.

[SET] COLPtr ON|OFF

Determines (on typewriter terminals) whether or not the column pointer (underscore) is displayed.

[SET] CTLchar char Escape
    OFF Protect [color] [exthi]
    [High|NOHIGH|Invisible] [PSs]
    Noprotect[color] [exthi]
    [High|NOHIGH|Invisible] [PSs]
    OFF

Defines control character.
XEDIT (continued)

[SET] CURLine ON M[+n|-n] | [+1|-]n
Defines the nth line of the screen as the current line. Note that, on initial setting, the n is the middle line of the screen.

[SET] DISPLAY n1 [n2|*]
Specifies which selection level of lines (as displayed by SET SELECT) are displayed.

[SET] ENTER [BEFORE] [string]
[SET] ENTER [AFTER] NULLKEY
[SET] ENTER [ONLY] COPYKEY
[SET] ENTER [IGNORE] TABKEY
Defines a meaning for the hardware ENTER key or removes the meaning associated with the ENTER key.

[SET] ESCape ON|OFF [char]
Allows entry of subcommand (on typewriter terminals) when in input mode without leaving input mode.

[SET] ETARBCCH ON|OFF [char]
Defines an extended arbitrary character used in a target definition within a DBCS string. The initial setting is OFF.

[SET] ETMODE ON|OFF
Inform the editor that there are double-byte characters in the file. The initial setting is OFF.

[SET] FILLer [char]
Defines a character to be used when expanding a line (see EXPAND subcommand).

[SET] FMode fm
Changes the filemode of the edited file.

[SET] FName fn
Changes the filename of the edited file.

[SET] FType ft
Changes the filetype of the edited file.

[SET] FULLread ON|OFF
Allows 3270 null characters to be recognized in the middle of screen lines.

[SET] HEX ON|OFF
Allows subcommand operands and targets to be specified in hexadecimal. Note that the initial setting is OFF.

[SET] IMAGE ON|OFF|Canon
Determines how tab characters (X'05') and backspace characters (X'16') are handled.
XEDIT (continued)

[SET] IMPcmsgscp ON|OFF
Determine whether or not non-XEDIT recognized subcommands are transmitted implicitly to
CMS, and later to CP, for execution.

[SET] LASTLorc line
Specifies the contents of the LASTLORC subcommand. (Used within a macro.)

[SET] LINENd ON|OFF [char]
Determines whether or not # (pound sign) or other character is used as the line end
coloracter.

[SET] LRecl n|*
Defines a new logical record length for writing file to disk.

[SET] MACRO ON|OFF
Controls sequence of editor's search for subcommands and macros. Note that the initial
setting is OFF.

[SET] MASK Define
          Immed [text]
          Modify
Changes contents of mask. Note that the initial setting is a blank line.

[SET] MSGLine ON M[+n]|-n | [+|-]n [p|l] [Overlay]
OFF
Defines the location of the message line on the screen, and the maximum number of lines
that a message may occupy.

[SET] MSGMode ON [Short|Long]
OFF
Controls message display. Note that the initial setting is ON LONG.
[SET] NONDisp [char]
Defines a character to use in place of a nondisplayable character.

[SET] NULLs ON|OFF
Specifies whether trailing blanks in each line are written to the screen as blanks (X'40') or nulls (X'00'). Note that the initial setting is OFF.

[SET] NUMber ON|OFF
Determines whether or not line numbers are displayed in prefix area. Note that the initial setting is OFF.

[SET] PAAn [BEFORE] [string] [AFTER] [NULLKEY] [ONLY] [COPYKEY] [IGNORE] [TABKEY]
Defines a meaning for a specified hardware attention (PA) key or removes the meaning associated with the specified PA key.

[SET] PACK ON|OFF
Specifies whether or not packed file is entered on disk.

[SET] PENDING ON|BLOCK|ERROR string OFF
Controls the execution of a prefix macro and the status of the screen while the prefix macro is being executed.

[SET] PFn [BEFORE] [string] [AFTER] [NULLKEY] [ONLY] [COPYKEY] [IGNORE] [TABKEY]
Defines or removes a meaning for a specified program function (PF) key. Note that TABKEY is the initial setting of the PF4 key.

[SET] Point .symbol [OFF]
Defines or redefines the symbolic name for the current line.
XEDIT (continued)

[SET] PREFIX ON [LEFT|Right]
OFF
Nulls [LEFT|Right]

or

PREFIX Synonym newname oldname

Controls display of the prefix area. Also defines a synonym for a prefix subcommand.

[SET] RANGE target1 target2
Defines new limits for line pointer movement.

[SET] RECFm F|V|FP|VP
Defines the record format for the file.

[SET] REMOTE ON|OFF
Controls the way XEDIT handles the display, in terms of data transmission.

[SET] RESERVED M[+n]|-[n] [color] [exthi] [PSs] High
[OFF] [text] Nohigh

Reserves a specific line on the screen for displaying blank or specified information with or without any of the following features: color, extended highlighting, highlighting, and programmed symbol set.

[SET] SCALE ON [M[+n]|-[n] | [±|]n]
OFF

Displays a scale line under the current line (the default) or on a specified line.

[SET] SCOPE Display|ALL
Specifies the set of lines on which the editor operates.

[SET] SCREEN n [HORIZONTAL|Vertical]
Size s1 [s2 [s3...[sn]]]
Width w1 [w2 [w3...[wn]]]
Define s11 sw1 sh1 sv1 [sl2 sw2 sh2 sv2]...

Divides the screen into a specified number of logical screens to allow editing of multiple files or multiple views of the same file.
XEDIT

XEDIT (continued)

[SET] SELet [+|-] n [target]
Designates a "selection level" for specified lines. A selection level is a positive value assigned to a line in a file.

[SET] SERial ON

<table>
<thead>
<tr>
<th>incrno</th>
<th>startno</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>10</td>
</tr>
</tbody>
</table>

ALL

<table>
<thead>
<tr>
<th>incrno</th>
<th>startno</th>
</tr>
</thead>
<tbody>
<tr>
<td>1000</td>
<td>1000</td>
</tr>
</tbody>
</table>

string

<table>
<thead>
<tr>
<th>incrno</th>
<th>startno</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>10</td>
</tr>
</tbody>
</table>

OFF
Controls file serialization.

[SET] SHADow ON|OFF
Displays a notice (called a shadow line) that indicates how many lines have been excluded from the display.

[SET] SIDcode [string]
Inserts a character string in every line of an update file.

[SET] SPAN ON

<table>
<thead>
<tr>
<th>Blank</th>
<th>Nonblank</th>
</tr>
</thead>
<tbody>
<tr>
<td>[n]</td>
<td>[*]</td>
</tr>
</tbody>
</table>

OFF
Specifies whether a target-search character string must be included in one line or span a certain number of lines. Note that the initial setting is OFF Blank 2.
XEDIT (continued)

[SET] SPILL ON|OFF|WORD
Specifies if data is spilled onto new lines or lines are truncated following these subcommands: CHANGE, CINSERT, COVERLAY, CREPLACE, EXPAND, GET, INPUT, MERGE, OVERLAY, REPLACE, SHIFT, (and macros that use these subcommands internally, including CAPPEND, JOIN and PRFSHIFT(>, > >)).

[SET] STAY ON|OFF
Specifies whether or not the line pointer moves when target-search object is not found. Note that the initial setting is OFF.

[SET] STReam ON|OFF
Specifies whether to search entire file or only the current line for a character string. Note that the initial setting is ON.

[SET] SYNonym ON|OFF
or
SYNonym [LINEND char] newname [n] oldname
or
SYNonym [LINEND char] newname [n[format1...formatn]]
oldname[&l...&n]
Specifies whether or not to look for synonyms. Also assigns a synonym to any existing subcommand or macro (except prefix subcommands or prefix macros) and defines an abbreviation for the synonym.

[SET] TABLine ON [M[+n|-n] | [+|-]n]
OFF
Displays a "T" in every tab column according to current tab settings.

[SET] TABS n1[n2 ... n28]
Defines the logical tab stops for a file.

[SET] TERMINal Typewriter
Display
Specifies whether a terminal is to be used in line mode or in full screen mode.
[SET] TEXT ON|OFF
    Shows whether keys are available.

[SET] TOEOF ON|OFF
    Controls the display of Top of File, End of File, Top of Range, and End of Range null lines.
    Note that the initial setting is ON.

[SET] TRANSLat char1 char2 [char1 char2] ...
    OFF
    Controls uppercase translation of specified characters. This option is designed for terminals
    whose keyboards support characters other than English.

[SET] TRunc n|*
    Defines last column in which data may be entered.

[SET] VARblank ON|OFF
    Controls whether or not the number of blanks between two words is significant in target
    search. Note that the initial setting is OFF.

[SET] Verify [ON] [[Hex] startcol endcol] ...
    [OFF]
    Controls whether or not subcommand(s)-changed lines are to be displayed. Also defines
    columns to be displayed on screen. Data may also be displayed in hexadecimal.

[SET] WRap ON|OFF
    Controls use of wraparound. Note that the initial setting is OFF.

[SET] Zone zone1 zone2
    Defines starting and ending column of each record for target search scanning.

[SET]  = string
    Inserts specified string into the equal buffer (see = subcommand).
XEDIT (continued)

SHift Left  
  [cols [target]]

SHift Right 
  [cols [target]]

Moves data either to the left or to the right. Note that data loss is possible.

SI
Continually add lines for indented text to a file. A line is added immediately following the line that contains the cursor. The cursor is positioned at the column where the text on the previous line begins.

SORT target [AD] coll col2 [coll col2]...

Arranges a specified number of file lines in ascending or descending EBCDIC sequence according to specified sort columns.

SOS option

options:

  ALarm   POP
  CLEAR   PUSh
  LINEAdd TABB [n|l]
  LINEDel TABCmd
  NULLs   TABCMDB [n|l]
  NULLs ON TABCMDF [n|l]
  NULLs OFF TABF [n|l]
  PFn

Provides a set of functions used mainly in XEDIT macros or assigned to PF keys.

SPLIT [Aligned] [COLUMN]

or

[Aligned] [colno]

  [BEFORE] /string/ ...
  [After]

Splits a line into two or more lines at the column pointer or at the cursor. The second format splits a line into several lines. This is a macro.
SPLITJOIN

Either splits a line or joins two lines, depending on the position of the cursor on a file line. If the cursor is positioned before or at the last non-blank character, the line is split (at the cursor position). If the cursor is positioned after the last non-blank character on a line (that is, after the end of the data on a line), the next line is appended, starting at the cursor position. This is a macro.

STACK [target [startcol [length]]]

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

STATUS [filename]

Displays the SET subcommand options and their current settings or creates an XEDIT macro that contains the SET subcommands with their current settings. This is a macro.

TOP

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

keywords: (more than one can be specified)

APL   IMage   PFn   TABLine
ARBchar   IMPcmsgcp   Point   TABS
AUTosave   LASTmsg   PREFIX   TARGET
CASE   LEnGth   RANge   TERMINal
CMDline   Line   RECfM   TEXT
COLPtr   LINEnd   RESERved   TOF
COLumn   LRecl   SCALe   TOFFEO
CTLchar[char]   LScreen   SCReen   Trunc
CURLine   MACRO   Seg8   UPDATE
CURSor   MASK   SERial   VARblank
EOF   MSGMode   SIDcode   Verify
ESCAPE   NBFile   SIZE   VERSHIFT
FILrer   NONDisp   SPAN   Width
FMode   NULLs   STAY   WRap
FName   NUMBER   STReam   Zone
FType   PACK   SYNONym[name]   =
HEX

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

Type [target|1]
Displays a specified number of lines, starting with the current line.

Up [n|*|1]
Moves the line pointer a specified number of lines toward the top of the file.

UPPercas [target|1]
Translates all lowercase characters to uppercase ones, starting at the current line.

Xedit [fn [ft [fm]]] [[options... []]]
Edits multiple files in virtual storage.

Options: are the same as the command options (see XEDIT command).
The **XEDIT Prefix subcommands** (line subcommands) are used as follows:

<table>
<thead>
<tr>
<th>Prefix</th>
<th>Meaning and/or Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Add one line immediately after the line in which subcommand is entered</td>
</tr>
<tr>
<td>nA</td>
<td>Add n lines</td>
</tr>
<tr>
<td>An</td>
<td>Add n lines</td>
</tr>
<tr>
<td>C</td>
<td>Copy one line; must have F or P prefix subcommand to indicate destination of line</td>
</tr>
<tr>
<td>Cn</td>
<td>Copy n lines</td>
</tr>
<tr>
<td>nC</td>
<td>Copy n lines</td>
</tr>
<tr>
<td>CC</td>
<td>Copy block of lines</td>
</tr>
<tr>
<td>D</td>
<td>Deletes one line</td>
</tr>
<tr>
<td>Dn</td>
<td>Delete n lines</td>
</tr>
<tr>
<td>nD</td>
<td>Delete n lines</td>
</tr>
<tr>
<td>DD</td>
<td>Delete block of lines</td>
</tr>
<tr>
<td>E</td>
<td>Extend logical line by one more physical line</td>
</tr>
<tr>
<td>F</td>
<td>Data is entered following this point (using the C or M prefix subcommand)</td>
</tr>
<tr>
<td>I</td>
<td>Insert one line immediately following the line in which prefix subcommand is entered</td>
</tr>
<tr>
<td>nl</td>
<td>Insert n lines</td>
</tr>
<tr>
<td>ln</td>
<td>Insert n lines</td>
</tr>
<tr>
<td>M</td>
<td>Move one line from one location to another in the file</td>
</tr>
<tr>
<td>Mn</td>
<td>Move n lines</td>
</tr>
<tr>
<td>nM</td>
<td>Move n lines</td>
</tr>
<tr>
<td>MM</td>
<td>Move block of lines</td>
</tr>
<tr>
<td>P</td>
<td>Data is entered preceding this point (using the C or M prefix subcommands)</td>
</tr>
</tbody>
</table>
**XEDIT**

### XEDIT (continued)

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>S</td>
<td>Show all lines</td>
</tr>
<tr>
<td>S*</td>
<td>Show all lines</td>
</tr>
<tr>
<td>S(n)</td>
<td>Show the first (n) lines</td>
</tr>
<tr>
<td>S(+\ n)</td>
<td>Show the first (n) lines</td>
</tr>
<tr>
<td>nS</td>
<td>Show the first (n) lines</td>
</tr>
<tr>
<td>S-n</td>
<td>Show the last (n) lines</td>
</tr>
<tr>
<td>SCALE</td>
<td>Display the scale on this line</td>
</tr>
<tr>
<td>SI</td>
<td>Continually add lines for indented text to a file. A line is added immediately following the line that contains the cursor. The cursor is positioned at the column where the text on the previous line begins.</td>
</tr>
<tr>
<td>TABL</td>
<td>Displays a “T” in every tab column in the line</td>
</tr>
<tr>
<td>X</td>
<td>Exclude one line from display</td>
</tr>
<tr>
<td>X(n)</td>
<td>Exclude (n) lines</td>
</tr>
<tr>
<td>nX</td>
<td>Exclude (n) lines</td>
</tr>
<tr>
<td>XX</td>
<td>Exclude a block of lines</td>
</tr>
<tr>
<td>.xxxx</td>
<td>Assigns xxxx as symbolic name to this line</td>
</tr>
<tr>
<td>&lt;</td>
<td>Shift one line one column to the left</td>
</tr>
<tr>
<td>&lt;(n)</td>
<td>Shift one line (n) columns to the left</td>
</tr>
<tr>
<td>n&lt;</td>
<td>Shift one line (n) columns to the left</td>
</tr>
<tr>
<td>&lt; &lt;</td>
<td>Shift a block of lines (n) columns to the left</td>
</tr>
<tr>
<td>&lt; &lt;(n)</td>
<td>Shift a block of lines (n) columns to the left</td>
</tr>
<tr>
<td>n&lt; &lt;</td>
<td>Shift a block of lines (n) columns to the left</td>
</tr>
<tr>
<td>/[n]</td>
<td>Make this line current and move the column pointer under the (n)th column</td>
</tr>
<tr>
<td>[n]/</td>
<td></td>
</tr>
<tr>
<td>&gt;</td>
<td>Shift one line one column to the right</td>
</tr>
<tr>
<td>&gt;(n)</td>
<td>Shift one line (n) columns to the right</td>
</tr>
<tr>
<td>n&gt;</td>
<td>Shift one line (n) columns to the right</td>
</tr>
<tr>
<td>&gt; &gt;</td>
<td>Shift a block of lines (n) columns to the right</td>
</tr>
<tr>
<td>&gt; &gt;(n)</td>
<td>Shift a block of lines (n) columns to the right</td>
</tr>
<tr>
<td>n&gt; &gt;</td>
<td>Shift a block of lines (n) columns to the right</td>
</tr>
</tbody>
</table>
### XEDIT (continued)

Duplicates one line (must be a double quote symbol)

Doublc line n times

Duplicate block of lines

Duplicate block n times

### XMITMSG

Retrieves a message from a CMS message repository file or your own message repository file.

XMITMSG msgnumber sublist [(options ... [])]

options:

[FORmat nn]

[LINE nn|*]

[LETter a]

[APPLID applid]

[CALLER name]

[VAR]

[COMPress ] [HEAder ] [DISPlay ] [SYSLANG]

[NOCOMPress ] [NOHEAder ] [NONDISPlay ] [ERRMSG ]
ZAP

Modifies or dumps MODULE, LOADLIB, or TXTLIB files.

\[
\text{ZAP} \begin{cases}
\text{MODULE} \\
\text{LOADLIB} \\
\text{TXTLIB}
\end{cases}
\begin{array}{c}
\text{[libname1 ... libname3][[options...[]]]}
\end{array}
\]

options:
[\text{TERM}INPUT filename] [\text{PRINT}NO\text{PRINT}]

Control Statements:

BASE address

DUMP \begin{cases}
\text{membername} \\
\text{modulename}
\end{cases} \begin{cases}
csectname \\
\text{startaddress} \text{[endaddress]}
\end{cases}
\begin{array}{c}
\text{ALL}
\end{array}

LOG \begin{cases}
\text{fixnum}
\end{cases} \begin{cases}
\text{ZAPLOG}filetype
\end{cases} \begin{cases}
\text{userdata}
\end{cases}

NAME \begin{cases}
\text{membername} \text{modulename}
\end{cases} \begin{cases}
csectname
\end{cases}

REP disp data

\begin{cases}
\text{VERIFY}VER
\end{cases} disp data

* comment

END
Summary of Changes

The HELP Facility

The HELP Facility has been enhanced to include National Language Support and to improve performance and usability. These enhancements include:

- BRIEF layer of HELP for a subset of frequently-used commands
- RELATED information
- New HELP command options
- Toggling ability
- Windowing of the BRIEF layer
- MOREHELP command
- Simplified screens and syntax notation
- New control section keywords
- Enhancement of the DEFAULTS command
- National Language Support

NLS Parser Summary

The CMS parsing facility parses and translates command name arguments. It is important to National Language Support (NLS) because it lets users enter commands in their own national language.
Enhancements for EXECs in Storage

An optional Installation Discontiguous Shared Segement (DCSS) is added to contain frequently used EXECs and Editor Macros that your installation provides. All users can access the DCSS and share the same executing copy of the EXECs.

Advanced Printer Subsystem Support

The SPOOL System Service allows authorized users an interface for communication between CP and a printer subsystem.

The DESTination option allows you to select a specific printer or punch to process your print, punch, or console file.

The CMS PRINT Command has been enhanced to support an Oversize option to print files that have records larger than the virtual printer’s character size.

Transparent Services Access Facility

Transparent Services Access Facility is a facility that lets users connect to and communicate with local or remote virtual machines within a group of systems.
Enhanced 3270 Usability

Usability features for 3270 display stations have been enhanced. The VM logo message at the top of the screen is changed from “VM/370” to “VIRTUAL MACHINE/SYSTEM PRODUCT” and users can now log on directly from the logo screen.

New Commands for Release 5

ALARM VSCREEN  QUERY CURSOR  SET LANGUAGE
CLEAR VSCREEN  QUERY DISPLAY  SET LINEND
CLEAR WINDOW  QUERY FULLREAD  SET LOCATION
CONVERT COMMANDS  QUERY FULLSCREEN  SET LOGFILE
CMSSERV  QUERY HIDE  SET NODISP
CURSOR VSCREEN  QUERY KEY  SET REMOTE
DEFINE VSCREEN  QUERY LANGLIST  SET RESERVED
DEFINE WINDOW  QUERY LANGUAGE  SET VSCREEN
DELETE VSCREEN  QUERY LINEND  SET WINDOW
DELETE WINDOW  QUERY LOCATION  SET WMPF
DISKMAP  QUERY LOGFILE  SHOW WINDOW
DROP WINDOW  QUERY NODISP  SIZE WINDOW
GET VSCREEN  QUERY REMOTE  SNTMAP
HIDE WINDOW  QUERY RESERVED  SPGEN
ITASK  QUERY SHOW  SPLLOAD
LANGGEN  QUERY TEXT  UTILITY
LANGMERG  QUERY VSCREEN  VALIDATE
MAXIMIZE WINDOW  QUERY WINDOW  VMFASM
MINIMIZE WINDOW  QUERY WMPF  VMFDOS
MOREHELP  REFRESH  VMFKED
PARSECMD  RESTORE WINDOW  VMFLKED
POP WINDOW  ROUTE  VMFLOAD
POSITION WINDOW  SCROLL  VMMAC
PRELOAD  SET APL  VMFMERGE
PUT SCREEN  SET BORDER  VMFREMOV
PUT VSCREEN  SET CHARMODE  VMFTEXT
QUERY APL  SET CMSPF  VMFZAP
QUERY BORDER  SET FULLREAD  WAITREAD VSCREEN
QUERY CHARMODE  SET FULLSCREEN  WAIT T VSCREEN
QUERY CMSPF  SET FULLSCREEN  WRITE VSCREEN
QUERY CMSPF  SET FULLSCREEN  XMITMSG
TSAF Commands

- ADD LINK
- DELETE LINK
- QUERY
- RUNTSAF
- STOP TSAF

Changed Commands for Release 5

ACCESS  IPL
BACKSPAC  LOAD
CHANGE  OVERRIDE
CLOSE  PRINT
DEFAULTS  PURGE
DETACH  QUERY
DETACH  REPEAT
DRAIN  SAVENC
DUMPSCAN  SAVENC
EXECDROP  SET
EXECIO  SPOOL
EXECMAP  SPTAPE
EXECUPDT  START
FILEDEF  TERMINAL
FLUSH  TRANSFER
FORMAT  TXTLIB
GLOBAL  UPDATE
HELP  VARY
INCLUDE  XEDIT
Bibliography

Prerequisite Publications

Virtual Machine/System Product:

CMS Command Reference, SC19-6209
CMS Primer, SC24-5236
CMS Primer for Line-Oriented Terminals, SC24-5242
CP Command Reference, SC19-6211
Introduction, GC19-6200
Corequisite Publications

Virtual Machine/System Product:

*Application Development Guide, SC24-5247*

*VM/SP CMS for System Programming, SC24-5286*

*VM/SP CP for System Programming, SC24-5285*

*EXEC 2 Reference, SC24-5219*

*VM/SP System Facilities for Programming, SC24-5288*

*VM/SP Transparent Services Access Facility Reference, SC24-5287*

*IBM Cooperative Processing Programmer’s Guide for VM/SP, SC24-5291*

*Installation Guide, SC24-5237*

*Macros and Functions Reference, SC24-5284*

*Operator’s Guide, SC19-6202*

*Planning Guide and Reference, SC19-6201*

*VM Distributed Data Processing Guide, LY24-2451*

*Remote Spooling Communications Subsystem Networking Program Reference and Operations Manual, SH24-5005*

*System Messages and Codes, SC19-6204*
System Messages Cross-Reference, SC19-5264

System Product Editor Command and Macro Reference, SC24-5221

System Product Editor User’s Guide, SC24-5220

System Product Interpreter Reference, SC24-5239

System Product Interpreter User’s Guide, SC24-5238

Terminal Reference, GC19-6206
Index

Special Characters

* command (CMS) 53
* command (CP class any) 53
* command (GCS) 53
* command (RSCS) 53
#CP command (CP class any) 53

A

ACCESS command (CMS) 54
ACCESS command (GCS) 54
ACNT command (CP class A) 54
ADD LINK command (TSAF) 55
ADSTOP command (CP class G) 55
ALARM VSCREEN command (CMS) 55
AMSERV command (CMS) 55
APAR command (IPCS) 56
ASMGEND command (CMS) 56
ASM3705 command (CMS) 56
ASSEMBLE command (CMS) 57
assembling a source file 24
ASSGN command (CMS) 58
ATTACH command (CP class B) 58
ATTN command (CP class G) 58
AUTOLOG command (CP class A,B) 59

B

B (CMS Border Command) 59
BACKSPAC command (CP class D) 59
BACKSPAC command (RSCS) 59
BEGIN command (CP class G) 59

C

C (CMS Border Command) 60
CATCH command (CMS) 60
CHANGE command (CP class D) 61
CHANGE command (CP class G) 62
CHANGE command (RSCS) 63
CLEAR VSCREEN command (CMS) 63
CLEAR WINDOW command (CMS) 63
CLOSE command (CP class G) 64
CMD command (RSCS) 64
CMDCALL command (CMS) 64
CMS (Conversational Monitor Commands) commands
* 53
  ACCESS 54
  ALARM VSCREEN 55
  AMSERV 55
  ASMGEND 56
  ASM3705 56
  ASSEMBLE 57
  ASSGN 58
  B (CMS Border Command) 59
  C (CMS Border Command) 60
  CATCH 60
  CLEAR VSCREEN 63
  CLEAR WINDOW 63
  CMDCALL 64

Index 281
CMSBATCH 64
CMSSEND 65
CMSSERV 65
COMPARE 65
CONVERT COMMANDS 66
CONWAIT 66
COPYFILE 67
CP 67
CURSOR VSCREEN 69
D (CMS Border Command) 69
DDR 70
DEBUG 72
DEFAULTS 73
DEFINE VSCREEN 77
DEFINE WINDOW 78
DELETE VSCREEN 78
DELETE WINDOW 79
DESBUF 79
DIRECT 80
DISK 82
DISKMAP 82
DLBL 84
DOSLIB 86
DOSLKED 86
DROP WINDOW 87
DROPBUF 87
DSERV 88
EDIT 95
ERASE 102
ESERV 102
EXEC 103
EXEC 2 108
EXECDROP 118
EXECIO 119
EXECLOAD 120
EXECMAP 120
EXECOS 121
EXECSTAT 121
EXECUPDT 121
EXPAND 122
F (CMS Border Command) 122
FETCH 123
FILEDEF 124
FILELIST 126
FINIS 127
FORMAT 128
GENDIRT 130
GENIMAGE 130
GENMOD 131
GENMSG 131
GEN3705 131
GET VSCREEN 132
GLOBAL 132
GLOBALV 133
H (CMS Border Command) 134
HB 134
HELP 135
HELPCONV 135
HI 135
HIDE WINDOW 136
HO 136
HT 136
HX 137
IDENTIFY 137
IMAGELIB 137
IMAGEMOD 138
IMMCMOD 138
INCLUDE 139
ITASK (CMS) 141
L (CMS Border Command) 142
LABELDEF 142
LANGGEN 143
LANGMERG 143
LISTDS 143
LISTFILE 144
LISTIO 144
LKED 145
LOAD 145
LOADLIB 146
LOADMOD 146
M (CMS Border Command) 148
MACLIB 148
MAKEBUF 149
MAXIMIZE WINDOW 149
MINIMIZE WINDOW 150
MODMAP 150
MOREHELP 152
MOVEFILE 152
N (CMS Border Command) 153
NAMEFIND 153
NAMES 154
NCXPUMP 154
NOTE 157
NUCXdROP 157
NUCXLOAD 158
NUCXMAP 158
O (CMS Border Command) 158
OPTION 158
OSRUN 159
P (CMS Border Command) 160
PARSECMD 160
PEEK 161
POP WINDOW 163
POSITION WINDOW 163
PRELOAD 164
PRINT 164
PSERV 168
PUNCH 168
PUT SCREEN 169
PUT VSCREEN 169
QUERY 170
R (CMS Border Command) 179
RDR 179
RDRLIST 179
READCARD 180
RECEIVE 180
REFRESH 181
RELEASE 181
RENAME 182
RESERVE 183
RESTORE WINDOW 183
REXX 184
RO 195
ROUTE 195
RSERV 196
RT 196
RUN 196
S (CMS Border Command) 197
SAVENCP 197
SCROLL 199
SENDFILE 200
SENTRIES 200
SET 200
SET ABBREV 201
SET APL 201
SET AUTOREAD 201
SET BLIP 201
SET BORDER 202
SET CHARMODE 202
SET CMSPF 202
SET CMSTYPE 203
SET DOS 203
SET DOSLNCNT 203
SET DOSPART 203
SET EXECTRAC 203
SET FULLREAD 204
SET FULLSCREEN 204
SET IMESCAPE 204
SET IMPCP 204
SET IMPEX 205
SET INPUT 205
SET INSTSEG 205
SET LANGUAGE 205
SET LDRTBLS 206
SET LINEND 206
SET LOCATION 206
SET LOGFILE 206
SET NONDISP 206
SET NONSHARE 207
SET OUTPUT 207
SET PROTECT 207
SET RDYMSG 207
SET REDTYPE 208
SET RELPAGE 208
SET REMOTE 208
SET RESERVED 208
SET SYSNAME 209
SET TEXT 209
SET TRANSLATE 209
SET UPSI 209
SET VSCREEN 210
SET WINDOW 210
SET WMFP 210
SETKEY 215
SETPRT 215
Index 283
SHOW WINDOW 215
SIZE WINDOW 216
SNTMAP 217
SO 217
SORT 217
SPGEN 218
SPLOAD 219
SSERV 222
START 222
STATE/STATEW 224
SVCTRACE 226
SYNONYM 226
TAPE 228
TAPEMAC 229
TAPPDS 229
TE 230
TELL 230
TRAPRED 234
TS 234
TXTLIB 235
TYPE 235
UPDATE 236
UTILITY 236
VALIDATE 237
VMFASM 238
VMFDOS 238
VMFLKED 239
VMFLOAD 239
VMFMAC 239
VMFMERGE 239
VMFNLS 240
VMFPLC2 240
VMFREMOVE 241
VMFTXT 241
VMFZAP 241
WAITREAD VSCREEN 242
WAITT VSCREEN 242
WRITE VSCREEN 243
X (CMS Border Command) 243
XEDIT 244
XMITMSG 270
ZAP 271
CMS Border Commands
B 59

CMS DDR command return codes 42
CMS fileid 32
CMS immediate commands
HB 134
HI 135
HO 136
HT 136
HX 137
RO 195
RT 196
SO 217
TE 230
TS 234
CMSSERV (CMS command) 65
Control Program (CP)
See CP (Control Program) commands
conventions for notations 48
Conversational Monitor System (CMS)
See CMS (Conversational Monitor Commands) commands
CONVERT command (IPCS) 66
CONVERT COMMANDS (CMS) 66
CONVIPCS EXEC (IPCS) 66
CONWAIT command (CMS) 66
COPYFILE command (CMS) 67
correcting errors 25
COUPLE command (CP class G) 67
CP (Control Program) commands

- LOCK (class A) 147
- LOGOFF (class any) 147
- LOGON (class any) 147
- MESSAGES (class A,B) 149
- MESSAGES (class any) 149
- MIGRATE (class A) 150
- MONITOR (class A,E) 150
- MSGNOH (class A,B) 153
- NETWORK (class A) 155
- NETWORK (class B) 156
- NOTREADY (class G) 157
- ORDER (class D) 159
- ORDER (class G) 159
- PER (class A,B,C,D,E,F,G) 161
- PURGE (class D) 168
- PURGE (class G) 169
- QUERY (class A) 172
- QUERY (class B) 173
- QUERY (class C) 173
- QUERY (class D) 174
- QUERY (class E) 175
- QUERY (class F) 175
- QUERY (class G) 176
- QVM (class A) 179
- READY (class G) 180
- REPEAT (class D) 182
- REQUEST (class G) 182
- RESET (class G) 183
- REWIND (class G) 184
- SAVESYS (class E) 197
- SCREEN (class G) 198
- SEND (class G) 199
- SET (class A) 211
- SET (class B) 211
- SET (class E) 212
- SET (class F) 212
- SET (class G) 213
- SHUTDOWN (class A) 215
- SLEEP (class any) 216
- SAVESYS (class G) 216
- SMS (class G) 217
- SPACE (class D) 218
- SPNODE (class A) 219
- SPOOL (class G) 220
SPTAPE (class D) 221  
START (class D) 222  
STCP (class C) 225  
STORE (class G) 226  
SYSTEM (class G) 227  
TAG (class G) 227  
TERMINAL (class G) 231  
TRACE (class G) 232  
TRANSFER (class D) 232  
TRANSFER (class G) 233  
UNLOCK (class A) 235  
VARY (class B) 237  
VMDUMP (class G) 237  
WARNING (class A,B) 242

CP command (CMS) 67  
CP command (CP class any) 67  
CP command (RSCS) 68  
CP DIRECT return codes 41  
CP privilege classes 50-53  
CPQUERY command (RSCS) 68  
CPTRAP command (CP class C) 69  
creating an assembler language source file 22  
    assembling a source file 24  
    correcting errors 25  
    erasing unwanted files 28  
    testing and correcting 26  
CURSOR VSCREEN command (CMS) 69

D (CMS Border Command) 69  
DCP command (CP class C,E) 70  
DDR command (CMS) 70  
DEBUG command (CMS) 72  
    subcommands 72  
        BReak 72  
        CAW 72  
        CSW 72  
        DEFine 72  
        DUmp 72  
GO 72  
GPR 73  
HX 73  
ORigin 73  
PSW 73  
RET urn 73  
SET 73  
STore 73  
X 73  
DEFAULTS command (CMS) 73  
DEFINE command (CP class A,B) 74  
DEFINE command (CP class G) 75  
DEFINE command (RSCS) 77  
DEFINE VSCREEN command (CMS) 77  
DEFINE WINDOW command (CMS) 78  
DELETE command (RSCS) 78  
DELETE LINK command (TSAF) 78  
DELETE VSCREEN command (CMS) 78  
DELETE WINDOW command (CMS) 79  
DESBUF command (CMS) 79  
DETACH command (CP class B) 79  
DETACH command (CP class G) 79  
 developing program facilities 24  
 developing programs 22  
 DIAL command (CP class any) 80  
 DIRECT command (CMS) 80  
 DISABLE command (CP class A, B) 81  
 DISCONN command (CP class any) 81  
 DISCONN command (RSCS) 82  
 disconnect function 9  
 DISCONNECT status 9  
 disconnection of terminal 9  
 DISK command (CMS) 82  
 DISKMAP command (CMS) 82  
 DISPLAY command (CP class G) 83  
 display terminals, status action 11  
 DLBL command (CMS) 84  
 DLBL command (GCS) 85  
 DMCP command (CP class C, E) 85  
 DOSLIB command (CMS) 86  
 DOSLKED command (CMS) 86  
 DRAIN command (CP class D) 87  
 DRAIN command (RSCS) 87
DROP WINDOW command (CMS) 87
DROPBUF command (CMS) 87
DSERV command (CMS) 88
DUMP command (CP class G) 88
DUMPSCAN command (GCS) 89
DUMPSCAN command (IPCS) 90
subcommands 90-93
  + (increment) 90
  (null line) 90
  & 90
  &name 90
  - (decrement) 90
  ? 90
  Aregs 90
  ARIOBLOK 90
  C 90
  CHain 91
  CMS 91
  CMSPoint 91
  CORtable 91
  Display 91
  DOSPoint 91
  DUMPID 91
  END 91
  FORMAT 94
  G 92
  HELP 92
  HEX 94
  HX 92
  IPCSMAp 92
  Locate 92
  Locate Up 92
  MAPA 93
  MAPN 93
  Mregs 93
  MRIOBLOK 93
  OSPoint 93
  Print 93
  PRT 93
  QUIT 93
  Regs 93
  RIOblok 93
  Scroll 94
  Scrollu 94
  SYMP 94
  USERMAP 94
  VIOblok 94
  Vmblok 94

ECHO command (CP class G) 94
EDIT command (CMS) 95
subcommands and macros 95-101
  &DUP 95
  &MOVE 95
  ? 95
  ALter 95
  AUTOsave 96
  BBackward 96
  Bottom 96
  CASE 96
  Change 96
  CMS 96
  DELete 96
  DOwn 96
  DString 96
  FILE 97
  Find 97
  FMode 97
  FName 97
  FORMat 97
  FOrward 97
  Getfile 97
  IMAGE 97
  Input 98
  LINEmode 98
  Locate 98
  LONG 98
  Next 98
  nnnnnnnnnn 101
  Overlay 98
  PREserve 98
  PROMPT 98
  QUIT 98
ENABLE command (CP class A,B) 101
ERASE command (CMS) 102
erasing unwanted files 28
ESERV command (CMS) 102
ETRACE command (GCS) 102
EXEC command (CMS) 103
   built-in functions 106
      &LITERAL 106
      &variable = &CONCAT 106
      &variable = &DATATYPE 106
      &variable = &LENGTH 106
      &variable = &SUBSTR 106
control statements 103
      &ARGS 103
      &BEGEMSG 103
      &BEGPUNCH 103
      &BEGSTACK 103
      &BEGTYPE 103
      &CONTINUE 103
      &CONTROL 104
      &EMSG 104
      &END 104
      &ERROR 104
      &EXIT 104
      &GOTO 104
      &HEX 104
      &IF 104
      &LOOP 104
      &PUNCH 105
      &READ 105
      &SKIP 105
      &SPACE 105
      &STACK 105
      &TIME 105
      &TYPE 105
      &variable 103
special variables 106
      &* and &$ 106
      &0 106
      &DISK* 106
      &DISK? 106
      &DISKx 106
      &DOS 106
      &EXEC 106
      &GLOBAL 107
      &GLOBALn 107
      &INDEX 107
      &LINENUM 107
      &n 106
      &READFLAG 107
EXEC command (RSCS) 107
EXEC 2 command (CMS) 108-117
control statements 110
      &ARGS 110
      &BEGPRINT 110
      &BEGSTACK 110
      &BEGTYPE 110
      &BUFFER 111
      &CALL 111
      &CASE 111
      &COMMAND 111
      &CRASH 111
      &DUMP 111
      &ERROR 112
      &EXIT 112
      &GOTO 112
&IF 112
&LOOP 113
&PRESUME 113
&PRINT 113
&READ 113
&RETURN 114
&SKIP 114
&STACK 114
&SUBCOMMAND 114
&TRACE 114
&TRUNC 115
&type 113
&upper 115

predefined functions 115
&CONCAT 115
&CONCATENATION 115
&DATATYPE 115
&DIV 115
&DIVISION 115
&LEFT OF 116
&LENGTH OF 116
&LITERAL OF 116
&LOCATION OF 116
&MULT 116
&MULTIPLICATION OF 116
&PICE 116
&POSITION OF 116
&RANGE OF 117
&RIGHT OF 117
&STRING OF 117
&SUBSTR 116
&TRANS 117
&TRANSLATION 117
&TRIM OF 117
&type 115
&WORD OF 117

predefined variables 108
&0 108
&ARGSTRING 108
&BLANK 108
&CMDDSTRING 108
&COMLINE 108
&DATE 108
&DEPARTH 109
&FILENAME 109
&FILETYPE 109
&FROM 109
&LINE.&LINENUM 109
&LINK 109
&N.&INDEX 109
&RC.&RETCODE 109
&TIME 109
&1,&2,&n 108

user-defined functions 117
label 117
line-number OF 117

EXEC 2 Interpreter 17
EXECDROP command (CMS) 118
EXECIO command (CMS) 119
EXECLOAD command (CMS) 120
EXECMAP command (CMS) 120
EXECOS command (CMS) 121
EXECUPDT command (CMS) 121
EXESTAT command (CMS) 121
EXPAND command (CMS) 122
EXTERNAL command (CP class G) 122

F (CMS Border Command) 122
FETCH command (CMS) 123
FILEDEF command (CMS) 124
FILEDEF command (GCS) 126
FILELIST command (CMS) 126
FINIS command (CMS) 127
FLUSH command (CP class D) 127
FLUSH command (RSCS) 127
FORCE command (CP class A) 127
FORCE command (RSCS) 128
FORMAT command (CMS) 128
FORMAT/ALLOCATE command (service aid) 129
FREE command (CP class D) 129
FREE command (RSCS) 129
FWDSPAC command (RSCS) 130

G

GCS (Group Control System) Ccommands
* 53
ACCESS 54
DBL 85
DUMPSCAN 89
ETRACE 102
FILEDEF 126
GDUMP 130
GLOBAL 132
HX 137
ITRACE (GCS) 142
LOADCMD 146
OSRUN 160
QUERY 177
RELEASE 181
REPLY 182
SET 214
GDUMP command (GCS) 130
GENDIR command (CMS) 130
GENIMAGE command (CMS) 130
GENMOD command (CMS) 131
GENMSG command (CMS) 131
GEN3705 command (CMS) 131
GET VSCREEN command (CMS) 132
GLOBAL command (CMS) 132
GLOBAL command (GCS) 132
GLOBALV command (CMS) 133

Group Control System (GCS)
See GCS (Group Control System) Ccommands
GTRACE command (RSCS) 134

H

H (CMS Border Command) 134
HALT command (CP class A) 134
HB command (CMS) 134
HELP command (CMS) 135
HELPCONV command (CMS) 135
HL command (CMS) 135
HIDE WINDOW command (CMS) 136
HO command (CMS) 136
HOLD command (CP class D) 136
HOLD command (RSCS) 136
how to IPL CMS 16
HT command (CMS) 136
HX command (CMS) 137
HX command (GCS) 137

I

IDENTIFY command (CMS) 137
IMAGELIB command (CMS) 137
IMAGEMOD command (CMS) 138
IMMCMD command (CMS) 138
INCLUDE command (CMS) 139
INDICATE command (CP class A) 139
INDICATE command (CP class E) 139
INDICATE command (CP class G) 140
INIT command (RSCS) 140
input conventions 10
Interactive Problem Control System
See IPCS commands

IPCS commands
APAR 56
CONVERT 66
CONVIPCS EXEC 66
DUMPSCAN 90
+ (increment) 90
(null line) 90
& 90
&name 90
- (decrement) 90
? 90
Aregs 90
ARIOBLOK 90
C 90
CHain 91
CMS 91
CMSPoint 91
CORTable 91
Display 91
DOSPPoint 91
DUMPID 91
END 91
FORMat 94
G 92
HELP 92
HEX 94
HX 92
IPCSMAP 92
Locate 92
Locate Up 92
MAPA 93
Mregs 93
MRIOBLOK 93
OSPoint 93
Print 93
PRT 93
QUIT 93
Regs 93
RIOblok 93
Scroll 94
Scrollu 94
SYMP 94
USERMAP 94
VIOblok 94
Vmblok 94
IPCSDUMP 140
MAP 149
PRB 163
PROB 164
PRTDUMP 167
STAT 224
PCIS commands return codes 45
IPCSDUMP command (IPCS) 140
IPL command (CP class G) 140
ITASK command (CMS) 141
ITRACE command (GCS) 142
L (CMS Border Command) 142
LABELDEF command (CMS) 142
LANGGEN command (CMS) 143
LANGMERG command (CMS) 143
line length 11
line termination 11
LINK command (CP class G) 143
LISTDS command (CMS) 143
LISTFILE command (CMS) 144
LISTIO command (CMS) 144
LKED command (CMS) 145
LOAD command (CMS) 145
LOADBUF command (CP class D) 146
LOADCMD command (GCS) 146
LOADLIB command (CMS) 146
LOADMOD command (CMS) 146
LOADVFCB command (CP class G) 147
LOCATE command (CP class C,E) 147
LOCK command (CP class A) 147
logging off 8
logging on 3
logical line-editing characters 10
logoff and security 9
LOGOFF command (CP class any) 147
LOGON command (CP class any) 147
M (CMS Border Command) 148
MACLIB command (CMS) 148
MAKEBUF command (CMS) 149
MAP command (IPCS) 149
MAXIMIZE WINDOW command (CMS) 149
MESSAGE command (CP class A,B) 149
MESSAGE command (CP class Any) 149
MIGRATE command (CP class A) 150
MINIMIZE WINDOW command (CMS) 150
MODMAP command (CMS) 150
MONITOR command (CP class A,E) 150
MOREHELP command (CMS) 152
MOVEFILE command (CMS) 152
MSG command (RSCS) 153
MSGNOH command (CP class A,B) 153

N (CMS Border Command) 153
NAMEFIND command (CMS) 153
 NAMES command (CMS) 154
NCPDUMP command (CMS) 154
NETWORK command (CP class A) 155
NETWORK command (CP class B) 156
NETWORK command (RSCS) 157
notational conventions 48
notational usage 48
NOTE command (CMS) 157
NOTREADY command (CP class G) 157
NUCXDROP command (CMS) 157
NUCXLOAD command (CMS) 158
NUCXMAP command (CMS) 158

O (CMS Border Command) 158
OPTION command (CMS) 158
ORDER command (CP class D) 159
ORDER command (CP class G) 159
ORDER command (RSCS) 159
OSRUN command (CMS) 159
OSRUN command (GCS) 160

P (CMS Border Command) 160
PARSECMD command (CMS) 160
PEEK command (CMS) 161
PER command (CP class A,B,C,D,E,F,G) 161
POP WINDOW command (CMS) 163
POSITION WINDOW command (CMS) 163
PRB command (IPCS) 163
PRELOAD command (CMS) 164
PRINT command (CMS) 164
printing files 29
printing, punching and reading files 29
privilege classes 50
PROB command (IPCS) 164
program development facilities 22
PROGRAMMABLE OPERATOR (CMS) 165
commands 165-166
CMD 165
FB 166
FEEDBACK 166
GET 166
LGLOPR 166
LOADTBL 166
LOG 166
QUERY HOSTCHK 166
QUERY LGLOPR 166
QUERY LOGGING 166
QUERY PROPCHK 166
QUERY RTABLE 166
SET DEBUG 167
SET HOSTCHK 167
SET LOGGING 167
SET PROPCHK 167
PRTDUMP command (IPCS) 167
PSERV command (CMS) 168
PUNCH command (CMS) 168
punching files 29
PURGE command (CP class D) 168
PURGE command (CP class G) 169
PURGE command (RSCS) 169
PUT SCREEN command (CMS) 169
PUT VSCREEN command (CMS) 169

RECEIVE command (CMS) 180
RECONN command (RSCS) 181
REFRESH command (CMS) 181
RELEASE command (CMS) 181
RELEASE command (GCS) 181
Remote Spooling Communications Subsystem (RSCS)
   See RSCS (Remote Spooling Communications Subsystem) commands
RENAME command (CMS) 182
REPEAT command (CP class D) 182
REPLY command (GCS) 182
REQUEST command (CP class G) 182
REROUTE command (RSCS) 183
RESERVE command (CMS) 183
reserved filetype descriptions 34
RESET command (CP class G) 183
RESTORE WINDOW command (CMS) 183
return codes 39
   CMS DDR command return codes 42
   CMS return codes 39
   CP DIRECT return codes 41
   IPCS commands return codes 45
   System Product Editor command return codes 42
   XEDIT command return codes 42
REWIND command (CP class G) 184
REXX language (CMS) 184
REXX built-in functions 190-193
   ABBREV 190
   ABS 190
   Address 190
   Arg 190
   BITAND 190
   BITOR 190
   BIXOR 190
   CENTER 190
   CENTRE 190
   COMPARE 190
   COPIES 190
   C2D 191
   C2X 191
   Datatype 191
   DATE 191

Index 293
DELSTR 191
DELWORD 191
D2C 191
D2X 191
ERRORTEXT 191
EXTERNALS 191
FIND 191
FORMAT 191
INDEX 192
INSERT 192
JUSTIFY 192
LASTPOS 192
LEFT 192
LENGTH 192
LINESIZE 192
MAX 192
MIN 192
OVERLAY 192
POS 192
QUEUED 192
RANDOM 192
REVERSE 192
RIGHT 193
SIGN 193
SOURCeline 193
SPACE 193
STRIP 193
SUBSTR 193
SUBWORD 193
SYMBOL 193
TIME 193
TRACE 193
TRANSLATE 193
TRUNC 193
USERID 193
VALUE 193
VERIFY 194
WORD 194
WORDINDEX 194
WORDLENGTH 194
WORDS 194
XRANGE 194
X2C 194
X2D 194

REXX instructions 184-188
ADDRESS 184
ARG 184
CALL 184
DO 185
DROP 185
EXIT 185
IF-THEN-ELSE 186
INTERPRET 186
ITERATE 186
LEAVE 186
NOP 186
NUMERIC DIGITS 186
NUMERIC FORM 186
NUMERIC FUZZ 186
PARSE 187
PROCEDURE 187
PULL 187
PUSH 187
QUEUE 187
RETURN 187
SAY 187
SELECT 188
SIGNAL 188
TRACE 189
UPPER 189

RSXYSFN package 195
CMSFLAG 195
DIAG 195
DIAGRC 195
STORAGE 195

RO command (CMS) 195
ROUTE command (CMS) 195
ROUTE command (RSCS) 196
RSCS (Remote Spooling Communications Subsystem) commands
• 53
  BACKSPAC 59
  CHANGE 63
  CMD 64
  CP 68
  CPQUERY 68
  DEFINE 77
DELETE 78
DISCONN 82
DRAIN 87
EXEC 107
FLUSH 127
FORCE 128
FREE 129
FWDSpace 130
GTRACE 134
HOLD 136
INIT 140
MSG 153
NETWORK 157
ORDER 159
PURGE 169
QUERY 178
READY 180
RECONN 181
REROUTE 183
ROUTE 196
SET 214
SHUTDOWN 216
START 223
STOP 225
TRACE 232
TRANSFER 233
RSERV command (CMS) 196
RT command (CMS) 196
RUN command (CMS) 196
RUNTSaf command (TSAF) 197

S

S (CMS Border Command) 197
SAVENCP command (CMS) 197
SAVESYS command (CP class E) 197
SCREEN command (CP class G) 198
SCROLL command (CMS) 199
security
security and logoff 9
SEND command (CP class G) 199
SENDFILE command (CMS) 200

sending files 30
SENTRIES command (CMS) 200
service aid
    FORMAT/ALLOCATE 129
SET ABBREV command (CMS) 201
SET APL command (CMS) 201
SET AUTOREAD command (CMS) 201
SET BLIP command (CMS) 201
SET BORDER command (CMS) 202
SET CHARMODE command (CMS) 202
SET CMSPF command (CMS) 202
SET CMSTYPE command (CMS) 203
SET command (CMS) 200
SET command (CP class A) 211
SET command (CP class B) 211
SET command (CP class E) 212
SET command (CP class F) 212
SET command (CP class G) 213
SET command (RSCS) 214
SET command (SET) 214
SET DOS command (CMS) 203
SET DOSLNCNT command (CMS) 203
SET DOSPART command (CMS) 203
SET ETRACE command (TSAF) 214
SET EXECTRAC command (CMS) 203
SET FULLREAD command (CMS) 204
SET FULLSCREEN command (CMS) 204
SET IMESCAPE command (CMS) 204
SET IMPCP command (CMS) 204
SET IMPEX command (CMS) 205
SET INPUT command (CMS) 205
SET INSTSEG command (CMS) 205
SET LANGUAGE command (CMS) 205
SET LDRTBLS command (CMS) 206
SET LINEND command (CMS) 206
SET LOCATION command (CMS) 206
SET LOGFILE command (CMS) 206
SET NONDISP command (CMS) 206
SET NONSHARE command (CMS) 207
SET OUTPUT command (CMS) 207
SET PROTECT command (CMS) 207
SET Rdymsg command (CMS) 207
SET REDTYPE command (CMS) 208
SET RELPAGE command (CMS) 208
SET REMOTE command (CMS) 208
SET RESERVED command (CMS) 208
SET SYSNAME command (CMS) 209
SET TEXT command (CMS) 209
SET TRANSLATE command (CMS) 209
SET UPSI command (CMS) 209
SET VSCREEN command (CMS) 210
SET WINDOW command (CMS) 210
SET WMPF command (CMS) 210
SETKEY command (CMS) 215
SETPRT command (CMS) 215
SHOW WINDOW command (CMS) 215
SHUTDOWN command (CP class A) 215
SHUTDOWN command (RSCS) 216
SIZE WINDOW command (CMS) 216
SLEEP command (CP class any) 216
SMSG command (CP class G) 216
SMSG command (RSCS) 217
SNTMAP command (CMS) 217
SO command (CMS) 217
SORT command (CMS) 217
SPACE command (CP class D) 218
SPGEN command (CMS) 218
SPLOAD command (CMS) 219
SPMODE command (CP class A) 219
SPOOL command (CP class G) 220
SPTAPE command (CP class D) 221
SSERV command (CMS) 222
START command (CMS) 222
START command (CP class D) 222
START command (RSCS) 223
STAT command (IPCS) 224
STATE/STATEW command (CMS) 224
status action on display terminals 11
STCP command (CP class C) 225
STOP command (RSCS) 225
STOP TSAF command (TSAF) 225
STORE command (CP class G) 226
summary of status action on display terminals 11
SVCTRACE command (CMS) 226
SYNONYM command (CMS) 226
SYSTEM command (CP class G) 227

System Product Editor 16
System Product Editor command return codes 42
System Product Interpreter 18

T

TAG command (CP class G) 227
TAPE command (CMS) 228
TAPEMAC command (CMS) 229
TAPPDS command (CMS) 229
TE command (CMS) 230
TELL command (CMS) 230
TERMINAL command (CP class G) 231
terminal operating procedures
disconnecting your terminal 9
holding line 8
logging off 8
logging on 3
testing and correcting a program 26
TRACE command (CP class G) 232
TRACE command (RSCS) 232
TRANSFER command (CP class D) 232
TRANSFER command (CP class G) 233
TRANSFER command (RSCS) 233
Transparent Services Access Facility (TSAF)
See TSAF (Transparent Services Access Facility) commands
TRAPRED command (CMS) 234
TS command (CMS) 234
TSAF (Transparent Services Access Facility) commands
ADD LINK 55
DELETE LINK 78
QUERY 178
RUNTSAF 197
SET ETRACE 214
STOP TSAF 225
TXTLIB command (CMS) 235
TYPE command (CMS) 235
UNLOCK command (CP class A) 235
UPDATE command (CMS) 236
using CMS 38
CMS file system 33
CMS fileids 32
example of CMS program development facilities 22-31
assembling a source file 24
correcting errors 25
creating a load module 26
creating an assembler language source file 22
erasing unwanted files 28
printing, punching and reading files 29
testing and correcting a program 26
EXEC 2 Interpreter 17
how to IPL CMS 16
notational conventions 48
reserved filetype descriptions 34
return codes 39-46
System Product Editor 16
System Product Interpreter 18
using the VM/SP system
input conventions 10
line length 11
line termination 11
logical line-editing characters 10
virtual machine configuration 1
UTILITY command (CMS) 236

VALIDATE command (CMS) 237
VARY command (CP class B) 237
virtual machine configuration 1
VMDUMP command (CP class G) 237
VMFASM command (CMS) 238
VMFOS command (CMS) 238
VMFLKED command (CMS) 239
VMFLOAD command (CMS) 239
VMFMAC command (CMS) 239
VMFMERGE command (CMS) 239
VMFNLS command (CMS) 240
VMFPLC2 command (CMS) 240
VMFREMOC command (CMS) 241
VMFTXT command (CMS) 241
VMFZAP command (CMS) 241
WAITREAD VSCREEN command (CMS) 242
WAITT VSCREEN command (CMS) 242
WARNING command (CP class A,B) 242
WRITE VSCREEN command (CMS) 243
X (CMS Border Command) 243
XEDIT command (CMS) 244-270
prefix subcommands 268
.xxx 269
< 269
/[n] 269
> 269
" 270
A 268
C 268
D 268
E 268
F 268
I 268
M 268
P 268
S 269
SCALE 269
SI 269
<table>
<thead>
<tr>
<th>subcommands and macros</th>
<th>244</th>
</tr>
</thead>
<tbody>
<tr>
<td>&amp;</td>
<td>244</td>
</tr>
<tr>
<td>?</td>
<td>244</td>
</tr>
<tr>
<td>=</td>
<td>244</td>
</tr>
<tr>
<td>Add</td>
<td>244</td>
</tr>
<tr>
<td>ALL</td>
<td>244</td>
</tr>
<tr>
<td>ALT</td>
<td>245</td>
</tr>
<tr>
<td>Backward</td>
<td>245</td>
</tr>
<tr>
<td>Bottom</td>
<td>245</td>
</tr>
<tr>
<td>CANcel</td>
<td>245</td>
</tr>
<tr>
<td>CAppend</td>
<td>245</td>
</tr>
<tr>
<td>CDelete</td>
<td>245</td>
</tr>
<tr>
<td>CFist</td>
<td>245</td>
</tr>
<tr>
<td>Change</td>
<td>245</td>
</tr>
<tr>
<td>CInsert</td>
<td>246</td>
</tr>
<tr>
<td>CLast</td>
<td>246</td>
</tr>
<tr>
<td>CLocate</td>
<td>246</td>
</tr>
<tr>
<td>CMS</td>
<td>246</td>
</tr>
<tr>
<td>CMSG</td>
<td>246</td>
</tr>
<tr>
<td>COMMAND</td>
<td>246</td>
</tr>
<tr>
<td>COMPpress</td>
<td>246</td>
</tr>
<tr>
<td>COpy</td>
<td>246</td>
</tr>
<tr>
<td>COUnt</td>
<td>246</td>
</tr>
<tr>
<td>COVerlay</td>
<td>246</td>
</tr>
<tr>
<td>CP</td>
<td>246</td>
</tr>
<tr>
<td>CREplace</td>
<td>246</td>
</tr>
<tr>
<td>CURsor</td>
<td>247</td>
</tr>
<tr>
<td>DELete</td>
<td>247</td>
</tr>
<tr>
<td>Down</td>
<td>247</td>
</tr>
<tr>
<td>DUPlicate</td>
<td>247</td>
</tr>
<tr>
<td>EMSG</td>
<td>247</td>
</tr>
<tr>
<td>EXPand</td>
<td>247</td>
</tr>
<tr>
<td>EXTract</td>
<td>248</td>
</tr>
<tr>
<td>FILE</td>
<td>249</td>
</tr>
<tr>
<td>Find</td>
<td>249</td>
</tr>
<tr>
<td>FINDUp</td>
<td>249</td>
</tr>
<tr>
<td>FOrward</td>
<td>249</td>
</tr>
<tr>
<td>FUp</td>
<td>249</td>
</tr>
<tr>
<td>GET</td>
<td>249</td>
</tr>
<tr>
<td>Help</td>
<td>249</td>
</tr>
<tr>
<td>HEXType</td>
<td>249</td>
</tr>
<tr>
<td>Input</td>
<td>249</td>
</tr>
</tbody>
</table>
STAck 266
STATus 266
TOP 266
TRAnsfer 267
Type 267
Up 267
UPPercas 267
Xedit 267

XEDIT command return codes 42
XMITMSG command (CMS) 270

Z

ZAP command (CMS) 271
Quick Reference
Order No. SX20-4400-4

Is there anything you especially like or dislike about this book? Feel free to comment on specific errors or omissions, accuracy, organization, or completeness of this book.

IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you, and all such information will be considered nonconfidential.

Note: Do not use this form to report system problems or to request copies of publications. Instead, contact your IBM representative or the IBM branch office serving you.

Would you like a reply? ___ YES ___ NO

Please print your name, company name, and address:

____________________________________________________________________________
____________________________________________________________________________
____________________________________________________________________________
____________________________________________________________________________
IBM Branch Office serving you: __________________________________________________

Thank you for your cooperation. You can either mail this form directly to us or give this form to an IBM representative who will forward it to us.