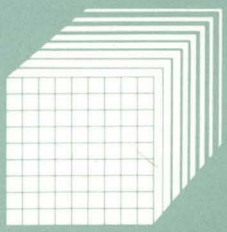
IBM

Personal Computer
Professional Series

IBM Virtual Machine/ Personal Computer User's Guide



6361530

Preface

This publication is intended for all users of VM/PC. It assumes a general knowledge of the VM operating environment. This publication has two sections:

1. Chapters 1 through 5 contain basic information to help you start using VM/PC.

2. Chapters 6 through 8 and the Appendixes contain more specialized and reference information.

"How to Use This Book" on page 1-3 describes the layout of this book in greater detail.

If you are not familiar with using VM/SP or something similar, you may want to order the IBM publication *Virtual Machine/Personal Computer Primer,* part number 6024174, from your IBM Marketing Representative. This Primer is intended for beginning users. It provides many examples of basic usage.

VM/PC is based upon VM/SP Release 2. It may be necessary to reference some VM/SP publications to find further reference information for VM/PC. Appendix C, "Comparison of VM/PC and VM/SP" on page C-1 provides information about where to find further information in the VM/SP library, and gives the Titles and Order Numbers for the VM/SP books.

When you see text printed in this color it indicates either:

1. This is the first occurrence of a term defined in the glossary.

2. This is a user entry (for example, a command).

Statement of Service

Program service for VM/PC is IBM Central Service including an IBM Support Center. This service will be available until discontinued by IBM upon six (6) months written notice or until 31 December 1985, whichever comes first.

The IBM Support Center will provide the customer technical coordinator with telephone assistance in problem diagnosis and resolution. The IBM Support Center is the established point of contact between the customer technical coordinator and IBM Central Service.

IBM Central Service will respond to a defect in the unaltered portion of a supported release of the licensed program by issuing defect correction information such as correction documentation, corrected code, or notice of availability of corrected code; or a restriction or a bypass. IBM does not guarantee service results or that the program will be error free or that all program defects will be corrected.

No on-site service is included.

The IBM Marketing Representative will register the customer technical coordinator with the IBM Support Center. It is expected that the customer technical coordinator is familiar with VM/CMS. The customer technical coordinator becomes the first level of customer end-user assistance. The technical coordinator's duties should include:

- answer end-user hardware and software questions.
- perform problem determination to determine if a problem is hardware or software related.
- provide the customer interface to the appropriate IBM Support Centers.
- provide corrections received from IBM Central Service through the IBM Support Center to all licensed VM/PC users for whom he is responsible in the customer organization.
- manage problems and changes for VM/PC end-users in the customer's organization.

Some customers may find it necessary to have more than one technical coordinator. However, to ensure consistency of support, the number of technical coordinators should be kept to a minimum. The IBM Marketing Representative is the contact to negotiate additional technical coordinators.

Contents

Chapter 1. Introduction to VM/PC ... 1-1

What You Need 1-2
How to Use This Book 1-3
The VM/PC System 1-7
Using Local VM/PC Services 1-8
Using Remote VM Services 1-9

Chapter 2. Installing VM/PC 2-1

About This Chapter 2-1
Before You Begin 2-2
Copying VM/PC to Backup Diskettes 2-4
Copying VM/PC to Your System 2-5
How Much Disk Space Is Required? 2-14
At Installation Time 2-14
After Installation 2-14
What About Existing Files? 2-17
If Something Goes Wrong? 2-19
DOS Facilities Missing 2-19
VM/PC Temporary Files Already Exist 2-19
Insufficient Space on C Disk 2-20
Unable to Read Space Available on C Disk 2-21
VM/PC Distribution Diskette Not Used 2-21
Repeating the Installation Process 2-23

Chapter 3. Configuring Your VM/PC System 3-1

The Configuration File 3-1
The Configurator Program 3-3
Starting the Configurator Program 3-3
Using the Configurator 3-3
Configurator Screens 3-4
Function Key Usage 3-4
Keyboard Usage 3-5
The Password Screen 3-6
Function Menu Screen 3-6
Quitting the Configurator 3-7

Normal Exit from the Configurator 3-7
System Configuration Screen 3-8
Configurator Password Screen 3-10
System Options Screen 3-10
User Selection Screen 3-11
User Function Selection Screen 3-12
Update Environment for User Screen 3-12
Minidisk List Screen 3-13
Link List Screen 3-16
Update Link Screen 3-17
Disk Allocation and Checking 3-18
Limitations 3-19

Chapter 4. Getting Started 4-1
Invoking VM/PC 4-1
Keyboard Usage 4-2
Session Selection 4-5
Local 3277 Session 4-6
Remote 3277 Session 4-6
Remote 3101 Session 4-7
Prerequisites 4-8
Using the Remote 3101 Session 4-8
370 Processor Control 4-11
Logon Sequence 4-12
Screen Format 4-15
Screen Scroll 4-15
Status Notices 4-16
The 25th Line 4-18
What To Do if Something Goes Wrong 4-21
System Failures 4-21
Startup Failures 4-22
Processing Failures 4-23
System Error Conditions 4-23
Error Messages 4-23
System Unavailable 4-23

Chapter 5. Using VM/PC Commands to Do Work 5-1
Using Commands to Do Work 5-1
Creating, Changing and Managing Files 5-2

Creating Files 5-2
What Is a File? 5-2
Creating A File 5-4
Entering Data 5-7
Capital and Lowercase Letters 5-8
Splitting and Joining Lines 5-9
Setting Tabs 5-10
Displaying PF Key Settings 5-10
Inserting Words in a Line 5-11
Power Typing 5-11
Creating an XEDIT Profile 5-12
Changing Files 5-13
Displaying a File on Your Screen 5-13
Adding Lines 5-13
Moving through a File 5-14
Deleting Lines 5-15
Duplicating Lines 5-17
Copying Lines 5-17
Moving Lines 5-19
Cancelling a Prefix Command - RESET Command 5-19
LOCATE and CHANGE Commands 5-19
Repeating a Command - The = (equal) Command 5-21
Combining Files 5-21
Cancelling Changes to a File 5-22
Numbering Lines of a File 5-23
Sorting a File 5-23
Protecting Files 5-26
SAVE Command 5-26
SET AUTOSAVE Command 5-27
Managing Files 5-27
Getting a List of Your Files 5-28
Using the FILELIST Command 5-28
Using the LISTFILE Command 5-35
Printing Files 5-38
PRINT Command 5-38
Remote Printing 5-38
Writing EXECs 5-40
An EXEC to Print a File 5-40

Setting Up Your System to Meet Your Needs ... 5-44
Finding Out Basic Information 5-44
How Much Disk Space Have I Used? 5-44
How Does CMS Choose Programs and Files? . 5-45
Modifying the ACCESS List 5-45
An Easy Way to Set Up a Concurrent Session . 5-47
Establishing Automatic LINKs and ACCESSes. 5-48
Sharing Data and Programs with Other Users .. 5-48
A Sample PROFILE EXEC 5-50
PF Keys Make Work Simpler 5-50
Reviewing What You Have Done 5-51
Scrolling Screen Contents 5-51
Retrieving Commands 5-52
Using Synonyms in Your Work 5-53
Controlling Your Display 5-54
Highlighting Your Input 5-54
Scrolling a Half Screen 5-54
Displaying Long Lines 5-54
Transferring Personal Computer DOS and VM/PC Files 5-55
Copying a Text File from Personal Computer DOS to VM/PC (IMPORT) 5-55
Copying a Text File from VM/PC to Personal Computer DOS (EXPORT) 5-56
Copying a Program from Personal Computer DOS to VM/PC (IMPORT) 5-57
Copying a Program from VM/PC to Personal Computer DOS (EXPORT) 5-57

Chapter 6. Remote Services 6-1
Remote Upload/Download 6-1
VM/PC Remote Services Program 6-3
Installing VMPCSERV 6-3
What To Do if Something Goes Wrong 6-6
VMPCSERV Program 6-8
Remote Minidisks 6-9
Remote Files 6-12
File Upload/Download 6-13
Remote Spooling 6-14
What to Do if Something Goes Wrong 6-15

How VMPCSERV Works 6-16
Collecting Names of Files Downloaded to VM/PC 6-18

Chapter 7. VM/PC Commands 7-1
Entering Commands 7-1
The Command Name 7-1
The Command Operands 7-1
The Command Options 7-2
Comments in CMS Command Lines 7-2
Notational Conventions 7-2
CP Commands 7-7
Format of CP Commands 7-7
Summary of CP Commands 7-7
CP Commands and Options 7-10
* 7-10
#CP 7-10
ATTN 7-11
BEGIN 7-12
CHANGE 7-12
CLOSE 7-14
CP 7-16
DEFINE 7-16
DETACH 7-18
DISPLAY 7-19
DUMP 7-23
EXTERNAL 7-26
IPL 7-26
LINK 7-27
LOGOFF 7-30
ORDER 7-30
PURGE 7-31
QUERY 7-32
REQUEST 7-37
SET 7-38
SPOOL 7-41
STORE 7-44
TAG 7-46
TERMINAL 7-48
TRACE 7-51

CMS Commands 7-56
Character Set Usage 7-56
CMS Command Search Order 7-56
Summary of CMS Commands 7-59
CMS Commands and Options 7-61
ACCESS 7-62
CMDCALL 7-66
COMPARE 7-66
CONWAIT 7-66
COPYFILE 7-67
CP 7-67
DEBUG 7-68
DEFAULTS 7-69
DESBUF 7-71
DROPBUF 7-71
ERASE 7-72
ESTATE/ESTATEW 7-72
EXEC 7-74
EXECIO 7-74
EXPORT 7-76
FILEDEF 7-78
FILELIST 7-78
FINIS 7-79
FORMAT 7-80
GENDIRT 7-84
GENMOD 7-84
GLOBAL 7-89
GLOBALV 7-90
IDENTIFY 7-93
IMPORT 7-94
INCLUDE 7-97
LISTFILE 7-98
LOAD 7-99
LOADMOD 7-99
MACLIB 7-100
MAKEBUF 7-101
MODMAP 7-102
NUCXDROP 7-102
NUCXLOAD 7-102
NUCXMAP 7-104

PRINT 7-104
QUERY 7-105
RELEASE 7-114
RENAME 7-114
SENTRIES 7-114
SET 7-115
SORT 7-118
START 7-119
STATE/STATEW 7-119
SYNONYM 7-119
TXTLIB 7-126
TYPE 7-126
UPDATE 7-126
XEDIT 7-127
Immediate Commands 7-128

Chapter 8. VM/PC Messages 8-1
How To Use This Chapter 8-1
Identifying Issuing Component 8-1
Reading a Sample Message Explanation 8-2
Message Explanations 8-3
List of All VM/PC CMS Messages 8-26

Appendix A. CP/CMS Interface A-1
Privileged Operation Simulation A-1
Set Storage Key - SSK A-1
Insert Storage Key - ISK A-1
Test I/O - TIO A-2
Set Clock - SCK A-2
Diagnose Instruction in VM/PC A-3
DIAGNOSE Code X'00' -- Store Extended-Identification Code A-5
DIAGNOSE Code X'08' -- Virtual Console Function A-7
DIAGNOSE Code X'0C' -- Pseudo Timer A-10
DIAGNOSE Code X'10' -- Release Pages A-11
DIAGNOSE Code X'18' -- Standard DASD I/O A-12
DIAGNOSE Code X'20' -- General I/O A-14
DIAGNOSE Code X'24' -- Device Type and Features A-16

DIAGNOSE Code X'54' -- Control The Function of the PA2 Key A-19
DIAGNOSE Code X'58' -- 3270 Virtual Console Interface A-20
Displaying Data A-20
Full Screen Mode A-22
Full Screen Interactions A-24
DIAGNOSE Code X'5C' -- Error Message Editing A-27
DIAGNOSE Code X'60' - Determining the Virtual Machine Storage Size A-28
DIAGNOSE Code X'64' - Finding, Loading, and Purging a Named Segment A-29

Appendix B. The 370 Processor Control Session B-1
370 Processor Control - Screen Format B-1
F2 Key - Start/Stop B-3
Processor Status Field B-3
F1 Key - Instruction Step B-4
Editing 370 Registers B-4
F3 Key - Edit (GPR/FPR/CR) B-5
Storage Edit B-5
F7 and F8 Keys - Page Forward and Backward . B-6
F6 Key - EBCDIC (On/Off) B-6
F5 Key - Edit (Real/Virtual) B-7
F4 Key - Edit PAT/Edit Main B-7
Address Compare Stop Function B-7
ALT F6 Keys - Program Reset B-8
ALT F7 Keys - Clear Reset B-8
ALT F8 Keys - External Interrupt B-8

Appendix C. Comparison of VM/PC and VM/SP C-1
Relationship of VM/PC and VM/SP C-1
Moving Application Programs from VM/SP to VM/PC C-6
VM/SP Commands Not Supported in VM/PC ... C-7
Finding Further Information in VM/SP Library ... C-8
Order Numbers for VM/SP Release 2 Books C-9

Appendix D. Operating Hints D-1
Backing up Minidisks D-1
Minidisks on Diskettes D-1
Transporting CMS Files D-1
EXPORT/IMPORT Method D-1
Minidisk Method D-2
File Access Performance D-2
Using DOS Features D-3
FORMAT Command Performance D-3
Remote Minidisks D-4
Spool Files D-4

Appendix E. ASCII to EBCDIC Conversion E-1

Appendix F. Distribution Diskette Contents F-1

Glossary X-1

Index X-5

Chapter 1. Introduction to VM/PC

VM/PC is an IBM licensed program product which runs on the IBM Personal Computer XT/370. VM/PC runs as an IBM Personal Computer Disk Operating System application and presents you with an interactive system which has the characteristics of a VM/SP Release 2 system. These characteristics include command entry, command format, messages, screen formats, data file naming conventions, key functions, and application interfaces.

VM/PC allows you:

- To set up your system quickly via a series of easy to use panels which guide you through the installation and configuration process.
- To establish a local 370 environment in your IBM Personal Computer XT/370.
- To manage a single virtual memory space of up to 4 million characters.
- To select the type of connection (3277 or 3101) to a remote computer that meets your processing needs.
- To maintain concurrent local and remote sessions over the 3277-2 Coaxial Connection or EIA RS232-C communication facilities (hereafter referred to as RS232).
- To transfer data and programs between your remote VM computer and your IBM Personal Computer XT/370.
- To execute many 370 CMS applications in your IBM Personal Computer XT/370 without modification.
- To transfer your VM/PC data files between the VM/PC format and IBM Personal Computer Disk Operating System 2.00, or equivalent, format (hereafter referred to as DOS).

What You Need

The minimum requirements of VM/PC are:

- IBM Personal Computer XT/370
- The IBM Personal Computer Disk Operating System version 2.00 (6024061) reference manual and diskettes (or equivalent)
- An IBM Monochrome Display or any other monitor designed to display 25 lines of 80 columns each
- One diskette drive
- One fixed disk.

VM/PC also supports the following:

- An IBM Matrix Printer or any compatible printer
- A second fixed disk drive, if more fixed disk storage is required
- A second diskette drive, if more diskette storage is required
- An IBM Asynchronous Communication Adapter and the IBM 3101 Emulation Program (6024042), if asynchronous communications is desired
- An IBM 3277-2 Device Emulation Adapter, if 3277-2 Coaxial Connection communications is desired.

How to Use This Book

We have assumed that you have a general knowledge of the VM operating environment. Take the time to read this manual to gain a more detailed understanding of the VM/PC system and its relationship to the remote VM system and to the IBM Personal Computer XT/370 on which it resides.

Note: If you are a beginner at system usage, you may want to start by reading the *Virtual Machine/Personal Computer Primer,* part number 6024174. You may order this publication through your IBM Marketing Representative.

All users should read these parts of this book (detailed descriptions of these parts follow):

- Introduction
- Configuring
- Getting Started
- Using Commands to Do Work
- Commands (for reference)
- Messages (for reference)
- Glossary (for meanings of terms and acronyms)
- Index (to help find information).

All users who have to do the specific tasks named should read these parts:

- Installing
- Remote Services (if installing Remote Server Program - VMPCSERV).

Users with the specific needs named should read these parts:

- Remote Services (those needing more information on Remote/Local sessions)
- CP/CMS Interface (those interested in Internal Processing)
- 370 Processor Control Session (those maintaining the Control Program)
- Comparison of VM/SP and VM/PC (those familiar with VM/SP)
- Operating Hints (those interested in System Operation)

- ASCII to EBCDIC Conversion (those interested in data representation)
- Distribution Diskette Contents. (those interested in file names)

The *VM/PC User's Guide* has the following parts:

1. Introduction - This is an overview of the VM/PC system. It tells what you need, what you get, and how the pieces interact. All users should read this chapter.

2. Installing Your System - This is a step-by-step procedure to follow when bringing up your VM/PC system. All users who must install VM/PC should read this section. If VM/PC has already been installed on your Personal Computer, you need not read this chapter.

3. Configuring Your System - This tells why you might, and how you would, configure your system. It includes a step-by-step example of a basic configuration process. All users should read at least the introduction to this chapter.

4. Getting Started - This tells what to do to begin using VM/PC on your Personal Computer. It explains the Keyboard (and how to use it), the terminal screen (and how to read it), Session Selection, Logon, and the History Buffer. All users should read this chapter.

5. Using VM/PC Commands to Do Work - This documents commonly used VM/PC commands, grouped by task. There are descriptions of which commands to use to do work for you and examples of how you would use them. All users should read this chapter.

6. Remote Services - Information on Remote/Local sessions, Upload/Download, and Local and Remote Spooling. Anyone who needs to install the Remote Server program to enable a Remote/Local session should read this chapter. If the Remote Server has already been installed at the remote system, you need not read the installation section. Anyone who wants further information about using Remote/Local sessions should read this chapter.

7. Commands - This chapter has two parts:

   a. CP Commands - Contains explanations, formats, and operand descriptions for all VM/PC CP commands.
   b. CMS Commands - Contains explanations, formats, and operand descriptions for all VM/PC CMS commands that are different from VM/SP Release 2 commands. For those that are not different, only the explanation and format are shown.

   All users who want Reference Information about commands used with VM/PC should read this chapter.

8. Messages - This chapter contains all VM/PC messages, and all VM/PC CP messages with a brief indication of what recovery action to take. The same information is provided for those VM/PC CMS messages an end user is most likely to encounter. Users are referred to the VM/SP System Messages book for any further information. All users who want Information about messages received from VM/PC should read this chapter.

9. Appendix A - CP/CMS Interface. All users who want information about the internal operations of VM/PC should read this appendix.

10. Appendix B - 370 Processor Control. All users who maintain the Control Program should read this appendix.

11. Appendix C - Comparison of VM/SP and VM/PC - This appendix should be read by experienced VM/SP users. It lists the functions found in VM/SP but not found in VM/PC and explains why they're not there.

12. Appendix D - Operating Hints

13. Appendix E - ASCII to EBCDIC Conversions

14. Appendix F - Distribution Diskette Contents

15. Glossary - All users should read this to find the meaning of terms and acronyms used in this book.

16. Index - All users should use the index when trying to find where information is in this book.

If you want further information about using VM/PC, you must consult the VM/SP Release 2 publications. The names and order numbers for those publications are listed in "Order Numbers for VM/SP Release 2 Books" on page C-9. Note that you must use the special order numbers given there to order those books to ensure that you get the correct-level manuals.

The VM/PC System

After you start the VM/PC application, you have the option of selecting the type of connection that meets your processing needs. You can:

- Connect to a remote computer as a 3277 or a 3101 terminal. This computer does not have to be located in your immediate area. This type of connection is called a 'remote' session, or

- Connect to the local computer in the IBM Personal Computer XT/370 and run the local VM/PC system. This type of connection is called a 'local' session, or

- Connect to a remote computer and the local computer at the same time and maintain 'concurrent' remote and local sessions.

  The concurrent session selection allows you:

  – To maintain up to three separate and distinct sessions, that is, a local VM/PC session and two remote sessions with any computer system that supports 3277 or 3101 data streams, or

  – To combine the local and remote 3277 sessions into a single processing environment and allow data sharing between your IBM Personal Computer XT/370 and your remote computer. In this case, only the 3277 communication facilities can be used. The remote system must be an IBM VM/370 Basic System Extensions Program Release 2 or VM/SP level system on which the Remote Server Program has been installed. This installation process is described in Chapter 6, "Remote Services" on page 6-1.

The selection of these options is described in Chapter 4, "Getting Started" on page 4-1.

Using Local VM/PC Services

Several people can share the 370 system but only one person can use it at a time. If many users have access to your IBM Personal Computer XT/370, you can control access to VM/PC and to each users' data. This process is described in Chapter 3, "Configuring Your VM/PC System" on page 3-1.

If you are familiar with VM on a 3277 Display, you will at once notice the differences between that keyboard and the IBM Personal Computer keyboard. The VM/PC key selection closely adheres to the function of the key on the IBM Personal Computer. The VM/PC keyboard layout is described in Chapter 4, "Getting Started" on page 4-1.

VM/PC commands are format compatible with VM systems. Some commands and options have been modified or deleted to accommodate the IBM Personal Computer XT/370 environment. Some commands and options have been added for the same reason. The CP and CMS commands are described in Chapter 7, "VM/PC Commands" on page 7-1.

Using Remote VM Services

As mentioned above, you can combine your remote VM session with your local session into a single environment. This will allow you to use some of the resources of your remote system as if they were attached to your local VM/PC session. The remote server program, called VMPCSERV, will allow you to:

- Print files on your remote VM system's printers
- Gain access to CMS minidisks on your remote system
- Use data located on your remote VM system.

When you are using the VMPCSERV program to provide remote services to your local session, you will be able to spool your print output to your remote virtual printer, and any printing done on your local session will be sent to your remote virtual printer. In this way, you have access to high speed printers and can tag the print files so that they may be sent to other printers in your VM network.

VMPCSERV also allows you to link and access CMS minidisks which are not located on your local session, but are located at your remote session. This allows you to:

- Run programs that are on minidisks attached to your remote system
- Copy files from your remote system for use on your local system
- Copy files from your local system for use on your remote system.

See Chapter 6, "Remote Services" on page 6-1, for more information on how you can use the remote server, VMPCSERV, to gain access to the resources of your remote VM system.

Chapter 2. Installing VM/PC

About This Chapter

The program files and data files that make up the VM/PC product are stored on six diskettes that you can find at the back of this book. You will have to move many of those files onto your system before you can use the functions they provide. The process of moving the needed files to your system is called "installation."

In this chapter, we will cover these topics dealing with installation:

- We will take you through a list of things that need to be checked before beginning the installation process.
- We will ask you to make backup copies of your VM/PC Distribution Diskettes.
- We will guide you step-by-step through the VM/PC installation process. As you go through each step, you may need to make some choices about files already on your system. If so, we will tell you what you need to consider in making each choice.
- We will tell you about the amount of space on your fixed disk(s) that VM/PC needs during installation and afterward.
- We will tell you what to do if, for some reason, something goes wrong during the installation process.
- We will tell you how to repeat the installation process. You will want to know this if something goes wrong during installation or if the copy of VM/PC you installed on your fixed disk becomes damaged.

Before You Begin

The VM/PC installation process is designed to make it easy for you to use. There are, however, several things you need to check before you begin:

- First of all, make sure that you are installing VM/PC on a computer system satisfying the hardware and software requirements described in the previous chapter. In particular, before beginning the installation of VM/PC, you should have completed the installation of the Personal Computer XT/370 and DOS as described in the *Personal Computer XT/370 Guide to Operations* and the *Personal Computer Disk Operating System* manuals. If this has been done, your fixed disk (the C disk) will have been formatted by DOS and will contain the Personal Computer XT/370 Personality file (IMLPU.1VM).

- Next, make sure that VM/PC needs to be installed.

  If you are sharing your computer with others who may already be using VM/PC, ask them if you need to install VM/PC. You may only need to add some information to certain existing files to begin using the VM/PC system yourself.

  If you are the first person to use VM/PC on your computer, you will have to follow the installation procedure in this chapter. Otherwise, you can skip this chapter and go to Chapter 3, "Configuring Your VM/PC System" on page 3-1, for information on adding users to a VM/PC system.

- Next, make sure you have all the diskettes you will need.

  You will need all six of the VM/PC Distribution Diskettes found at the back of this manual.

- A fixed disk on your system will be used as "the C disk" during the installation process. The disk should contain DOS, as many of its commands and services will be used to install VM/PC. You may need to refer to the manual supplied with DOS.

– NOTE:

  The VM/PC installation process always uses only one fixed disk (the C disk). You may have a system with two fixed disks. If so, you can direct the installation to your D disk prior to beginning the installation process. To do this, issue the DOS command:

  ```
  ASSIGN  C=D
  ```

  Following installation, the ASSIGN command can be used whenever you wish to use VM/PC. If you want to run VM/PC from your D disk without using the ASSIGN command, or if you want VM/PC to use both of your fixed disks, read Chapter 3, "Configuring Your VM/PC System" on page 3-1.

- Find out how much unused space is available on your C disk. You can do this by using the DOS commands CHKDSK or DIR. If you do not have at least 1,600,000 bytes of available space, you should read the section "How Much Disk Space Is Required?" before continuing.

Once you have satisfied the items to be checked, you are ready to make your backup copy of the VM/PC Distribution Diskettes.

Copying VM/PC to Backup Diskettes

We strongly recommend that you make backup copies of all six VM/PC diskettes (provided with this manual) before you copy them onto your system. If a diskette somehow becomes damaged, or if files are accidentally erased, you will still have all your information.

The backup procedure is described in your Personal Computer Disk Operating System manual under the heading "Backing Up a Diskette." In choosing a way to backup your diskettes, it may be helpful to refer to Appendix F, "Distribution Diskette Contents" on page F-1.

Copying VM/PC to Your System

Before giving you detailed instructions, you should have a general idea of what the process of copying VM/PC to your system involves.

Basically, you:

- Insert one of the VM/PC Distribution Diskettes in drive A
- Type the name of a VM/PC installation program that, after you press ENTER, takes over the installation process and tells you what to do next
- Answer any questions displayed on your screen by the program
- And, finally, insert in drive A the remaining VM/PC Distribution Diskettes as requested by the installation program.

If you have satisfied the items listed in "Before You Begin," the entire installation process (which we describe in thirteen steps) often takes less than ten minutes.

When you are ready, bring up your DOS (You MUST boot from drive C) so that the "C" prompt appears on your screen; then follow these instructions:

- STEP 1. Insert VM/PC Distribution Diskette 1 in drive A.

  This "Utilities" diskette contains the VM/PC installation program and several related files.

- STEP 2. Type A:INSTALL and press ENTER.

  INSTALL is the name of the VM/PC installation program. It uses DOS commands and a program written in Advanced BASIC to guide you through the rest of the installation process.

If you wish, you can simply let the installation program lead you from here. If not, or if something happens you don't understand, you can read the rest of these instructions.

– NOTE:

  If, at any time during installation processing, something happens that seems to be an error, or if a screen is presented that is not described in the step-by-step instructions, refer to the section "If Something Goes Wrong?" later in this chapter.

After you press ENTER, the installation program displays a logo screen and asks you to press ENTER to continue.

- STEP 3. Press ENTER.

  After pressing ENTER, the installation program makes sure that the files it needs are on your C disk. The program also checks to see if you are already using files that have the same names as it intends to use temporarily during the installation process. If everything is alright, it copies several temporary files from the distribution diskette to your C disk, and you see:

```
                    I B M   V M / P C

          I N S T A L L A T I O N   P R O C E D U R E   M E S S A G E

The installation process has completed its preliminary checking of your
system.  So far, everything is in order.

Several temporary files will now be copied to your C disk.  These files
will be used to continue the installation process.

DO NOT REMOVE the diskette in drive A until told to do so.

                  After reading this message, press ENTER.
```

After you see the above screen, continue with STEP 4.

If, instead of the awaited screen, you see the following screen:

```
                    IBM VM/PC INSTALLATION

                INSTRUCTIONS FOR EXISTING FILES

Your C disk already has one or more files with the same name as a file
needed during VM/PC installation.

The next screen you will be shown is a list of the EXISTING file(s) that
you may wish to SAVE.

For each EXISTING FILE, you will have to choose one of three actions:

     SAVE the file     - a new file with same name will NOT be installed.

     RENAME the file   - the EXISTING file will be given a new name (which
                         you supply), and a new file will be installed.

     REPLACE the file  - a new file with same name WILL be installed.

You will also be given a chance to QUIT (if you need to study your
choices).

To see the list, press ENTER.
```

then you will need to make some choices about files already on your C disk. This DOES NOT MEAN anything has gone wrong. The installation program will help you with your choices, but you will have to refer to the "What About Existing Files?" section of this chapter for the instructions you need.

- STEP 4. Press ENTER.

  After pressing ENTER, you will see the names of the temporary files as they are copied to your C disk. Next, the following screen will appear:

```
                    I B M   V M / P C
      I N S T A L L A T I O N   P R O C E D U R E   M E S S A G E
Please remove any diskette now in drive A.
Insert the following VM/PC distribution diskette into drive A:

        VM/PC   Control Program                    2

To continue, press ENTER.
                (If you wish to bypass this diskette, press 'B'.)
```

After the screen appears, continue with STEP 5.

- STEP 5. Insert the requested diskette in drive A and press ENTER.

  The installation program will then copy the files from the diskette to your C disk and you will see these messages:

```
C>COPY A:CONFIG.1VM C:/V
        1 File(s) copied
C>COPY A:VMPCCON.EXE C:/V
        1 File(s) copied
C>COPY A:VMPC.COM C:/V
        1 File(s) copied
C>COPY A:CPIO.1VM C:/V
        1 File(s) copied
C>COPY A:CP.1VM C:/V
        1 File(s) copied
C>COPY A:CPMSG.1VM C:/V
        1 File(s) copied
C>ECHO OFF
```

After the above messages appear, go to STEP 6.

If you cannot find the requested diskette, or if you have changed your mind about installing VM/PC at this time, just press the 'B' key.

The installation program will then show you this screen:

```
                    IBM VM/PC INSTALLATION

                  * * * DISKETTE 2 BYPASS * * *

You have chosen to BYPASS installation processing
for VM/PC distribution diskette 2.
If you are sure you wish to do so, press B again, and
no VM/PC functions will be loaded from the diskette

If you wish to return to the previous screen, press ENTER.
```

This ensures that you do not bypass the diskette accidentally.

If you still wish to bypass the diskette, just press 'B' again, and you will prevent the copying of any files from the requested diskette. Your decision to bypass will NOT AFFECT installation processing for any other distribution diskettes.

If you decide not to bypass, just press ENTER and you are back at the start of STEP 4.

- STEP 6. Wait for the following screen to appear:

```
                    I B M   V M / P C

      I N S T A L L A T I O N   P R O C E D U R E   M E S S A G E

Please remove any diskette now in drive A.

Insert the following VM/PC distribution diskette into drive A:

        VM/PC   CMS                                  3

To continue, press ENTER.

                (If you wish to bypass this diskette, press 'B'.)
```

After the screen appears, continue with STEP 7.

- STEP 7. Insert the requested diskette in drive A and press ENTER.

  The installation program will then copy the files from the diskette to your C disk and you will see this message:

```
C>COPY A:CMS.1VM C:/V
        1 File(s) copied

C>ECHO OFF
```

  After the above message appears, go to STEP 8.

  As in STEP 5, you can bypass this step by pressing the 'B' key.

- STEP 8. Wait for the following screen to appear:

```
                    I B M   V M / P C

    I N S T A L L A T I O N   P R O C E D U R E   M E S S A G E

Please remove any diskette now in drive A.

Locate the following three VM/PC distribution diskettes ...

   VM/PC Library  (backup diskette 01)          4

      VM/PC Library  (backup diskette 02)       5

         VM/PC Library  (backup diskette 03)    6

These diskettes contain ONE large file which will be moved to your C disk
automatically using the 'RESTORE' command, which will ask for the three
diskettes (in order) using the names ...
    'backup diskette 01', 'backup diskette 02' and 'backup diskette 03'.
When you are ready with these three diskettes, press ENTER.
```

  After the screen appears, locate the three requested distribution diskettes. You will notice that you have not

yet been told to insert any of the three diskettes into a diskette drive. The screen was shown to prepare you to properly respond to future prompts from the DOS RESTORE command. The RESTORE command will soon be used to create a single file on your C disk from data previously stored (using the DOS BACKUP command) on the three VM/PC Distribution Diskettes 4, 5, and 6.

– NOTE:

  You may wish to read the description of the RESTORE command in the DOS manual before continuing.

After locating ALL THREE diskettes, go to STEP 9.

- STEP 9. Press ENTER.

  After you press ENTER, the installation program invokes the RESTORE command. Then you will see:

```
Insert backup diskette 01 in drive A:
Strike any key when ready
```

  When you see the above prompt, insert VM/PC Distribution Diskette 4 in drive A and go to STEP 10.

- STEP 10. Press any key.

  – NOTE:

    When we say "any key," we mean "any key or combination of keys that generates a character." This meaning is used consistently by DOS.

  After moving the data on the diskette in drive A to the C disk, the RESTORE command prompts you for the next distribution diskette. You will see:

```
Insert backup diskette 02 in drive A:
Strike any key when ready
```

When you see the above prompt, insert VM/PC Distribution Diskette 5 in drive A and go to STEP 11.

- STEP 11. Press any key.

  After moving the data on the diskette in drive A to the C disk, the RESTORE command prompts you for the next distribution diskette. You will see:

```
Insert backup diskette 03 in drive A:
Strike any key when ready
```

When you see the above prompt, insert VM/PC Distribution Diskette 6 in drive A and go to STEP 12.

- STEP 12. Press any key.

  After moving the data on the diskette in drive A to the C disk, the RESTORE command processing is complete. Go to STEP 13.

- STEP 13. Wait for the following screen to appear:

```
                    I B M   V M / P C

    I N S T A L L A T I O N   P R O C E D U R E   M E S S A G E

The VM/PC installation processing is nearly complete.

When you see the standard prompt from DOS,
the VM/PC installation processing will be complete.
```

After the screen appears, wait for the DOS prompt to appear. When you see the prompt, the VM/PC installation process is complete. You may remove any distribution diskette still in drive A.

Now that you have completed the installation process, you may begin to use the VM/PC functions. The instructions you will need to do that are found in Chapter 4, "Getting Started" on page 4-1.

You will not need to use any of the VM/PC Distribution Diskettes again until you are ready to use the Remote Server functions described in Chapter 6, "Remote Services" on page 6-1.

The remainder of this chapter deals with information that you will need only if something went wrong or if you simply want to understand more about the installation process.

How Much Disk Space Is Required?

The amount of space on your fixed disk(s) required by VM/PC can be looked at in two ways:

- How much space is needed to simply complete the installation process?
- Once VM/PC is installed, how much additional space is needed?

At Installation Time

You will need just under 1,600,000 bytes of available space on your C disk to install VM/PC. This space is needed to store the VM/PC permanent files for as long as you wish to have them resident on your system.

You can determine the available space on your C disk by using DOS commands (CHKDSK or DIR).

If you attempt to install VM/PC on a system having too little available disk space, the installation process will not complete.

If you intend to run the Remote 3101 Session, you will need to also install the IBM 3101 Emulation Program Product, which has disk space requirements that must be added to those of VM/PC.

After Installation

Three of the four VM/PC sessions can be used immediately after installation without requiring additional disk space. These three sessions are:

- The Remote 3277 Session
- The Remote 3101 Session
- The 370 Processor Control Session.

To use the Local 3277 Session, you will need to have enough disk space for three things:

- Your 370 Virtual Machine Paging Activity

  When you use the Local 3277 Session, you will be making use of "370 Virtual Machine" facilities. These facilities require disk space equal in size to the size of your "virtual machine."

  The size of your virtual machine can be as large as 4 megabytes (4,194,304), but 1 megabyte (1,048,576 bytes) or even 512k bytes (524,288 bytes) is enough for many users.

- The VM/PC CMS Minidisks

  Users of the Local 3277 Session generally prefer to have collections of files available on their fixed disk(s). These collections of files are called "CMS Minidisks," and their sizes are determined by the amount of information the user wants to have available.

  A minidisk can be as large as 10 megabytes (an entire fixed disk), but minidisks of about 1 megabyte provide adequate space for many activities.

  As you will learn in Chapter 3, you can allow several users to share the space on the fixed disk(s). Each of these users may wish to have their own minidisks. It is helpful to know, however, that it is possible for several users to use the same minidisk. Also, it is possible to define a minidisk on a diskette, in which case the maximum size of the minidisk is 354k.

- Printer Spooling Activity

  Users of the Local 3277 Session have the ability to tell VM/PC to print a file and, without waiting for the file to print, continue with other activities. This capability is called "printer spooling" and is accomplished by copying the files to be printed onto a disk until there is time to print them. As these files are printed, they are erased from the disk.

  You will want to leave some disk space free to allow for spooling your printer files. The kind of files you expect

to be printing will determine your spooling space requirements. For many users, 100k or 200k would be adequate.

An example may be useful. Suppose you want to install VM/PC and use the Local 3277 Session. You want two minidisks; one of 1 megabyte and the other twice that size. You also intend to run programs that will require a virtual machine size of 1 megabyte. You further suppose that your printer spooling requirements will be nearly 200k bytes. Your disk space calculations could look something like this:

```
VM/PC files (installation)  1,600,000
Smaller minidisk            1,048,576
Larger minidisk             2,097,152
Virtual machine activity    1,048,576
Printer spooling activity     200,000
                            ---------
Disk Space Required     =   5,994,304
```

Under the assumptions of our example, about 4,000,000 bytes of space would still be available on your fixed disk for additional printer spooling space, for DOS and for other uses.

What About Existing Files?

The VM/PC installation process will never write over one of your existing files without giving you a chance to prevent it.

If the installation process discovers an existing file having the same name as a file it wants to write, the process will either terminate itself or give you a chance to decide what to do before continuing.

The process terminates if the file it wants to write is just a temporary file used only during installation. The section on "If Something Goes Wrong?" discusses this situation.

If the file the installation process wants to write is one of the permanent files that make up VM/PC, you will be presented with the "INSTRUCTIONS FOR EXISTING FILES" screen described in STEP 3.

After reading the instructions, you should press any key to see a list of the files that already exist. The list looks like this:

```
                        EXISTING VM/PC FILES

 To choose an action for a file, use the '↑' and '↓' keys to move the
  '==>' pointer to the proper line.  Then press a function key:
    F1 - SAVE THE EXISTING FILE              F9  - TO REVIEW INSTRUCTIONS
    F2 - RENAME THE EXISTING FILE            F10 - QUIT
    F3 - REPLACE THE EXISTING FILE
 Press ENTER after all choices are complete.
--------FILENAME---------DESCRIPTION-------------------STATUS ----------
==> CONFIG.1VM       VM/PC CONFIGURATION FILE          Action Required
    VMPCCON.EXE      VM/PC CONFIGURATOR PROGRAM        Action Required
    VMPC.COM         VM/PC PROGRAM                     Action Required
    CPIO.1VM         VM/PC CP I/O SERVICES             Action Required
    CP.1VM           VM/PC CP 370 NUCLEUS              Action Required
    CPMSG.1VM        VM/PC CP MESSAGE FILE             Action Required
    CMS.1VM          VM/PC CMS PROGRAM                 Action Required
    CMS.100          VM/PC CMS SYSTEM LIBRARY          Action Required
```

The list will show only the files that exist, so the list you see may not include all eight of the VM/PC permanent files.

You will have to follow the instructions on the list screen (or refer to the instructions screen using the 'F9' key) and decide about each file on the list.

You will want to 'SAVE' a file if it is used by VM/PC and contains information that should NOT be over-written during the installation process.

You will want to 'REPLACE' a file if it is used by VM/PC but contains information that you are willing to have over-written by the installation process.

You will want to 'RENAME' a file if it is NOT to be used by VM/PC and it contains information that should NOT be over-written during the installation process.

Decisions to 'REPLACE' a file and to 'SAVE' a file are not acted upon until you complete the list, so they can be changed as often as you like prior to pressing ENTER.

However, decisions to 'RENAME' a file are handled differently. After pressing the 'F2' key, you will be prompted for the new name to be assigned the existing file. Once you enter the new name, the installation program attempts to rename the file immediately. If the rename attempt fails, you will see an explanatory message. If the rename attempt succeeded, the status field on the list will be changed to 'RENAMED'. Once a file has been successfully renamed, you cannot use the list screen to 'RENAME' it again. Also, the installation program will not let you use the 'RENAME' key to change the name of one of your existing files to a name that is already in use.

When you have made a decision for each file on the list screen, you press the ENTER key to notify the installation process that it can proceed. The installation process will change itself (if necessary) to prevent the installation of the files you chose to 'SAVE'. For this reason, some of the step-by-step instructions described above may be bypassed.

If Something Goes Wrong?

The VM/PC installation process is designed to be trouble free. In the step-by-step instructions above, we have talked about the screens you are expected to see and the actions you are supposed to take when everything goes as intended.

If, for whatever reason, something happens that you cannot find described earlier in this Chapter, we suggest you read this section. If you cannot find the explanation you need here, or elsewhere in this manual, refer to the *Personal Computer XT/370 Guide to Operations.*

DOS Facilities Missing

If you see this screen:

```
                    I B M   V M / P C

    I N S T A L L A T I O N   P R O C E D U R E   M E S S A G E

The procedure which copies the VM/PC system from the distribution
diskettes to your fixed disk uses DOS commands and facilities.

The installation procedure was UNABLE TO FIND one or both of the
DOS commands (BASICA or RESTORE) that it needs.

To use the VM/PC Installation Procedure, you will have to ...

       1) copy the DOS commands onto your fixed disk, and

       2) restart the installation procedure.
```

you need to review the list of items that must be checked before beginning the installation process (in the "Before You Begin" section earlier in this chapter). After doing so, start the installation process over again at STEP 1.

VM/PC Temporary Files Already Exist

If you see this screen:

```
                    I B M   V M / P C

    I N S T A L L A T I O N   P R O C E D U R E   M E S S A G E

During the installation process, several small files will be created
temporarily on your C disk.

These temporary files will be erased automatically at the end of the
installation process, but the file names used may conflict with
existing files.

The installation process found the following existing files with the
same names as those to be used for temporary files:

 TVMPC.DSK       TVMPCHD.MSG   TVMPCDD2.BAS   TVMPC01.BAS
 TVMPCLC.BAT     TVMPC01.MSG   TVMPCDD3.BAS   TVMPC03.BAS
 TVMPC03.MSG     TVMPCDD4.BAS  TVMPC.100

To avoid possible damage to your existing files, you must either rename
or remove them using DOS commands such as RENAME, COPY, ERASE etc.
After doing so, you may restart the installation process.
```

you must use the appropriate DOS facilities (COPY, RENAME, BACKUP, ERASE etc.) to remove any conflicting filenames. After the names are removed, you can restart the installation procedure at STEP 1.

Note: The screen you see may not contain all of the file names shown.

Insufficient Space on C Disk

If there was not enough free space on your C disk to install the VM/PC functions, you will see:

```
                    I B M   V M / P C

    I N S T A L L A T I O N   P R O C E D U R E   M E S S A G E

              * * *  INSUFFICIENT FIXED DISK SPACE  * * *

            The installation process DID NOT COMPLETE.

1,600,000 bytes of C disk storage are required to install your VM/PC files.

          That amount of free storage is NOT available on your C disk.

             After reading this message, press ENTER.
```

If you see the previous screen, the installation process has failed. You will have to use the facilities provided by DOS to make sufficient space available. Before doing so, you should read the section on "How Much Disk Space Is Required?." After sufficient space is made available, read the section on "Repeating the Installation Process." Then repeat the installation process.

Note: You may see a value other than 1,600,000 appear on your screen. This can happen because the Installation Program calculates the storage required based on your needs.

Unable to Read Space Available on C Disk

If, for some reason, the installation program was unable to read how much free space is on your C disk, you will see:

```
                    I B M   V M / P C
I N S T A L L A T I O N   P R O C E D U R E   M E S S A G E
   * * *  UNABLE TO READ FREE SPACE ON FIXED DISK  * * *
          The installation process DID NOT COMPLETE.
          After reading this message, press ENTER.
```

If you see the previous screen, the installation process has failed. Before you can attempt to install VM/PC again, you will have to make sure your C disk is properly prepared. The list of items to check at the beginning of this chapter should help, as should the DOS CHKDSK command. When you are satisfied that your C disk is ready, you should read the section on "Repeating the Installation Process." Then repeat the installation process.

VM/PC Distribution Diskette Not Used

If you are the first to install VM/PC on your system, then all six of the VM/PC Distribution Diskettes will be used by the installation process. If you are the first to install VM/PC and not all six diskettes were used, then there is reason to suspect something has gone wrong. In that case, you should review the items listed at the beginning of this chapter. If

everything on the list is satisfied, attempt to repeat the installation process (after reading the section below on "Repeating the Installation Process").

If the problem persists, refer to the *Personal Computer XT/370 Guide to Operations.*

On the other hand, if you are not installing VM/PC on your system for the first time, it is likely that you needed to make choices about existing files (see STEP 4 and the section on "What About Existing Files?"). Those choices can result in the skipping of one or more of the VM/PC Distribution Diskettes by the installation process. If so, everything is fine, and you simply follow the step-by-step instructions as if the steps involving the unused diskettes were not there.

Repeating the Installation Process

For whatever reason, if it becomes necessary to repeat the VM/PC installation process, the following information will be useful:

- The installation process uses several temporary files which it erases upon normal completion. If the installation process failed or was ended before it completed, several of these temporary files may still be on the C disk. These files MUST be erased prior to repeating the installation process, since the installation program will not overwrite any of your existing files.

  All of the temporary file names begin with 'TVMPC'. By issuing the DOS command:

  ```
  DIR  TVMPC*.*
  ```

  you can determine the presence of temporary files.

- If VM/PC had been previously installed on your computer, or if the installation process had partially completed, all (or some) of the eight permanent files that are normally installed will still be on the C disk.

  It is not necessary to do anything about these permanent files before repeating the installation process. The installation program will detect their existence and give you a chance to decide what to do. Read the section above on "What About Existing Files?" for more information.

  Incidentally, this ability of the installation program makes it convenient to re-install even a single VM/PC permanent file if the copy of that file on your C disk becomes damaged. This is done by starting the installation process and, when eventually presented with the "EXISTING VM/PC FILES" list screen, pressing the SAVE key for all the files that are undamaged and the REPLACE key for any damaged files. If a file is severely damaged, it will not even appear on the list screen. Such severely damaged files will be re-installed automatically by the installation process.

Chapter 3. Configuring Your VM/PC System

The Configuration File

One of the files that is copied to your C disk when VM/PC is installed is called the Configuration file. It contains information about the way you want to use VM/PC and about the files and other resources you want VM/PC to use.

This information, which is read every time you run the VM/PC sessions, includes the following:

- The names of the VM/PC programs and files as well as the drive on which to find them (This gives you the ability to rename them or move them to a disk other than your C disk.)
- The names (USERIDs) of those allowed to use the Local 3277 Session
- For each user named,
  - A password to prove identity
  - The size of the 370 virtual machine
  - The number and size of any minidisks owned by the user
  - The names of other users' minidisks that the user needs to access (This is called "linking.")
- The amount of printer spooling activity to expect
- Which options and automatic features of VM/PC to use

Because of the information in the Configuration file, it is possible to have several people use the same VM/PC system without damaging each other's files. Also, a single minidisk can be shared by several users, making it unnecessary for

each to have their own copy of common data. Of course, more than one person cannot use the system at the same time.

To help you get started, the Configuration file that is installed with VM/PC contains a sample user name and other needed information. If you wish, you can use this Configuration file to become familiar with VM/PC. If so, you can go to Chapter 4 now and postpone reading the rest of this chapter until you feel the need to modify the Configuration file.

Since the Configuration file is so important to VM/PC, the information in the Configuration file must be accurate and consistent. For the same reason, when several users are using the same system it may be necessary to allow only one person to see or to change the Configuration file. To help ensure both the accuracy of Configuration file data and to prevent uncontrolled access to that data, VM/PC supplies you with the Configurator program.

The Configurator Program

The Configurator is a specialized editor for the Configuration file. It guides you through the information on the file and allows you to display or modify whatever you want. With this editor you can define new CMS minidisks, extend old ones, define new VM/PC users, delete old ones, define your local printer spooling options, inform VM/PC of new system definitions, etc.

The Configurator will also let you define a password so that only those who know the password can use the Configurator.

Starting the Configurator Program

The Configurator program runs directly under the control of DOS. So if you have installed VM/PC as instructed in Chapter 2, simply prepare your system so that the prompt of C> indicates that DOS is ready, and enter "VMPCCON."

DOS will then load the Configurator program (VMPCCON.EXE) and the IBM logo and copyright screen will be displayed.

Note: The Configurator program will always look for the Configuration file (CONFIG.1VM) on the DOS default drive (that is, the C drive if the prompt is C>). The Configurator uses a work file named CONFIG.UT1. If you have a file by this name on the default drive, you will be prompted for permission to erase it.

Using the Configurator

After starting the Configurator (which uses the standard, PC/XT keyboard layout), press the ENTER key to continue.

You will see a message indicating that the Configurator is reading the Configuration file.

Note: If, for whatever reason, the Configuration file does not exist, the Configurator will provide defaults from which you can build a new Configuration file. However, using the

Configurator Program to modify a copy of the Configuration file supplied with VM/PC is often an easier way of building a new Configuration file.

Before telling you how to work with specific Configurator screens, there is some general information about the way these screens are arranged, about the use of function keys, and about some of the other keys that you need to understand.

Configurator Screens

The screens presented by the Configurator program have much in common. Each will have a title for easy reference by other screens. This will usually be followed by a single line instruction such as "Make Changes and Press ENTER."

Most screens will have three boxes. See Figure 3-3 on page 3-14 for a sample of a typical screen.

The first of these contains the "input areas," which are the only places on the screen which will accept changes. The cursor will only move to the input areas within this box.

The second box contains "help" information, which is a brief explanation of the kinds of information to put in each input area. If there is not enough room to display all of the help information in the box, you may be directed to press a function key to see expanded information on another screen.

The third box contains descriptions of the allowable function keys and their uses.

Function Key Usage

The FUNCTION MENU screen is considered the main screen presented by the Configurator. If you have progressed to a later screen, pressing function key 3 (F3) will return you to the FUNCTION MENU screen. From that point you can start a new function if you desire. Function key 5 (F5), if pressed, will return you to the screen specified in the Function Key definition box on the current screen. This will nearly always be the previous screen.

In general, you must press ENTER to apply the changes you have made on any screen.

If you press a function key to return you to another screen without first pressing the ENTER key, the changes on that screen will NOT be made. The Configurator assumes that you changed your mind regarding the changes.

Function keys 7 and 8 (F7 and F8) are the scrolling keys. If pressed while viewing a particular screen, they will scroll you forward or backward to the next user, minidisk, or link, as indicated by the function key area on the screen.

Keyboard Usage

As we mentioned above, only one box on the screen contains input areas. These input areas are the only places on the screen where the cursor can be positioned.

As you type or try to move the cursor past the end of an input area, it will move to the beginning of the next input area if there is one. If not, the cursor will move to the beginning of the first input area in the box.

The left and right cursor movement keys on the key pad move the cursor left and right within an input area, again, moving to the next or previous area as you attempt to move past the end of the area. The up and down arrows will move the cursor to the previous or next input area respectively. The tab key also moves the cursor to the next input area, while the tab key plus the upshift key moves the cursor to the previous input area.

The INSERT and DELETE keys can be used to insert and delete characters within an input area. Insert mode will be indicated by the word INSERT displayed on line 25 of the screen. In insert mode, any characters moved out of the input area will be lost. Pressing INSERT again or any cursor control key will turn off insert mode.

The BACKSPACE key will move the cursor to the left one space and delete that character.

The Configurator Program will only accept the following characters as input for USERIDs, passwords and filenames:

A - Z (The Configurator converts lowercase to uppercase for you.)

0 - 9

. and : (These characters are allowed only in filenames)

No leading blanks are permitted in USERIDs, passwords and filenames. No imbedded blanks are permitted for any item. If you do enter an invalid character, the system will reject it. You can then correct the error and enter it again.

The Password Screen

After reading in the Configuration file, or determining that it needs to build a new one, the Configurator presents you with a screen that prompts you for your password. This is the password that is required in order to see or change the Configuration file.

The Configuration file installed with VM/PC requires a password of "SYSTEM."

To continue, type the password and press ENTER.

If, for whatever reason, you choose not to continue, you can return to DOS simply by pressing function key 10 (F10). Your Configuration file will not be changed.

Function Menu Screen

After providing the proper password, you will see the FUNCTION MENU screen, which is considered the main screen presented by the Configurator program. This screen, shown below, gives you a choice of six functions. To choose a function, type the number of the function you wish to perform, a number from 1 to 6, and press ENTER.

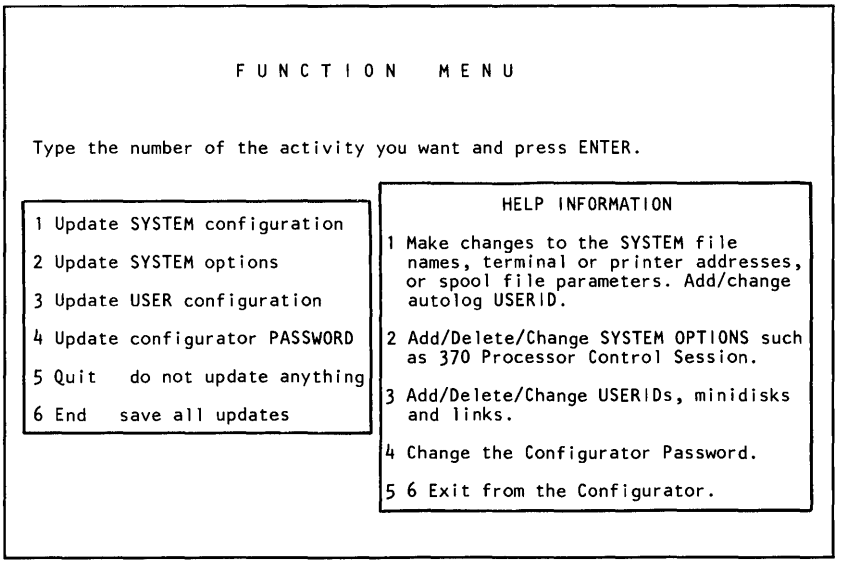


Figure 3-1. Function Menu

Quitting the Configurator

The Configurator program does not actually update the Configuration file until you have made all your changes AND then told it specifically to do so. This means that you can decide you want to quit at any time without applying the changes you have entered. If you do decide that you don't want to keep the changes you have entered, simply select the "Quit" function from the menu. You will then exit the Configurator and the Configuration file will remain unchanged. This includes any password changes you may have made.

Normal Exit from the Configurator

The "End" function is the normal exit from the Configurator. Use of this function will cause all the changes you have made since starting the Configurator to be stored on the Configuration file. First, the Configurator will perform any required file allocations for minidisks, giving you a chance to approve or cancel each allocation. After that, the Configurator writes the file and returns control to DOS, which will present its usual prompt.

System Configuration Screen

The System Configuration screen is used to define parameters that will apply to all the users of the VM/PC system. Parameters that are allowed to differ for each user are defined through the User Selection screen.

To reach the System Configuration screen from the Function Menu screen, type a "1" to choose the "Update SYSTEM Configuration" function and press ENTER. After doing so, you see the SYSTEM CONFIGURATION screen.

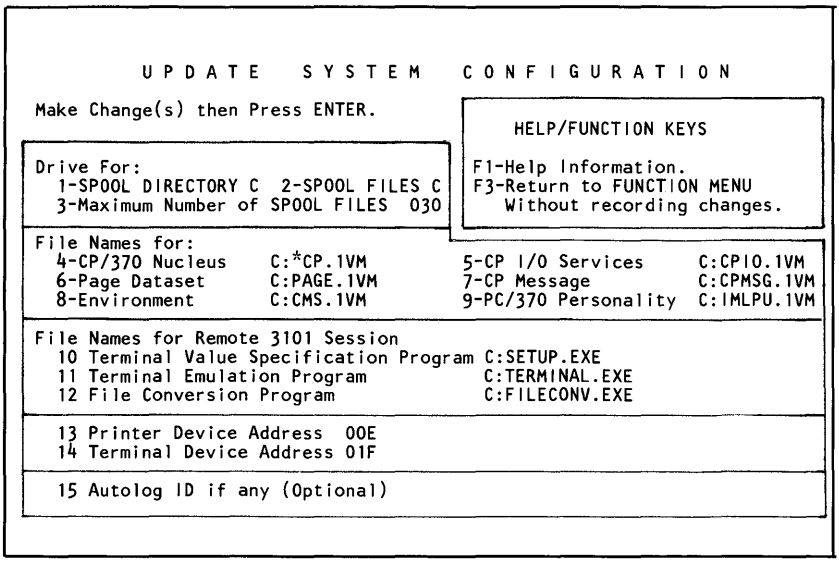


Figure 3-2. Update System Configuration Menu

The top part of the input area box deals with the system spool files, that is, printer files created by the users of the VM/PC Local 3277 Session. Printer file spooling is performed by creating a DOS file for each print file and a directory file that points to them. The directory and the spool files may or may not be on the same disk drive. By providing the drive letter (A, B, C, or D), you tell VM/PC which drive to use for these files. (Note that the B drive letter can only be used if you have two diskette drives.) You may also specify the maximum number of spool files the system will handle. The number can range from 1 to 999.

The next part of the input area box deals with the names of the six files used by VM/PC. All six of of these files must reside on one of your fixed disks (drive C or drive D), but they need not all be on the same drive. The files named are the PC XT/370 hardware personality file, the file containing the error messages issued by VM/PC CP, and the program files that make up VM/PC, including the environment program (VM/PC CMS).

You may wish to move one or more of these files to your other fixed disk. If you do, you must tell VM/PC where to look for them. This is done by changing the drive letter on the file specification displayed on this screen. Except for the Page Dataset, the drive letter for these six files must be C or D. For the Page Dataset, a file that is created by VM/PC, you may specify the drive letter to be C, D or *. In this case, the * tells VM/PC to create the file on the fixed disk with the greater amount of available storage.

Note: VM/PC allocates the Page Dataset when you start VM/PC and erases it when you logoff VM/PC, therefore this space will not be included in the response to a DOS CHKDSK command.

The third part of the input area box deals with the names of the files which contain the programs used by the Remote 3101 Session. These programs are supplied with the IBM 3101 Emulation program product. If you do not have these programs on your system, you may ignore this part of the input area box.

The fourth part of the input area box is used to specify the terminal device (console) address and printer address for your Local 3277 Session. You may wish to change these if the environment you are running in your virtual machine requires printer and/or console addresses other than those supplied.

The final part of the input area box presents an option. If supplied, the USERID specified will automatically be logged on when the Local 3277 Session is initialized.

WARNING: The use of the automatic logon option bypasses the password validation that normally occurs during the Local 3277 Session logon sequence.

Press ENTER when you are finished with changes on this screen. If no syntax or other errors are found, you will then be shown the Function Menu screen. If you have made changes but you decide you do not want them saved, press function key 3 (F3) instead of ENTER to return to the Function Menu screen without recording any changes.

Configurator Password Screen

The Configurator Password screen is used to change the password that controls access to the Configurator program. If you change the password, it will take effect the next time the Configurator program is started.

To reach the Configurator Password screen from the Function Menu screen, type a "4" to choose the "Update Configurator PASSWORD" function and press ENTER. After doing so, you see the Configurator Password screen. You may now change the password from the default of "SYSTEM" or its current value to whatever you like.

WARNING: If you change the password, you MUST NOT FORGET IT! If you do, you too will be unable to use the Configurator program in the future. If you do forget or lose the password, you will have to re-install the Configuration file supplied with VM/PC and, using the default password, rebuild the Configuration file to match your current needs.

If you return a blank password, the old password will be retained. Also if you just press ENTER without changing anything, you will, of course, not change the password. Make the change if you wish, and press ENTER to return to the Function Menu.

System Options Screen

This screen allows you to choose whether VM/PC user(s) will have access to the 370 Processor Control Session.

To reach the System Options screen from the Function Menu screen, type a "2" to choose the "Update SYSTEM Options" function and press ENTER. After doing so, you see the System Options screen.

The 370 Processor Control Session enables a skilled user to manipulate the internal registers, the memory, and the state of the 370 processor inside your system. This screen gives you a means of preventing users of the system from accidentally damaging their data by improperly using the 370 Processor Control Session.

Type a Y for Yes or an N for No and press ENTER.

User Selection Screen

The User Selection screen is the first of a sequence of screens that are used to collect parameters that are different for each user of the system.

To reach the User Selection screen from the Function Menu screen, type a "3" to choose the "Update USER Configuration" function and press ENTER. After doing so, you see the User Selection screen.

This screen allows you to add or delete users from the system and to specify the parameters of the 370 virtual machine each user will experience.

The User Selection screen contains a list of all the USERIDs currently listed in the Configuration file. If you are processing the Configuration file supplied with the VM/PC system, there will be two users listed. They are CMS and SAMPLE. Place the cursor next to the user you wish to modify/display/delete, or place the cursor on the top line next to "ADD new user" to add a user, then press ENTER. Notice that function key F3 will return you to the Function Menu.

There are five screens that are used to collect parameters for an individual user. To explain the use of these screens, we will assume the use of the Configuration file supplied with VM/PC. The next five sections of this chapter will take you through each of the screens.

User Function Selection Screen

From the USER SELECTION screen let us select user CMS and press ENTER. This takes us to the USER FUNCTION SELECTION screen. If you had chosen to add a new user, this screen would have been bypassed, and you would have gone directly to the UPDATE ENVIRONMENT FOR USER screen, which we will discuss later. Here, notice that we have 4 function keys activated. F3 returns us to the main menu as is always the case. F5 returns us to the USER SELECTION screen which was the previous screen. We will not discuss these two keys any further since they always perform the same function. This is the first screen in which the Scroll keys are activated. If you had more than 1 user listed on the previous screen, you would be able to scroll through the list using the F7 and F8 keys. Notice the name change on the top of this screen as you scroll.

Update Environment for User Screen

Select "User Environment" for user CMS and press ENTER. You will see the environment items, for the user CMS, listed on the screen. Press the scroll keys and watch all of the environment items change as you scroll through the list of users. If you select "ADD new user" and then try to scroll through the user list, you will never come back to the screen with the blank input areas, because that entry is not yet considered a user. You must go back to the USER SELECTION screen and again select to add a new user.

The "USERID" is the name given to identify the VM/PC user. It is required in order to logon and use the system. It may contain from 1 to 8 characters. You may wish to establish a convention such as using the last name of users as their USERIDs. To delete a user from the configuration file just blank out the "USERID" field on this screen.

The "Password" is a character string that provides access authority to logon to the given USERID and access the associated data. This is a secret word which should only be known by the user and by the Technical Coordinator.

The "Virtual Machine Storage Size" specifies the amount of storage the user will have simulated in his virtual machine when he logs on to VM/PC. The amount of storage is

measured in kilobytes (K). For example, 1 K of storage is 1024 bytes, and 1024 K is one megabyte. The value entered for virtual machine storage size can range from 0 K to 4096 K, and must be a multiple of 4K.

Note: The user environments you define (USERIDs, passwords, etc.) need not agree with similar items you may be using on other systems running CMS. This is true even if you intend to use the VM/PC remote services.

The "Environment to Auto IPL" field is the name of the System/370 program to be loaded in the user's virtual machine upon logon. The environment name can only be "CMS." Press ENTER to record the changes, and return to the USER FUNCTION SELECTION screen. If F5 is pressed first, none of the changes will be recorded before returning.

If you have selected the user CMS, you will see that CMS is also the password for that USERID. The CMS USERID is the owner of the CMS System Library minidisk (known as the "System disk"), and is used to maintain the files or datasets used in common by the other CMS users on the VM/PC system. The CMS virtual machine storage size is 512K.

Minidisk List Screen

Now, select "User Minidisks" from the USER FUNCTION SELECTION screen and press ENTER. A minidisk is the simulation of a disk drive by using a portion of a real drive. The minidisk appears, to the program running in the virtual machine, like a real disk drive. However, it will usually be smaller than the actual disk drive being used. You should now see the MINIDISK LIST for the user selected. Notice that you can still use the scroll keys to move between the users currently in the list.

You can now select one of the users' minidisks for modification or deletion, or optionally add a new minidisk. Let us select the 101 disk of the user SAMPLE. Make the selection and press ENTER. You will now see the screen shown in Figure 3-3.

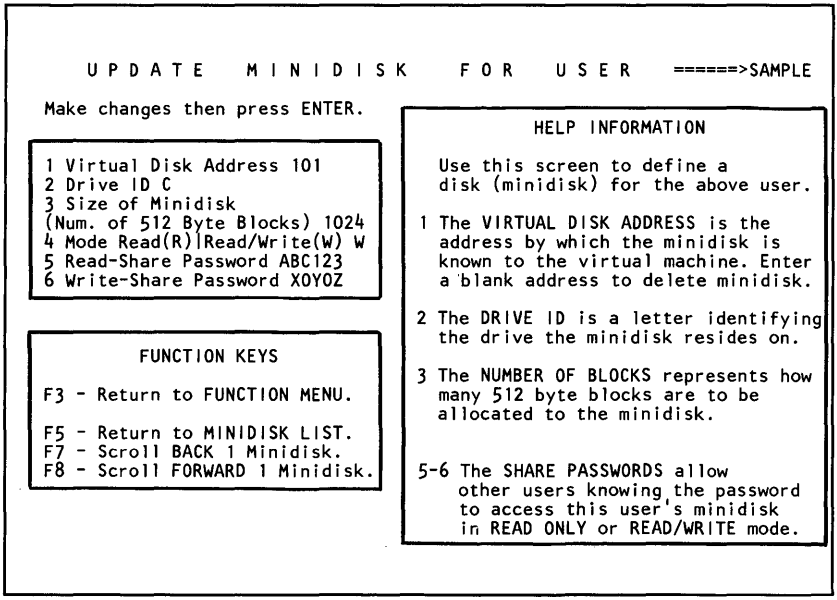


Figure 3-3. Update Minidisk for User Menu

Note that, for this screen, the scroll keys will now scroll through the minidisk list, not the user list.

The "Virtual Disk Address" is the address of that simulated disk as seen by this virtual machine. It is in the form of CUU (channel and unit address) as are all input/output device addresses in System/370. This field will be "101" since we selected the SAMPLE 101 disk. To delete a minidisk simply blank out the "Virtual Disk Address" area and press ENTER.

The "Drive ID" is the drive letter of the drive where the actual minidisk data will be simulated and stored. (Remember that the drive letter B can be used only if you have two diskette drives.)

The "Size of Minidisk" represents the size of the simulated disk. It is measured in the number of 512-byte blocks allocated to the disk. The SAMPLE 101 disk consists of 1024 physical blocks on the real drive C. That gives the SAMPLE 101 disk a storage capacity of 1024 X 512 = 512 K bytes or 1/2 megabyte. The size of a minidisk can range from 1 to 20480 blocks. For the minimum size

minidisk acceptable to CMS, refer to the CMS FORMAT command description in Chapter 7, "VM/PC Commands" on page 7-1.

The "Mode" letter can be either R (read) or W (read/write). If the disk is read only, the disk can not be modified, that is to say the virtual machine can not write on the disk. Obviously it would have to be read/write at some point in time to create it in the first place, but it can be changed to read only after it is built. A minidisk can also be copied to the real disk from another disk. A minidisk can also be copied (using DOS) from one disk drive to another. This permits you to move a minidisk from one VM/PC machine to another via diskette(s).

The DOS file name associated with a minidisk can be determined as follows. The drive letter is specified on UPDATE MINIDISK screen. The file name is the user ID of the owner of the minidisk, and the file extension is the minidisk address. For example, the CMS 100 disk as supplied with the VM/PC system will be named C:CMS.100.

The Read-share and Write-share passwords are optional. They permit links to be performed to the minidisk without having the links defined in the Configuration File. Such links are performed using the LINK and ACCESS commands described in Chapter 7.

The links defined in the Configuration file (using the UPDATE LINK screen of the Configurator) do not require Read-share or Write-share passwords and take effect automatically upon logon to the Local 3277 Session.

If these passwords are defined for a minidisk, other users of the system who know the passwords may link to and access the data on the minidisk. If you want to let users of the system link to a minidisk to read its files, you must define a Read-share password for the minidisk. If a Read-share password is not defined, the minidisk will only be readable by its owner and by users with links to the minidisk defined in the Configuration file.

If you want to let users of the system link to a minidisk to add, delete, or change files, you must define **BOTH** a Read-share and a Write-share password for the minidisk. If these two passwords are not both defined, the only users able to write to the minidisk will be its owner and those with read/write links to the minidisk defined in the Configuration file.

If you wish to allow others to access your minidisk without having to know the password, use the password of "ALL."

If you scroll through all of the minidisks owned by the user CMS, you will notice that CMS has a very large 100 disk. This is the CMS system disk, which is shared among all the CMS users as we will see later.

Press the ENTER key to record the changes, if any, and return to the MINIDISK LIST screen. If you are finished with the minidisks for this user and wish to select another user's minidisks, use the scroll keys, or go back two screens to the USER SELECTION screen by pressing function key F5 twice, and select another user.

It should be pointed out that even though the minidisk is defined to the system and it is allocated on a disk drive, it will not be usable by CMS until it is formatted by CMS using the FORMAT command. See "FORMAT" on page 7-82. Remember, when you format a disk, you are also erasing all the data on that disk. In the case of the CMS disk format, you will be erasing the files on the minidisk, not all the files on the real physical drive.

Link List Screen

Select user SAMPLE, and from the USER FUNCTION SELECTION screen select option 3 "User Links" and press ENTER. You should now see the LINK LIST screen. This screen is similar to the MINIDISK LIST screen. However, here we see a list of the Links for the selected user. You can select one of the existing Links for modification/viewing, or deletion, or select the option to add a new link.

To continue with our example, make sure the cursor is in the "ADD a new LINK" position and press ENTER.

Update Link Screen

You are now looking at the UPDATE LINK screen. This screen is used to build links to other users' minidisks.

Suppose another user of your system has a minidisk that contains information that the owner of the USERID SAMPLE routinely needs to access. Let's say that this other user is defined to the system with the USERID of STEVE, and that the minidisk that SAMPLE needs is known to STEVE's virtual machine as address "101."

You can build a link for SAMPLE to STEVE's 101 minidisk, so that both USERIDs can access the same minidisk by doing the following:

1. In the "Virtual Disk Address" field, type the address of the minidisk as seen by SAMPLE. Since SAMPLE already has a 101 minidisk, you must define the link with some other address, such as "201."

2. In the "USERID (owner)" field, type the USERID that owns the minidisk. In our example, the owner is "STEVE."

3. In the "Owner's Virtual Address" field, type the address of the minidisk as it is known to its owner. In our example, this is "101."

4. In the last field, "Link Mode," type either an "R" for read only or "W" for read/write. If SAMPLE only needs to look at the information on STEVE's minidisk, and should not be allowed to modify it, then the mode should be "R." If SAMPLE needs to modify the information, then the mode should be "W." For our example, type "R."

In this link example, you have given the user SAMPLE read-only access to the user STEVE's 101 minidisk, and the minidisk is known to SAMPLE's virtual machine as 201.

Since you will not want to save the example information, press function key 5 (F5) to return to the LINK LIST screen. Then move the cursor to the "CMS 100" link and

press ENTER. You will then see the UPDATE LINK screen that defines the link giving the user SAMPLE read/only access to the 100 minidisk owned by user CMS. This minidisk is also known to the SAMPLE virtual machine as 100. The user SAMPLE will not be able to write on his 100 disk since it is read/only. This is the CMS System disk, which contains the CMS commands, programs, macro libraries, etc. ALL CMS USERS SHOULD HAVE THIS LINK. If you go back and look at the minidisks owned by SAMPLE, you will see that SAMPLE has a read/write 101 disk and a read/write 102 disk. The 101 disk is his private storage disk, typically accessed mode "A" by CMS and commonly called the "A Disk." The access mode of his 102 disk is not automatically assigned.

Disk Allocation and Checking

When the END function is selected, (Function 6 on the FUNCTION MENU), the Configurator will test for the existence of all of the required system files. These file names were supplied on the SYSTEM CONFIGURATION screen. You will see a message if a required system file cannot be found.

Next the Configurator will look at all the changes and additions to the minidisks. If the minidisk does not exist on the designated PC disk drive, it will be allocated. If the minidisk has been enlarged, the minidisk file will be enlarged to match the specification.

Remember that CMS will not take advantage of the additional space until the CMS FORMAT command with the RECOMP option is used to recompute the minidisk size. If FORMAT is used without the RECOMP option, the new minidisk size will be computed, but all of the data previously on the minidisk will be lost.

If the size of the minidisk is reduced, the Configurator will warn you, but will not decrease the size of the allocated space. When the size of a minidisk is reduced, you run the risk of losing data.

WARNING: When the USERID that owns the minidisk next uses the Local 3277 Session, the minidisk will be reduced automatically by VM/PC CP.

The Configurator will not take that risk.

For the same reason, the Configurator will not erase a minidisk file when it is deleted from the Configuration file. If you really want to erase the minidisk, you must use the DOS ERASE command to erase the file.

Before any action is taken, you will be asked to confirm the change. You may have added a minidisk to a user, and may plan to copy the file from a diskette created by another VM/PC system. Under this condition you do not want the Configurator to allocate the space. Here you should respond with a N (No) to the allocation message. A better way to install the minidisk, in this example, would be to copy the minidisk file to the drive before doing the configuration. This way the Configurator can check the the size of the minidisk against the size you specified and extend it if necessary.

If you wish to remove a minidisk and assign it to another user, you must use DOS to rename the file according to the rules for minidisk file names specified above. You must then use the Configurator to add the minidisk to the USERID of the new owner.

Limitations

The Configurator program holds the configuration data in arrays of fixed dimension, thus imposing limitations on the amount of data it will handle. The Configurator is limited to handling 20 users, each of which can have a total of 26 Links and/or Minidisks.

The Configurator Program uses about 50,000 bytes of C disk storage for temporary workspace. This storage is freed upon exiting the Configurator Program. The filename used for the workspace is CONFIG.UT1. If such a file already exists on your system, the Configurator will ask for permission to erase it before returning to DOS. If you don't want to erase it, the Configurator will return to DOS and you

will then have to rename or remove your existing CONFIG.UT1 file (using DOS commands) before you can run the Configurator.

Chapter 4. Getting Started

Invoking VM/PC

VM/PC runs on your IBM Personal Computer XT/370 as a DOS 2.00 application. The required files were copied when you installed VM/PC according to the instructions in Chapter 2, "Installing VM/PC" on page 2-1. You start VM/PC by entering the command **VMPC** in either of these ways:

- If you want to start VM/PC every time you start DOS, you can enter the command by using the automatic program execution procedure described in the Disk Operating System manual.

- If you want to start VM/PC via keyboard entry, you can enter the command and press the ENTER key in response to the DOS prompt (C >). For example:

  C > VMPC

After you enter the **VMPC** command, you see an introduction screen which contains the product name and the IBM copyright statement.

Keyboard Usage

If you have been using an IBM 3277 Display, you will notice that the keyboard on your IBM Personal Computer is not the same as the keyboard on your IBM 3277 Display.

The following picture shows you the physical layout of your IBM Personal Computer XT/370 keyboard. The keytops which are shaded have their functions changed when the VM/PC application is running, that is, after the **VMPC** command is entered.

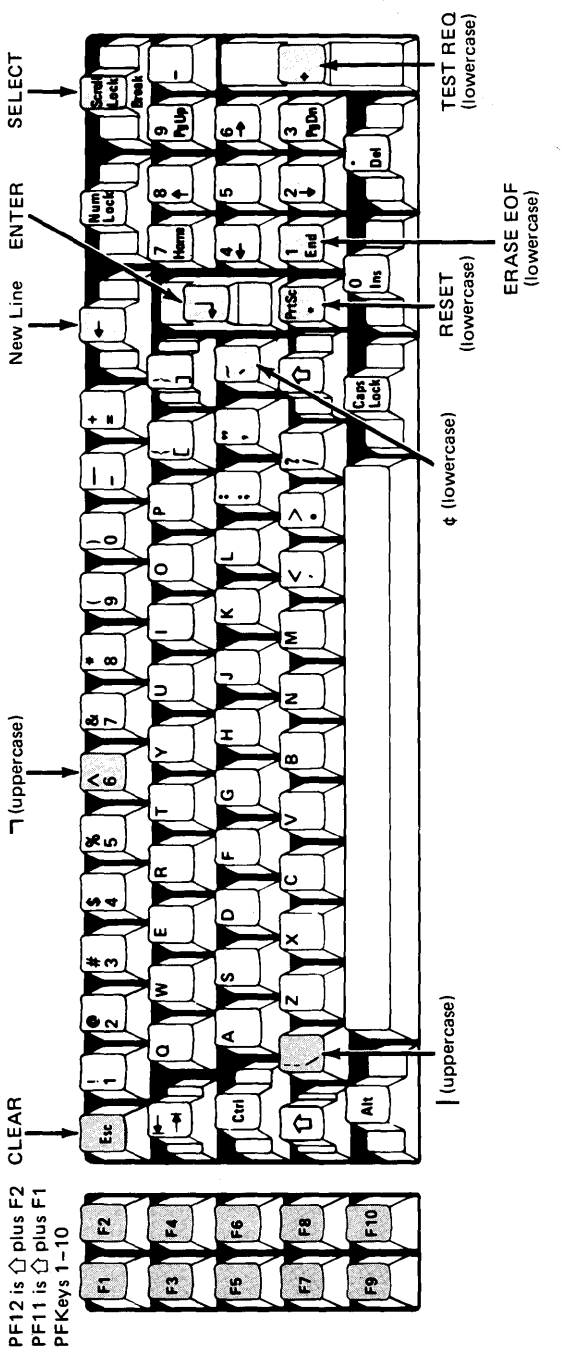


Figure 4-1. VM/PC Keyboard

Key	Description
Esc	'CLEAR' will erase all data on the display.
←——	'NL' will move the cursor to the first unprotected field on the next line.
Scroll Lock	'SELECT' will cause the Session Selection Menu to be displayed. If you press the 'Ctrl' key with this key, it causes a break and an immediate return to DOS.
←—┘	'ENTER' causes the data you have typed to be passed to the computer.
7 Home	'ERASE INPUT' will erase all the input data on your screen. It will not be passed to the computer. The 'Ctrl' key must also be pressed to use this function.
+	'TEST REQ'. When used with the shift key, it is a +.
PrtSc *	'RESET' will clear the Inhibit and Insert mode.
1 End	'ERASE EOF' will erase all data to the end of the current field.
F1-F10 F1-F2 F1-F3	'PF1' thru 'PF10' when used alone. 'PF11' and 'PF12' when used with the shift key. 'PA1' thru 'PA3' when used with the 'Alt' key.
^ 6	The uppercase on this key (^) is recognized by VM/PC as a Logical Not character.
¦ \	The uppercase on this key (¦) is recognized by VM/PC as a solid vertical bar.
~ `	The lowercase on the key (`) is recognized by VM/PC as a Cents sign.

Figure 4-2. Description of Keys

Note: The logical Not character, the solid vertical bar, and the Cents sign are not displayed on your terminal but are interpreted as such by VM/PC.

Session Selection

The Copyright screen instructs you to press the ENTER key to continue. You then see a Session Selection Menu which looks like this:

```
                        VM/PC Session Selection Menu
Press the number of the session that you want
  1 Remote 3277 Session
  2 Local 3277 Session
  3 Remote 3101 Session
  4 370 Processor Control




    Press Ctrl-Break for immediate exit to DOS
```

Figure 4-3. VM/PC Session Selection Menu

Note: Selections 1, 3, and 4 may not appear on your Session Selection Menu. See the following descriptions of these selections for information on when they will not appear. The numbers associated with particular sessions may change if you do not have all sessions available.

A 'session' is a connection you establish with a computer. Once a connection is established, you can give the computer instructions to do whatever work you want it to do. VM/PC allows you to use your IBM Personal Computer XT/370 by itself, or connected to a larger computer located somewhere else. These connections are referred to as 'local sessions' and 'remote sessions', respectively.

VM/PC will allow you have a local session and two remote sessions (3277 and 3101) at the same time. Since you only

have one display and one keyboard, you can only look at and talk to one computer at a time. This session is called the 'active' session.

The Session Selection Menu allows you to choose your active session. You see a list of the available sessions, each associated with a number. You select the active session by pressing the number assigned to that session.

You use the Session Selection Menu any time VM/PC is running to change the active session. Whenever you change sessions, the sessions which are not active are not lost. You will be notified whenever a session which is established, but not active, changes.

The Session Selection Menu is displayed whenever the SELECT key is pressed.

The sessions you may choose are:

Local 3277 Session

The Local 3277 Session allows you to activate the 370 environment within your IBM Personal Computer XT/370. For details see "Logon Sequence" on page 4-12.

Remote 3277 Session

The Remote 3277 Session allows you to connect your IBM Personal Computer XT/370 to a remote computer using the &anr . With this session active, you can communicate with any remote computer which supports 3277 model 2 data streams. The types of systems to which you may connect include VM, CICS, TSO, MVS, etc. Your IBM Personal Computer XT/370 will appear as a 3277-2 terminal to that system.

Note: This selection appears on the Session Selection Menu only if an operational 3277-2 Device Emulator Adapter is installed in your IBM Personal Computer XT/370. If an adapter is installed and the Remote 3277 Session does not appear on your Session Selection Menu, turn off your IBM Personal Computer XT/370, then turn it on again and restart VM/PC. If the problem persists, call your technical

coordinator or run the diagnostics described in the diagnostic testing section of the *IBM Personal Computer XT/370 Guide to Operations.*

If you select this session, VM/PC will display the Remote 3277 Session screen, where you will see your normal remote system logo. Logon and processing procedures are the same as your current procedures. The keyboard will be configured exactly the same as the Local 3277 Session described above.

If you happen to be running a long running remote application and wish to do work on your VM/PC system at the same time, all you have to do is press the **SELECT** key and the VM/PC Session Menu will be displayed again. You can then select the Local 3277 Session if you wish, and logon and IPL your local VM/PC (you could have also done this before starting your Remote 3277 Session). You can now work on your local session at the same time that you are doing work on your remote session.

If the remote computer is a VM system with the VMPCSERV program installed, you can use the remote services as described in Chapter 6, "Remote Services" on page 6-1.

Remote 3101 Session

You can use the Remote 3101 Session to communicate with other computers over an RS232 interface without leaving the VM/PC environment.

This is possible because this session allows you to run the IBM 3101 Emulation Program (6024042), which provides four main functions:

- It allows you to use your Personal Computer XT/370 as if it were an IBM 3101 Display Terminal. This is called "3101 emulation."

- It allows you to adapt the specifications of your emulated 3101 to match what the remote computer (the computer at the other end of the communications line) expects to find. This is done using a separate DOS file to contain the specifications for each type of connection

you wish to make. Files containing specifications for seven popular connections are provided with the IBM 3101 Emulation Program

- It transmits ASCII format files between your Personal Computer DOS environment and the remote computer.

- To complement the file transfer capability, it provides a file conversion utility that will convert a binary format file to ASCII format or an ASCII format file to binary format.

The Remote 3101 Session provides an alternate way of communicating with computers that can or cannot be accessed with the Remote 3277 Session facilities.

If you intend to use the Remote 3101 Session, you should read the documentation supplied with the IBM 3101 Emulation Program.

Note: This selection appears on the Session Selection Menu only if an Asynchronous Communication Adapter is installed on your IBM Personal Computer XT/370.

Prerequisites

To use the Remote 3101 Session, you need to have the Asynchronous Communications Adapter installed in your Personal Computer XT/370. You also need to install the IBM 3101 Emulation Program, which is not supplied with VM/PC.

Note: The names of the programs supplied with the IBM 3101 Emulation Program are used by VM/PC to call those programs when the Remote 3101 Session is active. If you ever change the names of the IBM 3101 Emulation Program programs, you MUST update the VM/PC Configuration File to reflect all name changes. If you fail to do so, VM/PC will be unable to find the programs.

Using the Remote 3101 Session

The Remote 3101 Session presents you with the following menu screen:

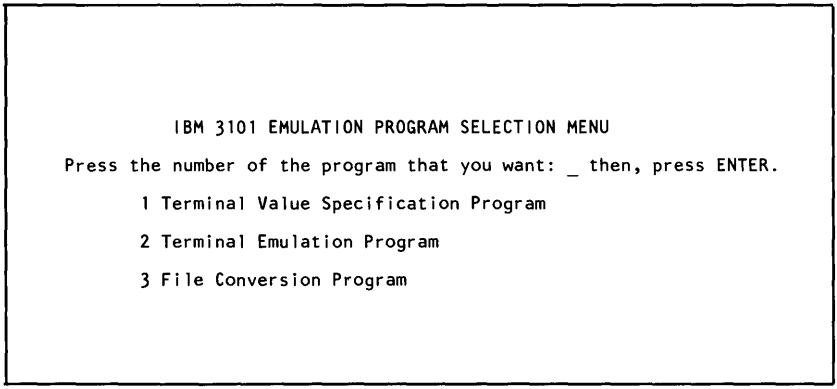


Figure 4-4. 3101 Emulation Program Selection Menu

The Terminal Value Specification Program is used to select, modify or create a specification file for use by the Terminal Emulation Program. Press the "1" key to run the Terminal Value Specification Program. VM/PC will then load this program (using the default DOS file name of C:SETUP.EXE) and give it control of the Remote 3101 Session.

The Terminal Emulation Program is used to actually communicate with another computer over the RS232 communications interface. Press the "2" key to run the Terminal Emulation Program. VM/PC will then load this program (using the default DOS file name of C:TERMINAL.EXE) and give it control of the Remote 3101 Session.

The File Conversion Program is used to convert ASCII format files to or from binary format files. The files involved in the conversion must be regular DOS files, but they can be on the fixed disk(s) or on diskette(s). Press the "3" key to run the File Conversion Program. VM/PC will then load this program (using the default DOS file name of C:FILECONV.EXE) and give it control of the Remote 3101 Session.

Once any of the three programs begins to run, its operation is as described in the IBM 3101 Emulation Program documentation.

When using the Remote 3101 Session, the keys on your keyboard behave as the IBM 3101 Emulation Program expects to find them. The one exception to this is the SELECT (Scroll Lock) key, which remains your means of switching to other VM/PC sessions.

You may use the IBM 3101 Emulation Program key definition menu to re-define the keys on your keyboard, but DO NOT re-define the Scroll Lock key. Your key re-definitions will not affect the other VM/PC sessions.

It is possible to use the Remote 3101 Session and all the other VM/PC sessions concurrently. This means that you may change active sessions and then return to the Remote 3101 Session right where you left it. Also, if you have begun an activity on another session (for example, a compilation on the Local 3277 Session), you can change to the Remote 3101 Session while the activity you started continues to execute. You should not, however, attempt to start a file transfer from the Remote 3101 Session while the Remote 3277 Session is also sending/receiving a file (using VM/PC Remote Services). In that event, some loss of data may occur in the Remote 3101 Session, making it necessary to restart the file transfer. This restriction is due to the timing dependencies inherent in asynchronous communications.

Since the IBM 3101 Emulation Program makes use of line 25 of your display screen, the indicators normally seen on that line while using other VM/PC sessions are not available when using the Remote 3101 Session. Normal VM/PC use of the 25th line is described in "The 25th Line" on page 4-18.

For related reasons, when you change your active session from the Remote 3101 Session to another session, any program you have started in the Remote 3101 Session (unlike other VM/PC sessions) is forced by VM/PC to wait until you return to the Remote 3101 Session. As a result, once you begin an activity (for example, a file transfer) using the Remote 3101 Session, you should not change your active session until the activity completes.

The Remote Server facilities described in Chapter 6 are not available using the Remote 3101 Session.

370 Processor Control

The 370 Processor Control Session is intended to be used by a system programmer to debug 370 programs. This session will appear on your Session Selection Menu only if it was specified during the configuration process described in Chapter 3, "Configuring Your VM/PC System" on page 3-1. For details on the use of this session refer to Appendix B, "The 370 Processor Control Session" on page B-1.

Logon Sequence

If you select the local session, you see the VM/PC Logon Screen which looks like this:

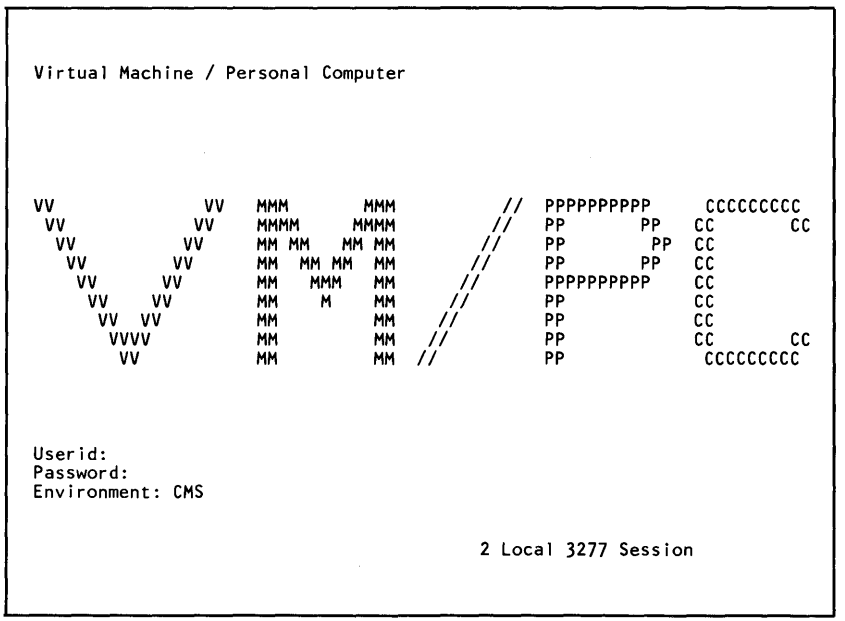


Figure 4-5. VM/PC Logon Screen

You must now identify yourself to the local session. This process is called 'logon'. During the VM/PC configuration process, described in Chapter 3, you or your technical coordinator specified special identification words to allow you, and only you, access to certain data. These words are called USERID and PASSWORD. The USERID can be any word 1 to 8 characters long, your name for instance. The PASSWORD can also be any 1 to 8 character word. These identification words work as a pair, that is, every USERID has a unique PASSWORD associated with it. You must give the computer both of these words to complete the logon process

NOTE: You or your technical coordinator may have specified during the configuration process that you want to

use the Autolog ID option and you do not want to use USERIDs and PASSWORDs. If this is the case you will not see the logon screen and should ignore this section.

The logon screen has a flashing cursor after the word **Userid:**. Type your USERID there.

After you type the USERID, press the NL key. The cursor moves to the next line after the word **Password:**. Type your PASSWORD there. Your password is secret so whatever you type will not be displayed. Don't be alarmed.

When you have typed both your USERID and your PASSWORD, press the ENTER key. If you have entered your USERID and PASSWORD correctly and have not changed the **Environment: CMS** field, the computer will do some processing and display the word **Ready;** which indicates the local session is active and you may skip the remainder of this section.

If you did not type a USERID, and one is required, or the one you entered cannot be found by the computer, the alarm will sound and an error message will be displayed. This message will inform you that the USERID you entered is not valid and position the cursor for you to enter it again.

If you did not type a PASSWORD, and one is required, or the PASSWORD you typed does not match your USERID, the alarm will sound and a message will be displayed. The message will inform you that a PASSWORD is required or the one you entered is not valid and position the cursor for you to enter it again.

The third line on the Logon Screen is **Environment: CMS**. If you press the ENTER key without changing the word CMS, the CMS environment will be started automatically. CMS stands for Conversational Monitor System. CMS is part of the VM/PC system. It provides a variety of tools to help you do your job. If you should change the word CMS to anything else, the computer will search its files to find the environment you selected. Do not change the word CMS unless you are told to do so.

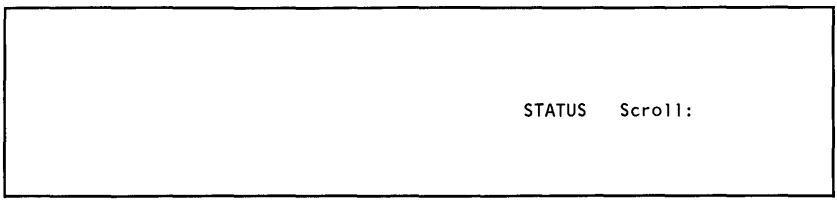
If you should change the word CMS, the computer may give you a message that the environment you selected does not exist. If you see such a message type **ipl cms** and press the ENTER key. The computer will do some processing and display the word **Ready;** which indicates the CMS environment has been started and you may continue.

NOTE: If you have not run the Configuration program, you may logon to VM/PC and run the Local 3277 Session. A default logon configuration was provided for you during the installation process. This configuration has a USERID of 'SAMPLE' and a PASSWORD of 'SAMPLE'. It provides you with an A-disk and access to the system files. The first time you use this logon, you will receive an error message 'Disk'A (101)' has invalid directory format'. Don't be alarmed. Enter the command **FORMAT 101 A (N** and reply **YES** to the 'Do you wish to continue?' question.

Screen Format

The IBM Personal Computer Display has 25 lines of 80 characters each. Once you have established a local VM/PC session or a remote 3277 session, the first 24 lines will look exactly like a 3277 display screen.

The 23rd and 24th lines are the place on the screen where you enter commands which allow you to tell the computer what you want it to do. This place is called the input area. The kinds of commands you may use are described in Chapters 5 thru 7. The 24th line looks like this:



The end of the 24th line is reserved for screen scroll commands and status notices.

Note: You can enter commands in both the input area and the scroll area. When you press ENTER, both commands will be executed.

Screen Scroll

The Screen Scroll facility allows you to look at data which was on your screen but has been erased. All transactions to and from the terminal (write output and read input) are saved in a terminal history buffer. As this information is scrolled off the display by the CLEAR key, PA2 key, or the Diagnose 58 erase option, it is not lost but placed in the terminal history buffer.

The history buffer may be viewed any time the Local 3277 Session is active. The screen scroll commands are entered in the **Scroll:** area at the end of the 24th line. The screen scroll commands are:

- **F[nn]** - scrolls the history view so that more recent information is displayed (forward in time). The option

nn, indicates the number of lines to be scrolled into the bottom of the screen (the bottom of the screen contains the most recent history item within the view). If nn is not specified, a full screen is scrolled forward (the item on the bottom of the window is moved to the top).

- **B[nn]** - scrolls the history view so that less recent information is displayed (backward in time). The option nn, indicates the number of lines to be scrolled into the top of the screen (the top of the screen contains the least recent history item within the view). If nn is not specified, a full screen is scrolled backward (the item on top of the window is moved to the bottom).

- **TOP** - moves the window to the least recent items in the history buffer.

- **BOT** - restores the window to the display of the current activity.

- **R[nn]** - scrolls the history viewing window to the right so that information lines longer than the display size can be displayed (this option is enabled only if TERM WRAP OFF is active).

- **L[nn]** - scrolls the history viewing window to the left (this option is enabled only if TERM WRAP OFF is active).

The Screen Scroll facility can be used to look at data which has been cleared from the screen. It can also be used with the RETRIEVE facility of the CP SET command to retrieve commands from the history buffer and move them to the input area. Commands can then be executed by pressing the ENTER key or modified before execution. See the CP SET command for more information.

Status Notices

To the left of the **Scroll:** area, VM/PC will display various STATUS notices. They will tell you what is happening in the computer at the present time. These notices are:

- **CP Read** - This notice appears if you logon and do not automatically start up the CMS environment or press the PA1 key. It means a part of VM/PC called the Control Program is waiting for you to enter any CP command described in Chapter 7, such as the **ipl cms** command.

- **Waiting** - This notice means a part of VM/PC called CMS is waiting for you to enter any CMS command described in Chapter 7. The computer is idle whenever this notice is displayed.

- **VM Read** - This notice means some part of the system is waiting for you to enter some data. It will usually be preceded by a message asking for the data.

- **Running** - This notice means the computer is working on something and you should not enter any commands.

- **More...** - This notice means that VM/PC has more information than can fit on one screen. After 20 seconds the alarm is sounded and after another 10 seconds, VM/PC displays the next screen.

  To see the next screen right away, press the CLEAR or the PA2 key. To hold this information on the screen, press the ENTER key. This changes the **More...** notice to **Holding**.

- **Holding** - This notice appears when the screen displayed a **More...** notice and you pressed the ENTER key. The screen will remain the same until you press the CLEAR or PA2 key.

  Pressing the ENTER key changes the **Holding** notice to **More....**

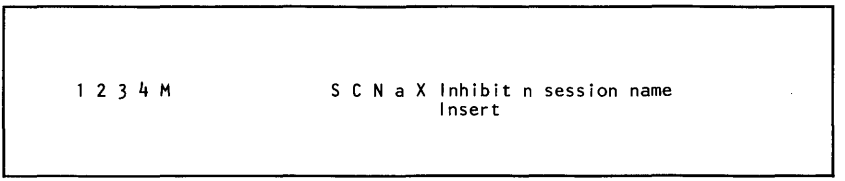
- **Not Accepted** - This notice appears if you enter a command when the screen displayed **More...** or a **Holding** notice. The command you entered is not accepted by the computer. **Not Accepted** will remain on the screen for about three seconds.

If you press the CLEAR key, the command you entered is ignored and must be entered again. If you press the PA2 key, your command will remain in the input area after the screen is cleared.

The 25th Line

The 25th line gives you information on the status of other sessions, the state of certain keyboard functions, the state of your session connection, and a reminder of the name and number of the active session.

The indicators on the 25th line are arranged as follows:



where:

- **1 2 3 4** - These numbers indicate activity has taken place on sessions which have been established but are not active. To see the activity, press the SELECT key and the number indicated. When you look at the session which has been changed, the number is turned off.

- **M** - The M indicates activity has taken place on the Session Selection Menu screen. To see the activity, press the SELECT key. When you look at the menu that has been changed, the indicator is turned off.

- **S** - The **S** indicates the SHIFT key is depressed and all keys pressed will cause the uppercase character to be entered. The **S** will be turned off when the SHIFT key is released.

- **C** - The **C** indicates the CAPS LOCK key was pressed and all alpha characters will be entered as uppercase characters. Pressing the CAPS LOCK key a second time will cancel the function and turn off the **C**.

If the SHIFT key is pressed while the CAPS LOCK indicator is on, the keys will reverse states, that is, all alpha keys will be entered as lower case and all other keys will be entered as uppercase.

- **N** - The **N** indicates the NUM LOCK key was pressed and the numeric keypad is active. Pressing the NUM LOCK a second time will cancel the function and turn off the **N**.

- **a** - The **a** indicator is an arrow indicating the direction of the cursor movement. If no arrow is displayed, the cursor movement is standard left to right.

  You can select the direction you want the cursor to move by pressing and holding the ALT key, pressing one of the cursor control keys as listed below, and releasing both keys. The **a** indicator will display the direction of cursor movement. Press the RESET key to return the cursor direction to normal (Right).

  - LEFT - Press the ALT and ← cursor control keys. The **a** indicator will display an arrow pointing to the left and the cursor will move one position to the left whenever a keystroke is entered.

  - RIGHT - Press the ALT and → cursor control keys. The **a** indicator will not be displayed. The cursor will move in a standard manner, one position to the right whenever a keystroke is entered.

  - UP - Press the ALT and ↑ cursor control keys. The **a** indicator will display an arrow pointing up and the cursor will move one line up whenever a keystroke is entered.

  - DOWN - Press the ALT and ↓ cursor control keys. The **a** indicator will display an arrow pointing down and the cursor will move one line down whenever a keystroke is entered.

- **X** - The **X** indicates the computer associated with the active session is not available. This indicator will come on for short periods of time whenever the ENTER key is pressed. Don't be alarmed.

  If the indicator remains on for a long period and no activity appears to be taking place, there may be a problem with the computer. Please refer to 'What To Do if Something Goes Wrong' later in this chapter.

- **Inhibit/Insert** - The **Inhibit** indicator is controlled by the computer associated with the active session. If the word **Inhibit** is displayed, the keyboard is locked and input is inhibited. This indicator is normally on when the **X** is also on.

  The word **Insert** indicates the INS key was pressed and the keyboard is set to insert mode. All data will be entered at the position of the cursor and all data to the right will be moved to the right. Press INSERT again, or the RESET key (or any key that causes communication with the host system - ENTER, PA keys, PF keys, etc.) to turn off the Insert indicator.

  Note: VM/PC 3277 emulation does not require you to set nulls on before you can insert characters into a line.

- **n session name** - The **n session name** indicator is the number and name of the active session. The number and name displayed are the same as the number and name on the Session Selection Menu.

What To Do if Something Goes Wrong

System Failures

System failures are errors detected in the IBM Personal Computer XT/370 hardware or in the VM/PC software. They can be detected during the VM/PC startup or at any time during normal VM/PC processing. Whenever a system failure is detected, you will be notified in two ways.

- The Session Selection Menu will be updated to indicate an error on the Local 3277 Session. The Session Selection Menu will instruct you to select the Local 3277 Session for more information. When you select the Local 3277 Session, you will see the message listed below.

- When the Local 3277 Session is active, the following message will be shown:

```
Local 3277 Session Terminated (or Initialization Failure)
    Press F1 for retry, SELECT to return to Session Menu
XXXXXXXXXXXXXXXXXXXXXXXXX
```

where XXXXXXXXXXXXXXXXXXXXXXXXX is an error message.

If you receive this message, press the F1 key to attempt a retry. If the retry is successful, the local session will be restarted. Your last entries may be lost and may be entered again. Any remote sessions which were active will not be affected.

If the retry is unsuccessful, you must take some further action. Refer to the procedures below or call your technical coordinator and tell him/her the error message you received.

Startup Failures

During the start up period, VM/PC can detect and report the following types of failures:

- STORAGE ERROR, See *VM/PC Users Guide* RC = 2670

  A storage error was detected, press the F1 key to attempt a retry. If the retry is successful, the problem may be intermittent. Record the numbered code (RC) for reference and continue processing.

  If the retry is unsuccessful, call your technical coordinator or run the diagnostics described in the 'Diagnostic Testing' section of the *PC XT/370 Guide to Operations*.

- PROCESSING UNIT ERROR, See *VM/PC Users Guide* RC = 2680

  A processing unit error was detected, press the F1 key to attempt a retry. The recovery actions are the same as for the STORAGE ERROR described above.

- HARDWARE ERROR, See *VM/PC Users Guide* RC = 2681

  An undetermined hardware error was detected, press the F1 key to attempt a retry. The recovery actions are the same as for the STORAGE ERROR described above.

- nnnnnnn File Not Found, See *VM/PC Users Guide* RC = 2695

  A file that is necessary to complete the start up process could not be found. nnnnnnn is the file name. It must be reloaded from the distribution diskette. Notify your technical coordinator or refer to the procedures in Chapter 2 (or Chapter 3 if you have recently reconfigured your system).

- nnnnnnn File Bad, See *VM/PC Users Guide* RC = 2696

A file that is necessary to complete the start up process could not be read. nnnnnnn is the file name. It must be reloaded from the distribution diskette. Notify your technical coordinator or refer to the procedures in Chapter 2 (or Chapter 3 if you have recently reconfigured your system).

Processing Failures

During processing, VM/PC can detect and report many types of hardware and software failures which prevent it from continuing. These messages and the recommended recovery actions are described in Chapter 8, "VM/PC Messages" on page 8-1 of this book.

System Error Conditions

System error conditions are errors caused by external conditions which may or may not be detected by the IBM Personal Computer XT/370 or VM/PC. They can occur at any time during normal processing.

Error Messages

During processing on the Local 3277 Session, error messages may be displayed to you. Most of these messages will be easy to understand and you should be able to continue with no trouble. If any message does not make sense to you, refer to Chapter 8, "VM/PC Messages" on page 8-1 for a description of what happened to cause the message.

System Unavailable

During processing on the Local or Remote sessions, the **X** indicator on the 25th line is on whenever contact is lost with the computer with which you are in session. This indicator, System Unavailable, comes on whenever the ENTER key is pressed but should be turned off in 1 to 3 seconds.

If the indicator remains on for an unusually long period of time you can assume something has gone wrong. The action you should take depends on the type of session:

- Remote sessions usually indicate a problem at the remote computer. Call your technical coordinator and determine if the remote computer is operational. If the computer is operational, press the SELECT key.

  - If the screen does not change, the problem is in the Personal Computer hardware. Call your technical coordinator or run the diagnostics described in the 'Diagnostic Testing' section of the *PC XT/370 Guide to Operations*.

  - If the Session Selection Menu appears, the problem is in the connection with the remote computer. Call your technical coordinator and report the connection failure.

- Local sessions usually indicate an undetected problem with the hardware. Press the SELECT key.

  - If the screen does not change, the problem is in the Personal Computer hardware and the Personal Computer diagnostics should be executed to pinpoint the cause.

  - If the Session Selection Menu appears, the problem is in the 370 hardware and the IBM Personal Computer XT/370 diagnostics should be run to pinpoint the cause.

Chapter 5. Using VM/PC Commands to Do Work

Using Commands to Do Work

In this section, we'll tell you how to use commands to do your work and we'll give you examples of how to use them. We've grouped the commands under task headings. Here's a list of the task headings and an indication of who should read each section.

1. Creating, Changing and Managing Files - All VM/PC users should read this section.

2. Printing Files - All VM/PC users should read this section.

3. Writing EXECs - All VM/PC users who will be writing EXECs to help them do their work should read this section.

4. Setting Up Your System to Meet Your Needs - All VM/PC users should read this section. Those who plan to have concurrent Local and Remote sessions should read the examples of how to set up their system.

5. Transferring Personal Computer DOS and VM/PC Files - Everyone who has Personal Computer DOS files that they would like to use with VM/PC or who would like to use the files they create in VM/PC on a Personal Computer DOS system should read this section.

Creating, Changing and Managing Files

The VM/PC Editor, referred to by the command that calls it, XEDIT, provides a wide range of useful facilities to help you create, change, and manage your files.

More information on the XEDIT commands discussed in this section can be found in the *VM/SP System Product Editor User's Guide* and the *VM/SP System Product Editor Command and Macro Reference.*

Creating Files

What Is a File?

Information stored in the system is organized in collections or groups of data called files. You organize the information that you store in the system in your own personal file collection. You may choose to store a collection of short stories on the system by putting each story in its own file. Or a collection of memos to Mr. Smith may be stored in a file to keep them separate from the file containing memos to Mr. Jones. The system provides you with programs and information that will help you choose a way to organize your own personal file collection.

You will need a way to identify each file so that later, when you want to use it again, you will be able to tell the system which information to give you.

You can identify a VM/PC file by giving it its own name. The three parts of a file identifier are filename (fn), filetype (ft), and filemode (fm).

The filemode identifies the collection to which the file belongs, as well as giving it certain attributes. Most data files that you create will have a fm of A1. The "A" identifies the file as part of your personal file collection, and the "1" means that the file will be visible to other users with "read access" to your personal file collection. A1 is assumed by the system if you do not specify fm when you issue file

manipulation commands. Be sure that you understand the system action and purpose for filemode before you specify a mode other than A1.

Filetype specifies the type of data contained in the file. VM/PC recognizes some specific filetypes that give the file contents special meaning. Files containing programs have a filetype that identifies the source language interpreter or compiler. For example, EXEC/EXEC2, FORTRAN, COBOL, and ASSEMBLER language programs are stored in files with EXEC, FORT, COBOL, and ASSEMBLE filetypes respectively.

You can specify any filetype that you choose for files containing data. There are, however, certain programs that require their input data to be stored in files with a specific filetype. SCRIPT is an example of such a program -- it requires data to be in a file with a filetype of SCRIPT.

Finally, the filename is the least restrictive of the three elements. A filename may be up to eight numbers and letters, in any combination that you choose. You will find that filenames that help identify the file contents are the most useful. When used in combination with the 8-character filetype, the fn ft can readily identify a specific file or group of files according to content.

In the examples that follow, files containing similar types of data have the same filetype. For example, you can name files containing monthly reports JAN84 REPORT, FEB84 REPORT, MAR84 REPORT, etc. You can choose a filetype of CALENDAR for files containing daily calendar information. 112383 CALENDAR, for example, might contain daily calendar information for November 23, 1983. MEMO TOJOE would be a good name for a file containing a message for Joe until you find it necessary to distinguish one message to Joe from another. You will discover more uses for grouping filetypes later, when you read about other file manipulation commands.

When you type commands, be sure to type them as they are shown in this book, including spaces if there are any. If a

space is missing or a word is misspelled, the system won't recognize the command. Commands can be typed in capital or lowercase letters, or a combination of the two.

When we show you the format of a command, we will print everything that must be typed exactly as shown in capital letters. When you have to substitute your own information, we'll show the word in all lowercase letters. For example, a command format is:

```
XEDIT filename filetype filemode
```

If a word is shown in capital letters, it must be typed exactly as shown, although you can use capital or lowercase letters. Because the words filename, filetype, and filemode are shown in all lowercase letters, you would substitute your own three-part file label in their places.

If command names can be abbreviated, we will show the minimum abbreviation in capital letters. For example,

```
Input
```

You could enter the command as I, IN, INP, INPU, or INPUT. The capital letter (I) shows the minimum abbreviation.

Creating A File

Use the XEDIT command to create a file. Its format is:

```
XEDIT filename filetype filemode
```

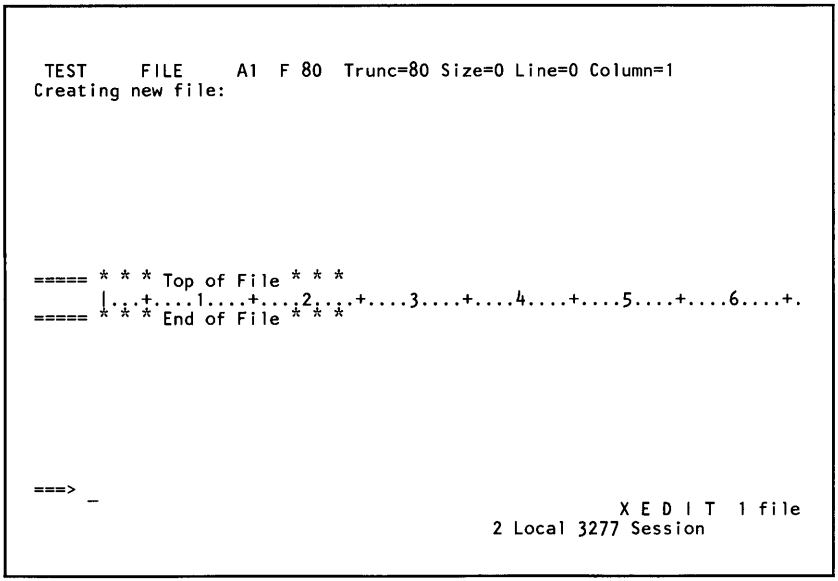
After you type the command, press the ENTER key.

Note: The ENTER key on your keyboard is the long key on the right side that has the bent arrow on it (←┘).

If you entered the command:

```
xedit test file a1
```

your screen would look like this:



Here's an explanation of the various information on the screen:

1 File Identification Line
The first line has the three-part file label - filename filetype filemode. F 80 means that the length of a line is fixed and the file will hold lines up to 80 characters long. TRUNC means "truncation column." TRUNC=80 means that if you type more than 80 characters in a line, the additional characters are rejected by the system. SIZE=0 is the number of lines in the file. LINE=0 is the file line number of the current line, which is explained in number 7, below.

2 Message Line
The editor communicates with you by displaying messages on the second line of the screen. These messages might tell you that you made an error in a command, or they might provide information.

3 Large Arrow Command Line
The large arrow (===>) at lower left of the screen points to the XEDIT command line. One of the ways you communicate with the editor is by entering XEDIT

commands on this line. XEDIT commands can be typed in either uppercase or lowercase, or a combination of both, and many can be abbreviated.

After typing a command on this line, **you must press the ENTER key** to send the command to the editor. When the ENTER key is pressed, the command disappears from the command line and is "executed," or carried out.

4 Status Area
The lower right corner displays the current status of your editing session, for example, edit mode or input mode. It also shows the number of files you are editing. The status area in this screen shows that one file is being edited.

5 Prefix Area
The prefix area is the five equals signs (=====) that appear to the left of each line in the file. You can perform various editing tasks on a line by entering "prefix commands" in the prefix area of that line. We'll discuss prefix commands later.

6 File Area
The rest of the screen is available to display the lines in the file. The file lines are inserted between the ***TOP OF FILE*** and ***END OF FILE*** notices you now see on the screen.

7 Current Line
The current line is the file line in the middle of the screen (above the scale). It appears brighter than the other file lines or is "highlighted." In this screen, the current line is the TOP OF FILE line.

The current line is important because most commands you type in the command line perform their functions starting with the current line. Naturally, the line that is current will change as you move up and down in the file. Many XEDIT commands perform their functions starting with the current line, and change which line is current when they are finished.

8 Scale
The scale appears under the current line to assist you in editing. It's like the margin scale on a typewriter.

9 VM/PC Information Line
The last line on the screen is shown in "reverse video." VM/PC uses this line to provide information to you. See "The 25th Line" on page 4-18 for a complete description of the information provided on this line.

Entering Data

One way to enter data in a file is to use **input mode.** The INPUT command puts you input mode. Its format is:

```
Input
```

In input mode, the prefix areas (=====) disappear. The message line and status area display the notice INPUT MODE. You type your data in the **input zone.** The input zone is the bottom half of the screen, between the scale and the command line.

The command line contains the notice ***INPUT ZONE***. It contains this notice because **you can't enter commands** from the command line while using input mode. The notice marks the end of the input zone and reminds you that you cannot enter commands. You have to go back to edit mode to enter commands. We'll show you how later.

The cursor is placed automatically on the first line in the input zone. This is where you should begin typing the data. After you've typed each line, press the new-line key to return the cursor to the beginning of the next line. You could use one of the other cursor-movement keys, but the new-line key is faster. (The new-line key is a wide key at the upper right of your keyboard; it looks like this, ←—).

The lines you typed move to the top half of the screen. The last line you typed is the new current line. After you pressed the ENTER key, the editor changed all the letters you typed to capitals. Later, you'll learn how to tell the editor not to change them to capitals. If you had more data to type, you could continue typing in the input zone (between the current line and the command line).

If you have no more data to type, press the ENTER key again. This takes you out of input mode and back into edit mode. The screen returns to edit mode layout, and the file contains the data. The file identification line changes to indicate the number of lines in the file and the number of the current line.

You can enter input mode at any time to insert new lines of data in the file by using the INPUT command. Press the ENTER key twice to return to edit mode.

When you're finished working with a file, you use the FILE command if you want to keep it. After you enter the FILE command, you'll see a ready message (Ready;) It tells you that this editing session is over, and VM/PC is ready to do more work for you. Notice that the cursor returns to the command line. That's where you'd type the command to create another file.

Note: Remember to press the ENTER key after you type a command on the command line. That sends the command to VM/PC for processing.

Capital and Lowercase Letters

If you want the letters in your file to be entered in the file exactly as you typed them, use the SET CASE command. This tells the editor to use mixed case, that is, both capital and lowercase letters, when the data is entered in the file. Its format is:

```
SET CASE Mixed
```

If you fill up the input zone but want to continue typing data in input mode, press the ENTER key one time. The data moves up on the screen, and you can continue typing in the cleared input zone. When the input zone is filled up again, press the ENTER key once again, etc. You can stay in input mode and continue typing as long as you need to. When you're ready to return to edit mode, then press ENTER twice.

Splitting and Joining Lines

To split or join lines, you can press a PF key instead of typing a command in the command line and then pressing the ENTER key. During an editing session, each PF key is set to an XEDIT command. You don't press the ENTER key when you use PF keys. The command is sent to the editor automatically when you press the PF key.

Important Notes:

1. On your keyboard, the function keys are labelled F1, F2, etc. On other keyboards, they're labelled PF1, PF2, etc. When we use the term PF key we mean the F keys on the left of your keyboard.

2. In this book, we will be talking about PF keys 1 through 12. Your terminal has 10 F keys. You get PF 11 by pressing the uppercase key with F1. You get PF 12 by pressing the uppercase key with F2. We'll refer to them as PF 11 and PF 12, but remember that you must press two keys to get them on your keyboard.

To split a line in two:

- Move the **cursor** under the character where you want the line to be split.

- Then **press PF10.**

The cursor remains where it was before the line was split.

You may want to split a line if you need to add information to a line. The cursor is in place, and you can continue typing on that line.

To join two short lines:

- Place the **cursor** in the first line where you want the second line to go.
- Then **press PF11.**

Setting Tabs

Sometimes you may want to place information in certain columns. The PF4 key acts like a tab key on a typewriter. Each time PF4 is pressed, the cursor moves to the next tab column. Initial tab settings are defined by the editor. You can change these tab settings using a SET TABS command.

You can find out what the tab settings are with the QUERY TABS command. Its format is:

```
Query TABS
```

The tab settings are displayed on a cleared screen. Press the CLEAR key to get your file back on the screen.

You can change these tab settings with the SET TABS command. Its format is:

```
SET TABS n1 n2 n3 n4 etc.
```

You substitute the column numbers where you want the tabs set for n1, n2, and so forth. These settings remain in effect for this editing session only. The next time you edit this file, you must enter the SET TABS command. Otherwise, the initial tab settings will be in effect.

Displaying PF Key Settings

Use the QUERY PF command to find out what the PF keys are set to. Its format is:

```
===> Query PF
```

The XEDIT commands assigned to all the PF keys are displayed, on a cleared screen. Press the CLEAR key to get your file back on the screen.

Use the QUERY PF command with the PF key number to find out the setting for a particular PF key. Its format is:

```
===> Query PF10
```

The PF key setting is displayed in the message line.

Inserting Words in a Line

One way to insert letters, words, or spaces in a line is as follows:

- Press the **Ins key.**
- You enter insert mode, and you can type the characters to be inserted. The word "Insert" is highlighted near the middle of the bottom line on your screen.
- Press the **RESET** key to type normally again.

When characters are inserted in a line, the other letters move over to make room for them.

Note: Remember each time you need to insert characters in a different line, you must press **Ins** in order to use insert mode. After you're finished inserting characters in that line, press **PrtSc**. You can use insert mode in both edit and input modes.

Power Typing

Power typing is another kind of input mode. The advantage of using power typing is that you can type continuously, as if the screen were one long line. You do not have to be concerned with line length or word length. You can start typing a word on one line of the screen and finish it on the next. In fact, if you're a skilled typist, you don't even have to look at the screen. When you reach the end of a line, the cursor automatically moves to the beginning of the next line. It's like having an automatic carriage return on a typewriter.

To enter power typing mode, enter the POWER command. Its format is:

```
POWer
```

If you want a sentence to begin on a new line, you must precede it with a pound/number sign (#). It will not appear in your file. It simply tells the editor to start that text on a new line. If you run out of space on the screen, press the

ENTER key, and more typing room will be provided for you. When you want to stop power typing, press the ENTER key twice.

You can use power typing mode at any time during an editing session by entering the POWER command. The data entered using power typing is inserted after the current line, as it is when you use the INPUT command. When you enter POWER, the current line of your file is displayed as the top line on the power typing screen, but you cannot type over it.

If you want to insert characters or spaces in a line while you're using power typing, press the Ins key. Remember to press the RESET key when you are finished using insert mode.

Creating an XEDIT Profile

You can use various SET commands to tailor an editing session to meet your needs. For example, if you want the cursor to tab to columns 10 and 20, you enter the command SET TABS 10 20. If you want an automatic SAVE command issued periodically, you use a SET AUTOSAVE command. If you want your data to appear in capital and lowercase letters, you enter SET CASE MIXED.

You can put all these commands (and others) in an XEDIT PROFILE. Its filename must be PROFILE and its filetype must be XEDIT. To create this file, enter XEDIT PROFILE XEDIT. Type the commands you want, and enter a FILE command. Then, all the commands in the PROFILE XEDIT file will be executed automatically each time you enter an XEDIT command. This saves you the time it takes to type these commands each time you edit a file.

A sample PROFILE XEDIT file might contain these commands. Each command must be typed on a separate line.

```
SET CASE MIXED
SET AUTOSAVE 20
SET TABS 10 20 30 40
SET PF12 FILE
```

You should be familiar with all the commands except the last. You already know that when you edit a file, the PF keys have various functions. If you would rather have one or more PF keys set to different commands, you can do this in the PROFILE XEDIT file. The command has the format:

```
SET PFn command
```

You substitute the number of the PF key for "n," and you substitute the command you want assigned to that PF key for "command." In our example, we have assigned the FILE command to PF12. After this file is put on disk, pressing PF12 would do the same thing as entering the FILE command.

Changing Files

This section discusses the commands that display a list of your files, add new lines, take out lines, duplicate, move, and copy lines, locate and change data in a file, combine files, and sort data in a file.

Displaying a File on Your Screen

To work on a copy of a file that is on your disk, use the XEDIT command:

```
XEDIT filename filetype filemode
```

If the filemode is **a,** you don't have to type it.

A copy of the file appears on your screen, but the original is still on your disk. It remains unchanged until you enter a FILE or SAVE command.

Adding Lines

One way to add a line of information is by using the INPUT command. Another way is to use a **prefix command** that adds a blank line. It is called a prefix command because you type it in the prefix area, not on the large arrow command line. Once the blank line is added, you type the new data in that line.

You type a prefix command over any of the five equal signs in the prefix area. You can use either capital or lowercase letters. Then press the ENTER key to carry out the command.

The prefix command that adds a blank line after the one in which it is entered is **a.** You can type a number before or after the **a** to add more than one line. The following are all valid ways to type this command:

```
====A   Adds one blank line after this line.
a====   Adds one blank line after this line.
10a==   Adds ten blank lines after this line.
===A5   Adds five blank lines after this line.
```

There are several prefix commands. You'll learn some of them in this section. If you type a letter that is not one of these commands and press ENTER, the prefix area fills up with question marks and the message line displays "Invalid Prefix Subcommand." To correct the error, you could press the ERASE EOF key to clear the prefix area. Then type the correct prefix command.

Moving through a File

As your files grow, sooner or later they won't fit on one screen. To look at each screen of the file, you use commands that "scroll" the file. Scrolling is like turning the pages of a book.

Moving Forward or Backward: Two PF keys are set to commands that scroll the file. PF8 scrolls forward one screen. PF7 scrolls backward one screen.

Moving to the Bottom or Top: Use the BOTTOM command to go forward to the end of the file. The BOTTOM command makes the last line of the file the new current line. **To move the cursor to the large arrow command line from any place on the screen, just press the ENTER key.**

The TOP command makes the TOP OF FILE line the new current line. You might want to use this command when you need to add a line at the beginning of the file.

Moving Down or Up: Use the DOWN command to move the display one or more lines toward the end of a file and the UP command to move the display toward the top of the file. Their format is:

```
Down n      or        Up n
```

You substitute the number of lines you want to move down for "n." Both these commands change which line in the file is the current line on your screen.

Another way to move up or down in a file is to use the / (slash) prefix command. You can type a / in the prefix area of any line on the screen. When you press the ENTER key, that line becomes the new current line. Then, if you enter an INPUT command from the command line, the new lines you type in input mode will be inserted between the current line and the line that followed it.

(Note: When the current line is somewhere in the middle of your file and you enter an INPUT command, the part of your file that was displayed in the bottom half of the screen disappears temporarily. It is **not** gone from the file. After you type your data in the input zone and return to edit mode, that part of the file will appear, after the lines you typed.)

You enter a / in any position of the prefix area. For example:

```
/====  or  ==/==  or  ====/
```

Deleting Lines

The **d prefix command** deletes the line in which it is entered. (Do not confuse it with the **Del key**, which deletes characters and spaces **within** a line.) You type a letter "d" in any position of the prefix area of that line. You can type a number before or after the "d" to delete a number of lines. The following are all valid ways to type a **d** prefix command.

```
====d Deletes this line.
d==== Deletes this line.
10d== Deletes this line and the next nine lines.
```

Can you press the ERASE EOF key to clear a line? Yes, but there are two things you should be aware of. First, you must move the cursor to column one in the line and press the spacebar once before pressing the ERASE EOF key. If you don't press the spacebar, the data will come back in the line the next time you press the ENTER key. This prevents you from erasing a line if you press the ERASE EOF key accidentally. Second, the data will be removed from the line, but the blank line will remain in the file. So it's better to use a **d** prefix command if you want to delete a line. If you want to delete only some of the characters in a line, instead of a whole line, you can use the delete key.

Deleting a Block of Lines: To delete a block of lines, type the letters "dd" in the first line of the block. Then type "dd" in the last line of the block. Then press the ENTER key. For example:

```
dd=== This is the first line I want to delete.
===== This is the second.
===== This is the third.
===== This is the fourth.
dd=== This is the last.
```

When you press the ENTER key, all of the lines are deleted.

A block of lines doesn't have to be all on one screen. You can type "dd" on the first line of the block and then scroll the display until you find the last line of the block. Then you can type the second "dd" and press the ENTER key. Use PF8 to scroll forward, PF7 to scroll backward.

If you type "dd" in a line and then press PF7 or PF8, the status area displays:

```
BLOCK INCOMPLETE
```

This reminds you that the editor is waiting for you to enter the second "dd," thus completing the block.

Cancelling A Deletion - RECOVER Command: Suppose you use a **d** prefix command to delete one or more lines, and then you change your mind. You want the lines back. You can get them back by using the RECOVER command. It is entered in the large arrow command line and has the format:

```
RECover n
```

You substitute the number of lines you want to recover for "n."

The recovered line(s) will appear immediately above the current line. You can put the lines back where they belong by using the **m** prefix command, discussed later.

Duplicating Lines

If you want to repeat a line in a file, you can use the " (double quotation mark) prefix command. If you want the line duplicated more than one time, type a number before or after the ".

Copying Lines

If you want to copy lines from one part of a file to another, you can use the **c prefix command.**

To copy one line, type a letter **c** in the prefix area of the line you want to copy. Then you have to tell the editor where you want it to be copied. You do this by typing either an **f** (for "following") or a **p** (for "preceding") in the prefix area of another line. When you press ENTER, a copy of the line where you typed the "c" is placed:

- Either after the line containing an "f"
- Or before a line containing a "p."

For example:

```
====c This is the line I want copied.
     .
     .
     .
====f I want it to be copied after this line.
```

When the ENTER key is pressed, file will look like this:

```
===== This is the line I want copied.
      .
      .
      .
===== I want it to be copied after this line.
===== This is the line I want copied.
```

Copying More Than One Line: You can type a number before or after the "c" to copy more than one line. The following are all valid ways to type the **c** prefix command.

```
====c   Copies this line.
=c5==   Copies this line and the next four lines.
10c==   Copies this line and the next nine lines.
```

You must indicate where the lines are to be copied with an **f** or **p** prefix command.

Copying a Block of Lines: To copy a block (a group of consecutive lines), type the letters "cc" in the prefix area of the first line of the block. Type "cc" in the prefix area of the last line of the block. Then press the ENTER key. This way, you don't need to count the number of lines you want copied. For example:

```
cc=== This is the first line I want to copy.
===== This is the second.
===== This is the third.
===== This is the fourth.
cc=== This is the last.
        .
        .
        .
p==== I want them to be copied before this line.
```

When the ENTER key is pressed, a copy of the block of lines will be placed before the line containing the "p." (If the line contained an "f," the block would go after that line.)

If the block of lines extends for more than one screen, you can type "cc" on the first line of the block and then scroll the file until you find the last line of the block. Then you can type the second "cc." You can also scroll the file before typing the "f" or "p."

While you scroll the file, the status area displays:

```
COPY/MOVE PENDING
```

The editor is waiting for you to enter an "f" (or "p").

```
BLOCK INCOMPLETE
```

The editor is waiting for you to enter the second "cc."

Moving Lines

Moving lines means to remove them from their original place and put them in another.

The procedure for moving lines is the same as for copying lines, except you use an **m prefix command** (instead of a **c** prefix command). As with copying, you must show the destination line with an "f" or a "p." You can move more than one line by typing a number before or after the "m."

You can move a block of lines the same way you would copy a block. Just type "mm" (instead of "cc") in the prefix area of both the first and last lines of the block. Indicate the destination by using "f" or "p."

Cancelling a Prefix Command - RESET Command

You know that scrolling the file while entering different parts of a prefix command causes the status area to display:

```
BLOCK INCOMPLETE   OR   COPY/MOVE PENDING
```

If you change your mind mid-stream, you can cancel the operation by entering this command (in the large arrow command line):

```
===> RESet
```

If you have typed any prefix commands (even those that do not cause the notices above to be displayed) but have not yet pressed ENTER, you can press the **CLEAR key** to remove them from the prefix area.

LOCATE and CHANGE Commands

You use a LOCATE command to locate the line that contains the text you want to change. Its format is:

```
Locate/your data
```

Note: You must type a separator between the command name (LOCATE) and your data. In the example above, a slash (/) is the separator. If the data to be located contains a slash, like VM/PC or 01/23/84, you must use a different special character as a separator. For example, LOCATE$VM/PC or LOCATE%01/23/84. The following special characters cannot be used as separators: ¬ (not), + (plus), - (minus), or . (period).

The LOCATE command tells the editor to look through the file, starting with the line following the current line, until it finds "your data."

You can change the located data using this CHANGE command:

```
Change/oldword/newword/
```

This form of the CHANGE command changes the first occurrence of the old word **in the current line.** This is why you used the LOCATE command. When the editor locates the data, it makes the line that contains it the new current line.

Changing a Word Throughout a File: To change data throughout a file, use this form of the CHANGE command:

```
===> c/olddata/newdata/ * *
```

The message x OCCURRENCE(S) CHANGED ON x LINE(S) tells you how many words were changed. The editor searched every line of the file, from the current line to the last, and changed "olddata" to "newdata." Note the format of this command. You must type a slash (/) after the new word (newdata), and you must type two asterisks **with a space between them** (* *). The first asterisk tells the editor to search every line of the file. The second tells it to change every occurrence of the old word in each line.

After a CHANGE command, the last line searched becomes the new current line.

Note: Remember that the change starts with the current line. If you enter this kind of command when the current

line is someplace in the middle of a file, the change is made until the editor reaches the end of the file. It doesn't go back to the beginning to make changes there.

Tips on Using LOCATE and CHANGE: The LOCATE and CHANGE commands both seem simple, but there are a few things you have to be careful of.

It's not a good idea to use the CHANGE command followed by * * unless you're absolutely sure of what you're doing. When you put * * after the CHANGE command, the editor takes you at your word that you want it to search every line of the file, from the current line to the last line, to find the letters and make the change.

It's important to include spaces both before and after the words you want to change. This reduces the number of wrong matches that might occur.

Backward Locate: A backward LOCATE works just like the LOCATE command, except the editor searches backward (toward the top of file) rather than forward (toward the end of file). You identify a backward LOCATE by typing a minus sign (-) before the slash (/).

Locating Adjacent Words: The LOCATE command can be used to locate two or more adjacent words in a line. For example, you could enter LOCATE/ word1 word2/.

Repeating a Command - The = (equal) Command

After you have issued a command, you can use the PF9 key, which is set to the = (equal) command, to automatically repeat the last command. It saves you from having to retype it.

Combining Files

Inserting Data from Another File: To insert all or part of one file into another file, you can use the GET command. You could use the GET command to combine chapters in a book. Or if you had three separate files on tax deductions, charitable deductions, and medical deductions, you could use the GET command to make one comprehensive deductions file.

Inserting a Whole File: To combine two files, file1 work and file2 work, you would do this:

1. Enter command "xedit file1 work."

   Note: The GET command inserts another file **after the current line** in the file you are editing. Therefore, you must make current the line after which you want the data inserted. We'll make the last line current by using the BOTTOM command.

2. Enter command "bottom."

3. Enter command "get file2 work."

The message EOF REACHED means that a copy of the entire file has been inserted at the end of "file1 work." The last line inserted is the new current line.

You copy "file2 work" when you GET it. You do not remove it from storage.

Inserting Part of a File: To insert part of a file after the current line, you indicate the **line number** of the first line and the **number of lines** you want to insert at the end of the GET command, like this:

```
GET filename filetype first-line number-of-lines
```

Cancelling Changes to a File

You've made changes to a file, but you decide you don't want to keep the changes. How can you get rid of the changes and keep the file as it was originally? That's easy. **Press PF3, which is set to the QUIT command.** (You can also enter QUIT on the command line, but pressing PF3 is easier.)

The following message is displayed: FILE HAS BEEN CHANGED. USE QQUIT TO QUIT ANYWAY.

If the file has been changed (or is new), the editor gives you that warning message to see if you really want to quit (that is, not put this file on disk).

If you QUIT when you really meant to enter a FILE command, you would then enter FILE. But if you intend to quit, enter QQUIT, as the message says, on the command line. You can enter an abbreviation, QQ.

When you get a copy of a file on your screen just to look at it, not to change it, it's better to QUIT instead of FILE when you're finished. The FILE command tells the editor to rewrite the changed file over the old file on disk. Because you haven't made any changes, you really don't need to have the editor rewrite the file. The QUIT command is faster because the editor doesn't have to rewrite the file.

Numbering Lines of a File

When you used the GET command to insert part of a file, you had to know the starting line number and the number of lines you wanted to insert. You use the SET NUMBER ON command to number the lines of the file. Its format is:

```
SET NUMber ON
```

The line numbers appear in the prefix area, replacing the five equal signs. Lines are numbered from the first line of the file, not from the first line on the screen.

The line numbers aren't placed on disk when you enter FILE or QUIT. If you want line numbers the next time you edit the file, you'll have to enter the SET NUMBER ON command again.

If you no longer need the line numbers, enter:

```
SET NUMber OFF
```

Sorting a File

XEDIT Sort: Starting with the current line, the XEDIT SORT command arranges a specified number of lines according to **sort fields.** If each line of your file started with a name, this would be a field you might choose to sort by. You determine the starting and ending columns of the longest name and use them as a parameter on the sort command.

You enter the SORT command on the command line. Its format is:

```
SORT number-of-lines column1 column2
```

You substitute the number of lines you want sorted for "number-of-lines." You can substitute an asterisk (*) here when you want all lines, from the current line to the last, to be sorted. "column1" is the first column in the sort field. "column2" is the last column.

The SORT command begins with the current line, so you need to make the first line you want sorted the current line.

If you're sorting numbers and want them in descending order, from highest to lowest, type a letter "d" in the SORT command, before the first column number. The "d" stands for "descending order." (If you don't type a "d," the file is sorted in ascending order.)

For example, the command:

```
===> sort * d 1 4
```

would sort the numbers in columns 1 through 4, from the current line to the end of the file, in descending order.

CMS SORT Command: The CMS SORT command allows you to sort information without requiring you to edit the file.

If, for example, you want to sort the information in the file RADIO WAVES A by frequency and store the result in a file called RADIO FREQS A, type the following information on the CMS command-line:

```
SORT RADIO WAVES A RADIO FREQS A
```

The CMS SORT command will ask you to specify which columns bound the fields on which you want the file sorted. You can specify up to 253 pairs of numbers, each pair designating the starting and ending columns of a data field to be used for sorting the file. SORT will use the leftmost pair of numbers as the major sort field. When the SORT is complete, you will have two files on your A-DISK: the original RADIO WAVES and a new file RADIO FREQS,

with the same information as RADIO WAVES sorted in ascending sequence of the fields you specified. Here's the screen before sorting:

```
RADIO    WAVES    A1  F 80  Trunc=80 Size=12 Line=5 Column=1



===== * * * Top of File * * *
===== WEST GERMANY       9765
===== VOICE OF AMERICA   6125
===== AUSTRALIA          6035
===== QUITO, ECUADOR     9560
===== NETHERLANDS        9865
      |...+....1....+....2....+....3....+....4....+....5....+....6....+.
===== GREECE             9615
===== BBC                9750
===== HAVANA             6165
===== CANADA             9655
===== SPAIN              6090
===== PEKING             9780
===== MOSCOW             9450
===== * * * End of File * * *

===> _
                                                    X E D I T   1 file
                                          2 Local 3277 Session
```

```
Ready;
 sort radio waves a radio freqs a
DMSSRT604R Enter sort fields:
 20 23
Ready;
```

Here's the screen after sorting:

```
 RADIO    FREQS    A1  F 80  Trunc=80 Size=12 Line=6 Column=1



===== * * * Top of File * * *
===== AUSTRALIA          6035
===== SPAIN              6090
===== VOICE OF AMERICA   6125
===== HAVANA             6165
===== MOSCOW             9450
===== QUITO, ECUADOR     9560
      |...+....1....+....2....+....3....+....4....+....5....+....6....+.
===== GREECE             9615
===== CANADA             9655
===== BBC                9750
===== WEST GERMANY       9765
===== PEKING             9780
===== NETHERLANDS        9865
===== * * * End of File * * *

===> _
                                                    X E D I T  1 file
                                            2 Local 3277 Session
```

Protecting Files

You can minimize the danger to a new file you're creating or the changes that you're making to an existing file from system or line failure by using either the SAVE command or the SET AUTOSAVE command, or both.

SAVE Command

You can enter a SAVE command at any time while you're creating or revising a file. Its format is:

```
SAVE
```

After you press the ENTER key, the file you're working on is written on disk. However, it will stay on the screen. After you add lines or make more changes, you'll have to enter the SAVE command again to save those new lines or changes.

If you've issued a SAVE command while working on a file, you still have to issue a FILE (or QUIT) command to end the editing session. If you have used a SAVE command while working on a file and then enter QUIT, the changes

you made before you entered SAVE will be written on disk. But the changes you made after you entered SAVE will not be written on disk.

SET AUTOSAVE Command

If you don't want to bother entering SAVE periodically, you can use the SET AUTOSAVE command. You need to enter it only once during an editing session. This command issues a SAVE command automatically each time you've typed in or changed a certain number of lines. You decide what the number should be.

If you want the file saved every time you change ten lines, your command is:

```
===> set autosave 10
```

The SET AUTOSAVE command can be entered at any time during an editing session. However, it's best to enter it at the beginning of an editing session, right after you call out the file.

If you've entered a SET AUTOSAVE command and system operation stops while you're working on a file, your file won't be saved under its own file label. It will have a new file label. Every time the automatic save occurs, you'll see the message AUTOSAVED AS n AUTOSAVE A1. The part of the message that reads "n AUTOSAVE A1" is the new label. The filename will be a number (like 1, 2, or 3) and the filetype will be AUTOSAVE. You would now have two files, one with the original label, and one with the AUTOSAVE label. However, the AUTOSAVE file is the more current one, so you want to replace the old file with it.

Managing Files

You use the editor to make changes to the contents of your files. But sometimes you may want to make other kinds of changes. For example, you may want to rename a file or even discard it. Or what do you do when you forget a file label and you want to edit that file? The commands discussed in this section make it easy for you to manage your files, no matter how many you have.

More information on the commands discussed in this section can be found in the *VM/SP CMS Command and Macro Reference* and the *VM/SP CP Command Reference for General Users.*

Getting a List of Your Files

If you can't remember the exact name of a file you want to work with, you can use the FILELIST or LISTFILE commands. Both these commands display a list of the files you have on your disk. Use the FILELIST command if you want to do something (EDIT, ERASE, RENAME, etc.) to the files that are listed. Use the LISTFILE command if you only want information about the names of your files.

Remember that every file has a three-part label: filename, filetype, and filemode. Generally, the files you create and change have a filemode of A. This means that they are all kept in a collection of files called your **A-disk.** The FILELIST and LISTFILE commands display a list of all the files on your A-disk. (Later you'll see how to display a list of files on other disks, such as B-disk, C-disk, etc.)

Using the FILELIST Command

Here's a sample filelist:

```
 yourid   FILELIST A1  V 106  Trunc=106 Size=6  Line=1 Column=1

Cmd  Filename Filetype Fm Format Lrecl Records Blocks   Date     Time
     FILE1    WORK     A1 F         80       7      1  1/23/84 14:48:20
     FILE2    WORK     A1 F         80       9      1  1/23/84 14:45:02
     FILE3    WORK     A1 F         80      22      2  1/23/84 13:27:28
     FILE4    WORK     A1 F         80       8      1  1/23/84 10:50:59
     FILE5    WORK     A1 F         80       4      1  1/23/84 10:45:19
     FILE6    WORK     A1 F         80      12      1  1/23/84 10:35:00









1= Sort(name) 2= Refresh    3= Quit        7= Backward 8= Forward 9= FL /n
4= Sort(type) 5= Sort(date) 6= Sort(size) 10= Execute 11= XEDIT  12= Cursor
===>
                                                    X E D I T   1 file
                                          2 Local 3277 Session
```

The file list shows a lot of information about the files on the A-disk.

1 The first line has information about the file list itself. Your userid appears instead of "yourid." The SIZE field shows how many files are in this list. Here, SIZE=6 means there are 6 files on this A-disk. The LINE field tells you which file in the list is the first on this screen. Here, LINE=1. As you scroll through the list, the number will change.

2 The next line has headings that show what information is listed about each file:

Cmd
: is a space where you can type certain commands. We'll discuss this later.

Filename
: is the filename of the file.

Filetype
: is the filetype of the file.

Fm
: is the filemode of the file. You can ignore the number that appears after the filemode letter, for example, A1.

Format Lrecl Records Blocks
: are fields that contain information about the file format and are not important to you now.

Date and Time
: is the last time you changed this file. If the file was never changed, this is the date you created it.

3 The PF keys are set to various useful functions, which are shown at the bottom of the screen. They are different from the PF key settings when you are editing a file. For now, note that when you are finished using the list, press PF3 to "quit," or leave, the filelist screen.

Finding Files in the List: If your list takes up only one screen, it's easy to find any file at a glance. But what if you have many files, and the list takes up more than one screen? There are several ways to find files in your list.

Scrolling through the List: If the list takes up more than one screen, you use PF keys to "scroll" through it. PF7 scrolls backward one full screen. PF8 scrolls forward one full screen. By pressing these keys repeatedly, you can scroll through the list for as many screens as you want.

Rearranging the List: The files in the list are arranged by date and time, newest to oldest. The date and time refer to the last time a file was changed (or if it was never changed, when it was created). If you'd like the list arranged in a different way, you can use these PF keys:

PF1
: arranges files alphabetically by filename.

PF4
: arranges files by filetype.

PF5
: arranges by date, newest to oldest. This is how the list is arranged initially. You might use this key if you rearranged the list, and want to put it back to its original order.

PF6
: arranges by size, biggest to smallest.

Using the LOCATE Command: If you know a filename and/or filetype, you can use the XEDIT command LOCATE to find it quickly. This method is faster than scrolling through the screens, looking for a certain file. You enter a LOCATE command on the command line at the bottom of the screen, next to the large arrow. Then press the ENTER key.

For example:

```
===> LOCATE/FILE4 WORK/
```

If FILE4 WORK is located, the line containing it becomes the first line on the screen (and is highlighted). If it isn't found, a message will tell you so.

Variations on the FILELIST Command: You can use the FILELIST command to display a list of **all** the files on your A-disk or only **some** of the files.

You can ask for a list that contains all files that have the same filename, or the same filetype, or all files that begin with a certain letter, etc.

If you don't specify a letter for "filemode," a filemode of A is assumed. If you specified B, all files on the B disk would be listed.

To see a list of all the files on your A-disk with the filetype WORK, you would enter:

```
FILEL * WORK
```

The asterisk means "all." Typing an asterisk where you'd normally type a filename means that you want the list to contain all filenames with the filetype work.

The FILELIST command can also be used to list files having the same filename, but different filetypes. Just use an asterisk where you would normally type the filetype. For example:

```
FILEL FILE1 *
```

The list would contain all files whose filename is FILE1.

If you want to create a file and don't remember if you already have a file with the file label you want to use, you can enter:

```
FILEL filename filetype
```

If the file exists, the file list will show only that file. If not, a message tells you that the file was not found.

Sometimes you may not remember a whole filename or a whole filetype. When you have many files, this can happen. You can give the computer "clues" to display the list of files you want.

For example, suppose you know you have a number of files whose filenames all start with FILE and whose filetype is WORK. You want the list to contain all these files. You would enter:

```
FILEL FILE* WORK
```

The command means, you want the list to contain all files whose filename starts with FILE and whose filetype is WORK. The resulting file list might contain these files:

```
FILE1     WORK  A
FILE2     WORK  A
FILE3     WORK  A
FILE4     WORK  A
FILE5     WORK  A
FILE6     WORK  A
```

You can use an asterisk to represent any letters you can't remember. The asterisk can be used anywhere in the filename or filetype.

Editing a File from the List: You know that to get a copy of a file on your screen, you use an XEDIT command. You can type an XEDIT command on the file list, next to the large arrow at the bottom of the screen. Then press the ENTER key. An easier way is to:

- Move the **cursor** to the line describing the file you want to edit.
- Then **press PF11** (which is set to the XEDIT command).

This does exactly what entering an XEDIT command does. The file appears on the screen, replacing the file list.

When you are finished editing the file and enter FILE or QUIT, the file list returns to the screen. The cursor remains on the line that describes the file you just edited, and an asterisk (*) appears in the "Cmd" space (next to the

filename). This is just to remind you that you did something with this file. It does not prevent you from working with this file again.

Discarding a File: You can use the DISCARD command to get rid of files you no longer want. To discard a file, you

- Type DISCARD in the "Cmd" space next to the file (or files) you no longer want. If the command name is longer than the space provided, just type right over the information already in that line.

- Press PF10 - not the ENTER key - to execute the command.

IMPORTANT NOTE:

DISCARD is one of several commands that you can type in the "Cmd" space next to the filename. When you type a command in the "Cmd" space, you must press PF10 to carry it out, or "execute" it. PRESSING THE ENTER KEY WILL NOT CARRY OUT COMMANDS TYPED IN THE "CMD" SPACE. Only commands typed in the large arrow command line at the bottom of the screen are executed by pressing ENTER.

Once a DISCARD command is executed, the computer doesn't ask if you'd like to think it over. The file is gone. So, if you have any doubts, you're safer if you use a command to rename a file.

Renaming a File: You can change the filename or filetype (but not the filemode) of a file with the RENAME command.

You type the RENAME command in the "Cmd" space next to the file you want to rename. You execute it by pressing PF10. The format of the RENAME command is:

```
RENAME  /  newfilename newfiletype filemode
```

First you type RENAME in the "Cmd" space. You continue typing the rest of the command right over the information that is already on the line. The slash (/) is a kind of

shorthand. It stands for the file label of the file you want to rename. You then type the new file label (filename filetype filemode). Press F10 to carry out the RENAME command.

Copying a File: By using the COPYFILE command, you can make a copy of a file, make changes in it, and keep both the original file and the copy.

You type the COPYFILE command starting in the "Cmd" space next to the file you want to copy. Press PF10 to execute the command. The format of the COPYFILE command is:

```
COPYFILE  /  newfilename newfiletype newfilemode
```

First you type COPYFILE in the "Cmd" space. You continue typing the rest of the command right over the information that is already on the line. The slash (/) stands for the file label of the file in this line. You then type the new file label (filename filetype filemode). Press PF10 to carry out the COPYFILE command.

Comparing Two Files: If you want to see how two similar files are different, you can use the COMPARE command. This command shows you only those lines that are different.

You type the COMPARE command in the "Cmd" space next to the file you want to copy. Press F10 to execute the command. The format of the COMPARE command is:

```
COMPARE  /  filename filetype filemode
```

First you type COMPARE in the "Cmd" space. You continue typing the rest of the command right over the information that is already on the line. The slash (/) stands for the file label of the file in this line. You then type the file label (filename filetype filemode) of the file you want to compare it with. Press PF10 to carry out the COMPARE command.

After you press PF10, you see this message: COMPARING name1 type1 A WITH name2 type2 A.

If the files are identical, nothing else is displayed. The status space (lower right corner) displays a MORE... notice. To get

the file list back on the screen, press the CLEAR key. The file list comes back on the screen with an asterisk in the "Cmd" space.

If the files are different, the lines that don't match are also displayed. The first line is from the first file, and the second line is from the second file. After all lines that are different are displayed, the message FILES DO NOT COMPARE appears. If one file is shorter than the other, you'll see this message: PREMATURE EOF ON filename filetype filemode. The screen will display MORE... in the status area. Press the CLEAR key to return to the file list.

Using the LISTFILE Command

The LISTFILE command displays the names of files. LISTFILE is useful when all you want to do is display file directory information. If you anticipate that you will want to edit, copy, rename or perform other file maintenance functions on the files in the list, you will probably find that FILELIST is more useful.

To display a list of files using LISTFILE, you type

```
LISTFILE (filename) (filetype) (filemode)
```

The same rules apply for using LISTFILE as for FILELIST.

Notice that CMS displays the list one line at a time, rather than in full-screen mode, which makes it impossible to do further operations with the list on the screen. You may find that for some functions, LISTFILE responds more quickly than FILELIST. For example, if you want to see a list of all filenames beginning with "UG" on your A-DISK,

```
listfile ug* * a
```

will display the list on your screen faster than FILELIST. Two more examples of using LISTFILE follow.

If you type the LISTFILE command with the ALL option,

```
listfile ug* (all
```

you will see more information about your files.

```
Filename Filetype Fm Format Lrecl     Recs     Blocks
UGDISK   SCRIPT   A1 V          74      153          9
UGFILES  SCRIPT   A1 V          71       78          4
UGOUTLIN SCRIPT   A1 V          64       10          1
UGPFKEY  SCRIPT   A1 V          72       14          1
UGPROF   SCRIPT   A1 V          70       10          1
UGSORT   SCRIPT   A1 V          74       25          2
UGSYN    SCRIPT   A1 V          74       24          2
UGLIST   SCRIPT   A1 V          74       51          3
```

If you type the LISTFILE command naming a specific disk to be searched, with the DATE option,

```
listfile ug* * b (date
```

you will see still more information about the files on the specific disk you requested.

```
Filename Filetype Fm Format Lrecl  Recs  Blocks    Date     Time
UGDISK   SCRIPT   B1 V          74   153      11  1/25/84  9:30
UGFILES  SCRIPT   B1 V          71    78       5  1/25/84  9:24
UGLIST   SCRIPT   B1 V          74    51       3  1/23/84 10:12
UGOUTLIN SCRIPT   B1 V          64    10       1  1/23/84 14:35
UGPFKEY  SCRIPT   B1 V          72    14       1  1/16/84 11:38
UGPROF   SCRIPT   B1 V          70    10       1  1/25/84 16:14
UGSORT   SCRIPT   B1 V          74    25       2  1/23/84  8:11
UGSYN    SCRIPT   B1 V          74    24       2  1/16/84 12:12
```

There is an additional function in LISTFILE that is not available from FILELIST. LISTFILE has an option called EXEC that will store the list in an EXEC file named CMS, instead of displaying it on the screen. For example, when we enter the LISTFILE command with the EXEC option:

```
LISTFILE UG* * A (EXEC
```

LISTFILE builds this file:

```
&1 &2 UGDISK   SCRIPT   A1
&1 &2 UGFILES  SCRIPT   A1
&1 &2 UGLIST   SCRIPT   A1
&1 &2 UGOUTLIN SCRIPT   A1
&1 &2 UGPFKEY  SCRIPT   A1
&1 &2 UGPROF   SCRIPT   A1
&1 &2 UGSORT   SCRIPT   A1
&1 &2 UGSYN    SCRIPT   A1
```

The "&1 &2" at the beginning of each line have special meaning in an EXEC file. When you learn more about EXEC files, you'll find uses for that information. You

should rename the file if you want to save it for later use. Remember, LISTFILE will replace the CMS EXEC with a new copy every time you specify the EXEC option.

Printing Files

When you want one of your files printed, you can use the PRINT command.

You can print any screen displayed on your terminal by pressing the uppercase key and the RESET (PrtSc) key. The screen will be printed on your Personal Computer printer.

Note: You can stop any printing that's in progress by issuing the command:

```
PURGE PRINT ACTIVE
```

This cancels only the current printing. To cancel all printing you've asked for, issue the command:

```
PURGE PRINT ALL
```

(You can also cancel a specific print by specifying the print number (nnn) instead of ALL.)

PRINT Command

You can type the PRINT command either on the command line or on a FILELIST screen. Its format is:

```
PRint filename filetype filemode
```

You must specify at least the PR of PRINT. You also must specify the filename and filetype. If you don't specify a filemode, VM/PC will assume the filemode is A. See "PRINT" on page 7-104 for further information about options you may specify.

See "An EXEC to Print a File" on page 5-40 for an example of using EXECs to make your printing easier.

Remote Printing

If you are using a Remote Session, you can use any Printing method available to the Remote System. The DCF Program Product, often referred to simply as SCRIPT, is an example of something that might be available on a Remote System.

You can use the printers of your remote VM system by spooling your local printer output to the remote VM system (when the VMPCSERV program is running in the remote VM session). You may also tag the files to send print files to other printers in your VM network.

To send a print file to your remote VM system, you:

1. Start the VMPCSERV program in your Remote 3277 Session.

2. SPOOL your local printer to the correct class.

3. (optionally) Issue the appropriate TAG command.

4. PRINT the file.

Normally, your local printer is spooled to class *. If you change this class, printer files are spooled to your remote VM system, where the VMPCSERV program issues the appropriate printer commands. When you want to use your local printer again, you spool your printer class back to *.

You may also spool your printer to or for another user, in the same way that you might do this on your remote VM system. Chapter 7 contains a full description of the CP SPOOL and TAG commands, which you should consult for further information.

Writing EXECs

What is an EXEC? An EXEC is a file with a filetype of EXEC. It contains a series of commands that are all executed when you enter the filename of the EXEC file.

An EXEC can help you by simplifying routine tasks. How does it do this? You have learned that you sometimes need to type several commands to do a task. If you find yourself often keying in the same sequence of commands during your terminal sessions, you can create an EXEC that contains the commands you would otherwise need to type. All these commands are executed when you enter the filename of the EXEC. An EXEC can also "prompt" (display a message), asking you to enter information that the system needs to complete the task you've given it.

In this section, you'll learn how to create simple EXECs. You'll learn some of the features of the EXEC 2 language, which has special words that are used to tailor your EXECs to meet your needs.

More information on EXECs can be found in the *VM/SP EXEC 2 Reference* and the *VM/SP CMS User's Guide.*

An EXEC to Print a File

You can use the PRINT command to print one copy of a file. But if you wanted five copies, you'd first have to enter a SPOOL PRINT COPY 5 command. Your first EXEC will contain these two commands. An EXEC can have any filename (up to eight letters long) and must have a filetype of EXEC. Let's call this file MYPRINT EXEC A.

Enter:

```
xedit myprint exec
```

Now enter an INPUT command and type these lines. (In the PRINT command, you can use the file label of one of your files instead of the one shown below.)

```
&trace off
cp spool print copy 5
```

```
print work1 file
cp spool print copy 1
&exit
```

Now press the ENTER key twice to return to edit mode. Let's look at the lines in this EXEC.

&TRACE OFF
calls the EXEC 2 processor (a part of the VM/PC system) to handle your EXEC. It also means that you don't want the lines in the EXEC to be displayed as they are executed. If you wanted them displayed, you would use &TRACE ALL.

CP SPOOL PRINT COPY 5
is the command you use to get more than one copy printed. In an EXEC, you must put "CP" in front of this command. It stands for "Control Program," a part of the VM/PC system. You don't need to know which commands are CP commands when you type this command at a terminal. But in an EXEC, you do. It tells the EXEC processor to get ready for a CP command.

PRINT WORK1 FILE
This command has the same format as when it is entered at the terminal. It gives the filename and filetype of the file you want printed.

CP SPOOL PRINT COPY 1
resets the number of copies printed to one.

&EXIT
is the last statement in an EXEC file.

Enter a FILE command. The EXEC is now on your disk. How do you use it? Just type the filename MYPRINT on the command line and press the ENTER key. A message will show that five copies of your file are being printed.

Creating an EXEC is like creating a new command. Each time you want five copies of this file printed, all you need do is enter MYPRINT.

Now, the MYPRINT EXEC is good for printing five copies of the file specified in the PRINT command. But what if you want three copies, not five? And what if you want a different file printed? Do you have to write a different EXEC for every file? No, you can make a few changes in the MYPRINT EXEC so that you can use it to print **any number** of copies of **any file.**

Enter XEDIT MYPRINT EXEC and make the following changes:

```
&TRACE OFF
CP SPOOL PRINT COPY &1
PRINT &2 &3
```

You changed the number of copies to &1, the filename to &2, and the filetype to &3.

These numbers preceded by an ampersand (&) are a signal to the EXEC processor that they will be replaced with values that you type in. Now when you enter MYPRINT, you will also type real values that will replace the &1, &2, and &3. The first value you supply must be number of copies you want printed, the second value must be the filename, and the third value must be the filetype.

Let's see how this works. Enter a FILE command. Then enter a command like this, using the filename and filetype of one of your files:

```
myprint 3 work1 file
```

The EXEC processor takes the values you type and assigns them to &1, &2, and &3, in that order.

This command	**Becomes this command for this execution:**
CP SPOOL PRINT COPY &1 PRINT &2 &3	CP SPOOL PRINT COPY 3 PRINT WORK1 FILE

You can use a different number and a different filename and filetype each time you enter MYPRINT. You must specify these values in the order shown, because the first value you

type after MYPRINT replaces &1 in the EXEC, the second replaces &2, etc. If you mixed them up, like typing the filename first, your EXEC wouldn't work.

Setting Up Your System to Meet Your Needs

Finding Out Basic Information

To find out your userid, and the date, time, and day of the week, use this command:

```
===>IDentify
```

You'll get a message like this one:

```
USERID AT *   VIA *   01/23/84  12:01:00 LOCAL  MONDAY
```

How Much Disk Space Have I Used?

When you enter a FILE command, a file is stored on a disk. The disks are identified by letters, known as filemodes. The filemode that you usually use when you create a file is "A." That means that your file is put on your A-disk. You can find out how much disk space is left with this command:

```
===> Query DISK A
```

If you do not specify the disk letter, VM/PC displays the disk information for all currently-accessed disks.

The response is a message similar to this:

```
LABEL  CUU M STAT CYL TYPE BLKSIZE FILES BLKS USED-(%) BLKS LEFT BLK TOTAL
mydisk 101 A  R/W  FB C:PC  1024       8  206-20          818      1024
```

The disk label is mydisk. The "address" of the disk (where the disk is) is 101. The filemode letter is A. R/W means that you can read and write on this disk. Some disks are R/O, which means "read only." That is, you can't write a file on them. The fields labeled CYL, TYPE, and BLKSIZE give information about the amount of space on this particular disk device and how that space is organized. FILES tells you how many files you have on the A-disk. BLKS USED shows the percentage of the space you have used. Here, you see that 20% of this disk is used. Therefore, 80% of this A-disk is still available.

You can get this information for any minidisks that you have ACCESSed by entering the appropriate filemode letter in place of A.

How Does CMS Choose Programs and Files?

When you enter a command, or specify a file name to XEDIT, CMS looks for a matching command name (EXEC or MODULE) or filename among those on the minidisks you have currently ACCESSed. If you have ACCESSed more than one minidisk, CMS begins its search with the letter closest to the beginning of the alphabet, and progresses through the hierarchy until it either finds the command or file, or it has searched all ACCESSed minidisks.

If you notice that commands that have worked in the past and files that used to be accessible have suddenly disappeared, check CMS's search order to be sure that you have access to the minidisks on which the data resides. You can use the QUERY SEARCH command to determine which minidisks CMS will search for programs and data, as well as the order in which the search will be done. In addition, CMS will tell you the ACCESS mode and the read/write status of each disk in the search hierarchy. R/O means "Read/Only" and R/W means "Read/Write."

A remote minidisk must appear as R/W, if you want to "Upload" files from VM/PC to your remote system. You may "download" files from a remote system with either Read or Write access to a remote minidisk.

Modifying the ACCESS List

You can add disks or change the search order of the disks in the list by using the ACCESS command. You can drop a disk from the list by using the RELEASE command.

Before you can ACCESS a disk, however, you need to make it available to your virtual machine by using the CP LINK command. CP LINK is also the means by which you can alter the read/write status of a minidisk. To find out which minidisks are LINKed to your virtual machine, use the QUERY DASD command.

The DETACH command is useful for eliminating LINKs to minidisks that you no longer want to ACCESS. You can drop a minidisk from the CMS search hierarchy and simultaneously destroy the LINK with the following command: RELEASE nnn (DET .

The example below shows you how to use these commands and data to set up a local VM/PC session with access to a set of minidisks owned by your remote virtual machine.

```
 query search
Mydisk  101   A    R/W
System  100   S    R/O
Ready;
 query dasd
DASD 100 FB-512 C R 2048 BLK CMS         100
DASD 101 FB-512 C W 2048 BLK
DASD 10E FB-512 C R 2048 BLK CMS         10E
Ready;
 cp link * 191 271 mw
DASD 271 LINKED R/W; R/W BY YOURID
Ready;
 access 271 r
Ready;
 cp link systems 500 500 rr tools
Ready;
 access 500 t
Ready;
```

In this example, you can see that, initially, we have ACCESS to two of the three minidisks that are LINKed to our virtual machine. QUERY SEARCH gives us information about the ACCESSed minidisks, and QUERY DASD tells us about the LINKs.

The two pairs of LINK and ACCESS commands that follow the QUERY DASD provide us with access to information and programs residing on the 191 (A-DISK) and 500 disks on our remote system. (We had already started the Remote Server program, VMPCSERV, on the remote system.) The following QUERY SEARCH and QUERY DASD commands show that the LINKs and ACCESSes have, indeed, been successful.

```
Ready;
 query search
Mydisk  101   A    R/W
A-DISK  271   R    R/W
```

```
System  100   S    R/O
TOOLS   500   T    R/O
Ready;
 query dasd
DASD 100 FB-512 C R 2048 BLK CMS         100
DASD 101 FB-512 C W 2048 BLK
DASD 10E FB-512 C R 2048 BLK CMS         10E
DASD 271 REMOTE   W          YOURID      191
DASD 500 REMOTE   R          SYSTEMS     500
Ready;
```

Next, suppose that we no longer want WRITE ACCESS to the 191-DISK on the remote system, but that we want to continue to refer to data and programs in READ/ONLY mode. To make that change, we must DETACH the R/W LINK to 191, and reestablish the LINK as R/O. The following example shows you how to change a minidisk from R/W to R/O.

```
 release r (det
DEV 271 DETACHED
Ready;
 cp link * 191 271 rr
DASD 271 LINKED R/O; R/W BY YOURID
Ready;
 access 271 r
Ready;
```

An Easy Way to Set Up a Concurrent Session

After you have set up a concurrent session linking your local VM/PC session and your Remote 3277 session, you can transfer files from one system to the other. This transfer is referred to as upload and download.

You can simplify the setup process for upload and download by saving the appropriate commands in an EXEC file. If you name each EXEC file with a name that you can easily associate with the process you want to complete, setup becomes a matter of typing a single command (the EXEC filename). You can see examples of two very simple EXECs below that perform the LINK and ACCESS needed to allow you to share data and programs between remote and local sessions.

The first example is UPLOAD EXEC and the second is DOWNLOAD EXEC.

```
&TRACE OFF
 RELEASE 271 (DET
 CP LINK * 191 271 MW
 ACCESS 271 R
&EXIT

&TRACE OFF
 RELEASE 271 (DET
 CP LINK * 191 271 RR
 ACCESS 271 R
&EXIT
```

Once you have filed away UPLOAD EXEC and DOWNLOAD EXEC, UPLOAD and DOWNLOAD become part of CMS's command vocabulary. When you type either word, CMS will carry out the commands contained in the designated EXEC file just as though you had typed them on your terminal.

Establishing Automatic LINKs and ACCESSes.

While we're on the subject of EXECs, CMS uses a special EXEC called PROFILE that it runs automatically every time you IPL CMS. One way you can use this feature is to store the LINK and ACCESS commands that establish the data sharing relationships you most often use in the PROFILE EXEC. You can also store other favorite commands in your PROFILE, so that you can be sure that they will be executed each time you begin a CMS session. And if, during a session, you discover that you have somehow lost a LINK or an ACCESS, typing "PROFILE" may add back the "lost" minidisk.

Note: Remember that if you have LINK and ACCESS commands for a remote system in a PROFILE EXEC, you must ensure that VMPCSERV has been started in the remote system before the PROFILE EXEC is executed.

Sharing Data and Programs with Other Users

Before long, you may find it useful to be able to share data and programs with other users. As you saw earlier in the section on establishing a concurrent session with your remote virtual machine, VM/PC allows you to LINK and ACCESS data and programs associated with other remote virtual

machines (users). You may also LINK and ACCESS data and programs with other virtual machines that have been defined in your VM/PC system.

Note: Before you can link to a virtual machine, you must add information about the virtual machine and its associated minidisks to your Configuration File. See Chapter 3, "Configuring Your VM/PC System" on page 3-1, for information on updating your Configuration File.

VM/PC is shipped with a virtual machine named SAMPLE defined in the Configuration File. The following example shows how to LINK to and ACCESS that virtual machine's A disk (101).

```
cp link sample 101 201 rr abc123

access 201 b
```

You have linked to disk 101 of the virtual machine whose userid is SAMPLE. You have defined that 101 disk as 201 to your system. You asked only for read access to the disk (rr) and you specified the read-share password for the 101 disk (abc123).

After you linked to the 101 disk as 201, you ACCESSed the 201 disk as disk B on your system.

You can issue the FILELIST command for the B disk to ensure that you have access to the files on the SAMPLE virtual machine's 101 disk.

```
filelist * * b
```

This will list all the files on SAMPLE's 101 A disk, which is the 201 B disk on your system.

Since you LINKed to the disk in read-only mode (rr), you can copy, examine, sort, and compare data in its files, but you cannot alter their contents. This is a very good way to allow other users to access data but prevent them from changing it.

If you want to change data on a disk, you must have write access to the disk. To get write access to SAMPLE's A-disk, issue the LINK and ACCESS commands like this:

```
cp link sample 101 201 w x0y0z

access 201 b
```

The commands are the same as in the example above, except for "w x0y0z" at the end of the LINK command. "w" indicates that you want read-write access to the disk, and "x0y0z" is the write-share password. (Remember that the password you gave earlier, "abc123," was called the read-share password.)

When you no longer need access to a disk, you end your link to the disk with this command:

```
rel 201 (det
```

You can keep often-used LINKs and ACCESSes handy in EXEC files that you can call when you want to access data on another disk. You might give the EXECs appropriate names, like GETSAM EXEC to provide access to the SAMPLE virtual machine.

A Sample PROFILE EXEC

We have included some useful commands to establish a joint session in the sample PROFILE EXEC below. In this example, we have already begun Local and Remote sessions and started the VMPCSERV program in our Remote 3277 Session virtual machine. If we do not start VMPCSERV before CMS runs this PROFILE, the LINK to the remote disk (in DOWNLOAD EXEC) will fail.

```
&TRACE OFF
 EXEC SETPF
 SYNONYM EASY2USE
 CP TERM HILIGHT ON
 CP TERM PA2 HALF
 CP SPOOL PRINT CHAR 80
 EXEC DOWNLOAD
&EXIT
```

PF Keys Make Work Simpler

You may have noticed that the sample PROFILE EXEC contains another EXEC command, called SETPF. We have

listed the contents of SETPF EXEC below so that you can see how commands can be assigned to the Program Function (PF) keys (labeled F1 through F10 on the left side of your keyboard). (You could also set a value for PF11 and PF12.)

```
&TRACE OFF
 CP SET PF1 IMMED FILELIST
 CP SET PF2 IMMED QUERY DISK
 CP SET PF3 DELAY LOG
 CP SET PF4 DELAYED IC_ RELEASE _NNN (DET
 CP SET PF5 IMMED UPLOAD
 CP SET PF6 IMMED DOWNLOAD
 CP SET PF7 IMMED QUERY DASD
 CP SET PF8 IMMED QUERY PRINT ALL
 CP SET PF9 IMMED QUERY SEARCH
 CP SET PF10 RETRIEVE
&EXIT
```

Of course, you can reassign the keys at any time with a CP SET PF(number) command. However, placing the assignments in an EXEC, and calling it from the PROFILE EXEC is a convenient way for you to establish a standard set of Function key assignments for every CMS session.

Note: The example above uses a SET command feature new with VM/PC in the PF4 setting. After this setting, when you press PF4, the command RELEASE nnn (DET will be displayed on the command line. The new feature is that the cursor will be positioned right under the nnn. This makes it easier for you to type in the replaceable part of the command. See "SET" on page 7-38 for further details on this feature.

Reviewing What You Have Done

There are two features you can use to help you review what you have done and do it again.

Scrolling Screen Contents

The first of these is called scrolling. You use commands to redisplay information that has previously been displayed on your screen. These commands are typed on the lower right of your screen, on the next-to-last line, following the word "Scroll:." This feature is only available in a Local 3277 Session. The commands you can use, and their meaning, are:

- F[nn] - Scrolls forward (bottom of screen moves to top) one screen. If you specify a number for nn, it scrolls forward that number of lines.
- B[nn] - Scrolls backward (top of screen moves to bottom) one screen. If you specify a number for nn, it scrolls backward that number of lines.
- TOP - Scrolls so oldest entry is at top of screen.
- BOT - Scrolls so latest entry is at bottom of screen.
- R[nn] - Scrolls screen to the right 80 characters or the number you specified as nn.
- L[nn] - Scrolls screen to the left 80 characters or the number you specified as nn.

  Note: Both the R and L scroll commands work only if you have issued the command TERMINAL WRAP OFF. See "TERMINAL" on page 7-48 for more information.

Note: Both the scrolling commands and the RETRIEVE command (described below) use an area called the history buffer. VM/PC also uses this buffer when displaying on the screen, such as when you TYPE a file. Be aware that the contents of the history buffer may be covered (and lost) by something displayed on the screen.

Retrieving Commands

You can set a PF key to RETRIEVE (as we did in the example in "PF Keys Make Work Simpler" on page 5-50). If you issue the command

```
set pf10 retrieve
```

each time you press the PF10 key VM/PC will redisplay the last command you issued. Press PF10 again and VM/PC will redisplay the next-to-last, and so on. The commands are redisplayed on the command line and you can reissue them by simply pressing the ENTER key.

You can also change the command before you reissue it. This can save you a lot of typing and time. Try it. Set a PF key to retrieve as shown above, and then press it and watch what happens.

When you have set a PF key to retrieve, if you place the cursor on a line you previously entered and press the PF key, the line will be written in the input area. You can then enter the input again, or alter it before entering it. You can scroll the display back to reach input not currently displayed.

When you have selected an input line in this way, successive pressings of the PF key will retrieve input lines previous to the one selected.

Using Synonyms in Your Work

You can give your VM/PC a character all its own by giving alternative names to the commands. Once again, look at the sample PROFILE EXEC. The second command in the file, SYNONYM, will inform CMS that you want certain commands to be recognized by names that you have specified in the synonym file. You can name the synonym file anything you like, provided its filetype is SYNONYM. You can also have several synonym files stored on your minidisks. At any given time, CMS will recognize only the synonyms stored in the file you named in the last SYNONYM command. You can see the contents of the EASY2USE SYNONYM file used in the sample PROFILE EXEC in this example:

```
XEDIT       CREATE    3
GETHOST     DOWNLOAD  4
ERASE       DELETE    3
QUERY       HOWMUCH   4
QUERY       WHAT      4
```

There are just a few more things you should know about synonym files: The first field specifies a CMS command, the second specifies a synonym, the third specifies the minimum number of characters required to recognize the command, and synonyms can be used only for CMS commands. CP will not recognize synonyms for its commands.

You can use the SYNONYM command at any time, to swap synonym files, or to reestablish the original synonyms. If you issue the SYNONYM command without specifying a filename, CMS will list the currently active synonyms, both yours (if any), and those supplied with the system.

Controlling Your Display

VM/PC has several options on the CP TERMINAL command you can use to control what is seen on your display. These options are HILIGHT, PA2, and WRAP. "TERMINAL" on page 7-48 contains a description of the TERMINAL command and all its options.

Highlighting Your Input

Specify TERMINAL HILIGHT ON if you want all input from the keyboard to be highlighted on the display. You can do this to help you identify the commands you typed among all that is displayed.

Scrolling a Half Screen

Specify TERMINAL PA2 HALF if you want VM/PC to scroll the display forward one-half screen when you press PA2 with the cursor in the input area. If you do this without specifying this command, VM/PC scrolls the screen forward one full screen.

If you press PA2 while the cursor is on a displayed line, VM/PC will scroll that line to the top of the display screen. Remember that for VM/PC the PA2 key is emulated by pressing the Alt and F2 keys.

Displaying Long Lines

Specify TERMINAL WRAP OFF if you want VM/PC to display each input line on a display screen line. With this option, lines that are longer than a screen line will be cut off at the right edge of the screen. VM/PC will put an ellipsis (...) on the end of the displayed portion of such lines to indicate that there is information not displayed. You can use the L and R commands in the scrolling area to see the rest of these lines.

If you do not specify TERMINAL WRAP OFF, any part of a input line longer than a screen line will be "wrapped around" and displayed on the following line.

Transferring Personal Computer DOS and VM/PC Files

Files created on the Personal Computer and those created on a VM/PC system are in different formats. VM/PC provides commands to convert these files from one format to the other so they can be used on either system. These commands are IMPORT and EXPORT.

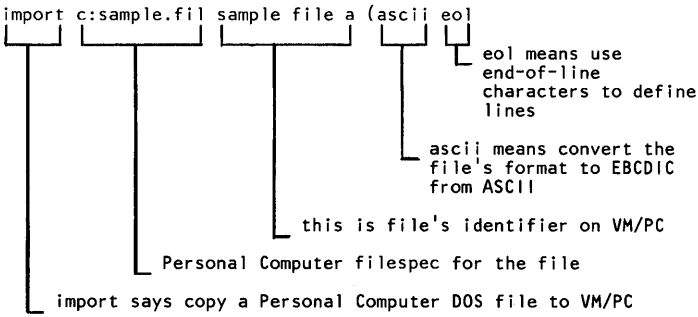
You can transfer data and programs between the IBM Personal Computer environment and the VM/PC environment with the IMPORT and EXPORT commands. IMPORT transfers from Personal Computer DOS to VM/PC, and EXPORT transfers from VM/PC to Personal Computer DOS.

As with other VM/PC commands, IMPORT and EXPORT have many options that offer you many ways to use the system. The following sections show you how to use these commands and options to move a typical text file and a program from one environment to the other. (For our purposes, a text file is one that contains alphanumeric data.)

For more information see "IMPORT" on page 7-94 and "EXPORT" on page 7-76.

Copying a Text File from Personal Computer DOS to VM/PC (IMPORT)

To transfer a Personal Computer text file (named sample.fil on your C disk) to your VM/PC (as sample file on your A disk), you could use the following options on the IMPORT command:



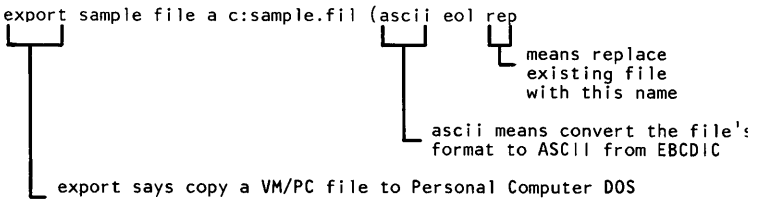
If the records in the file are longer than 80 bytes, you must specify the record length with the LRECL option (for example, lrecl 132). If you don't know the record length, you can use the Personal Computer command 'dir' to find it. (Note that you will have to leave VM/PC to issue this command, then restart VM/PC.)

If you used this IMPORT command, the Personal Computer DOS file named sample.fil on your C disk would be copied onto your VM/PC A disk. The file would be converted from ASCII to EBCDIC and any end-of-line characters would be used to indicate line ends in the VM/PC file.

You do not have to IMPORT files to your A disk. Simply change the file mode letter on the IMPORT command and you can copy the file onto any disk you have accessed in write mode. Similarly, you can EXPORT files from any accessed disk.

Copying a Text File from VM/PC to Personal Computer DOS (EXPORT)

Say that you have done the IMPORT shown above, then made some changes to the SAMPLE FILE A file, and now want to copy the changed file back to your Personal Computer C disk. You could use the EXPORT command with the following options to do that:



The other operands and options are as explained for IMPORT above. Because you specified the option 'rep', the old C disk file named sample.fil will be replaced with the one copied from VM/PC.

Note: You must specify the complete Personal Computer DOS filespec and the VM/PC filename, filetype, and filemode each time you issue the IMPORT or EXPORT commands.

Copying a Program from Personal Computer DOS to VM/PC (IMPORT)

If you have a file whose record length is 1200, you can use the IMPORT command with these options to copy a program (or binary file) from the Personal Computer to VM/PC environment:

```
import c:myprog.com myprog command a (noascii noeol lrecl 1200
```

As with the text file IMPORT shown above, the Personal Computer file on the C disk (named myprog.com) is copied onto your VM/PC A disk (named myprog command). However, different options are given here. NOASCII means don't convert the file from ASCII to EBCDIC when it is copied. NOEOL means don't use any end-of-line characters in the file to define line ends in the new file.

Note that in general, you will use the "ascii" option with the "eol" option if you are copying text files, and the "noascii" option with the "noeol" option if you are copying programs.

Copying a Program from VM/PC to Personal Computer DOS (EXPORT)

If you want to copy the program back onto your Personal Computer C disk, you could use this EXPORT command and options:

```
export myprog command a C:myprog.com (noascii noeol rep
```

The only difference here from the IMPORT command used above is the option 'rep'. That means replace the file named myprog.com on the C disk with the file being copied.

Chapter 6. Remote Services

Remote Upload/Download

This chapter tells you how to use the VM/PC Remote Services to access a remote VM system. VM/PC provides Remote Services that enable you to use the resources of a remote VM system in a number of ways. With the 3277-2 Coaxial Connection, your VM/PC will appear as a 3277 terminal to the remote VM system. This enables you to use the VM/PC as a 3277-like terminal and, in conjunction with the VM/PC Remote Services Program, provides you with the functions described below.

With the 3277-2 Coaxial Connection, VM/PC provides the following services:

1. **Remote 3277 Session** - This is the ability to **LOGON** to a remote VM system. You can use a Remote 3277 Session concurrently with a Local 3277 Session (for example, while you are editing a local file in the Local Session you can be compiling a program in parallel on the Remote Session). We describe the Remote 3277 Session in greater detail in the following section.

2. **Remote Minidisks** - When you are in a Local Session, you can **LINK** and **ACCESS** any CMS format minidisk that exists on the remote VM system that you are connected to (assuming you know the appropriate minidisk addresses and passwords). Even though these disks are not local to you, both you and your applications can utilize these disks as if they were in fact local.

3. **Remote Files** - When you have linked to, and accessed, remote minidisks, you can access files on those disks as if the files were indeed **local** to your VM/PC.

4. **Remote Printers** - The Remote Services program allows you to use your remote VM system printers as VM/PC printers. For example, if your installation has defined **Class K** as the print class for an IBM 3800 printer, then

you need only **SPool** your VM/PC printer to CLASS K. When you use the CMS command **PRINT** to print a file, that file will be printed on the IBM 3800 from the remote VM system.

VM/PC Remote Services Program

The Remote Services program allows you to:

1. LINK and ACCESS any CMS format minidisks on the remote VM system **as if those disks were local**; Such minidisks are called **remote** minidisks. You can access and use the minidisks as if they were located on your local VM/PC system, but in fact they are really minidisks on your remote VM system. When you read or write programs and data from or to these minidisks, the read or write requests are executed by the Remote Services program. The local programs function as if the read or write were being done locally, when it is really being done by the remote system, as a result of the local system requesting the remote services program to do the work for it.

2. Access any file on these disks in the normal VM fashion for local file access.

3. Spool printer output to any of the printers on the remote VM system as if those printers were local.

Now we'll describe each of these parts of the VM/PC remote services program.

VMPCSERV Module

To use the Remote Services described above, you must have access to the VMPCSERV module on the remote VM system. If you do not have access to a copy of the VMPCSERV program, you must install it. The following section describes how to install VMPCSERV (also referred to as the Remote Server program).

Installing VMPCSERV

If you wish to use the remote services but do not have access to VMPCSERV on your remote VM system, you must install it there. Note that the Remote Services are only available over the 3277-2 Coaxial Connection between VM/PC and your remote VM system.

You install the remote server in three parts:

- You load the HEXIN program in source code form using the EDIT command on your remote VM system, assemble it, and the HEXIN module is then created.

- You load the TEXT files needed to generate the VMPCSERV module to your remote system using the HEXIN module.

- You use these TEXT files to create the VMPCSERV module, and Upload the VMPCSERV exec using the EDIT command on your remote VM system.

You will need approximately 200 blocks of space on your Remote System's A-disk to Upload the Remote Server files (if your A-disk is formatted in 1024-byte blocks).

The remote server program is contained on distribution diskette 3. Follow these steps to install it:

- Step 1 - Bring up VM/PC (it is assumed that you have already installed VM/PC; if not, refer to Chapter 2 for that procedure).

- Step 2 - From the Session Selection Menu, select the Remote 3277 Session.

- Step 3 - Logon to your remote VM system, IPL CMS, and enter the following commands (they may be executed from an EXEC if you prefer):

  GLOBAL MACLIB DMSSP CMSLIB (or the maclib names on your remote system which contain the CMS macros)

  CP SET MSG OFF

  CP SET WNG OFF

  CP SET ACNT OFF

  CP TERMINAL ESCAPE OFF LINEND OFF LINEDEL OFF LINESIZE 80

CP TERMINAL HILIGHT OFF (if your system supports the HILIGHT option of the TERMINAL command)

SET AUTOREAD ON

SET BLIP OFF

SET IMPEX OFF (if you have an exec called EDIT on your system which would invoke an editor other than the CMS EDIT editor.)

Note: You will probably want to restore the SET and TERMINAL characteristics following this installation procedure.

- Step 4 - Now return to the Session Selection Menu (by pressing the Select key) and select the Local 3277 Session.

- Step 5 - Insert installation diskette 3 in drive A, and logon to your local session using a userid of CMS, a password of CMS (if the default password has not been changed), and an Environment of CMS. Diskette 3 contains the CMS 101 minidisk where the VMPCSERV installation is found. You will receive a message when the Remote Services program minidisk is automatically accessed, asking if you wish to continue with the installation. If you choose to continue, the files needed to create a Remote Server program (the exec and texts) are uploaded to your remote VM system.

  The installation is performed in three parts. You will be prompted on the local session before the start of each part, where you may decide to perform the part, skip the part, or stop the installation of the remote server. You should only skip the part if you are re-starting the installation, and are certain that this part has previously been completed successfully.

  While each part of the remote server installation is being done, you should select the remote session as your active session, and observe the installation procedure for correct functioning. If the message "NOT ACCEPTED"

appears on your remote session screen in the CP status area, you should consult the section "What To Do if Something Goes Wrong," which follows, for the procedure to follow to correct the error. In addition, the installation procedure may detect other errors, which will result in error messages being produced on the remote session screen. You can switch between your local and your remote sessions as you wish, but **do not** enter any commands or press any keys on your remote session until the installation is complete. When you receive the messages that the VMPCSERV module has been created and the VMPCSERV exec has been copied to your remote system, you can logoff from the CMS userid on the local session and logon your own userid.

- Step 6 - Return to the Remote Session and check that no errors have been encountered (such as being out of minidisk space on your remote minidisk, remote failure in the middle of processing, etc). If the Upload procedure completed normally, the VMPCSERV MODULE and EXEC should have been created on your remote VM system CMS A-disk. Erase all other files which were uploaded as they are no longer needed for the Remote Server program processing.

Note: The Remote Server source and the VMPCSERV maclib are contained on the CMS 101 minidisk on installation diskette 1. These files are in a packed format produced by the CMS Copyfile command.

What To Do if Something Goes Wrong

The Remote Server installation depends on your remote VM system being ready to read lines of data which are sent to it from your local VM session. The local session tries to pause between each line sent to your remote system, to allow the remote system to finish processing previous lines sent. Occasionally, however, the installation will not be accomplished because your remote system is not ready to receive lines from the local session. In this case, your remote system will display "NOT ACCEPTED" in the CP status area. If this happens consistently during the install process, you should stop the installation and restart it on your local session with the command "INSTALL delay-value." If you followed the above steps during the installation, the

INSTALL exec was invoked by the CMS user's profile with a delay value of 100. When you restart the installation, you should invoke the INSTALL exec with a larger delay value (150 or 200). If the installation continues to fail, you should continue to increase the delay value. Prior to re-starting the installation, you should erase any files created by the install process on your remote system A-disk.

VMPCSERV installation places 15 files on your remote VM system. If you have files there with the same names as those VMPCSERV is trying to install, the installation will fail. Before installing VMPCSERV you must rename or erase any files on your remote VM system that have these filenames and filetypes:

- DISPAX TEXT
- DISPIO TEXT
- GENPCSER EXEC
- HEXIN ASSEMBLE
- HSTCMS TEXT
- HSTCOM TEXT
- HSTFSR TEXT
- HSTGMEAC TEXT
- HSTGMELN TEXT
- HSTHYPE TEXT
- HSTLOGO TEXT
- HSTSPL TEXT
- HSTVSI TEXT
- VMPCSERV EXEC
- VMPCSERV MODULE

If the uploading of one or more text files in part 2 of the install fails, but the HEXIN module has been created successfully on your remote system, you can restart the install process by issuing the command "PROFILE" on the local session, and skip part 1 of the install. You may also load the text files in part 2 of the install selectively if only 1 or 2 of the files was not loaded successfully. Before restarting the installation, you should erase any of the text files uploaded to your remote system which you will be trying to upload again.

VMPCSERV Program

VMPCSERV is a **remote VM** application program that runs in your Remote 3277 CMS Session under your userid. VMPCSERV simply passes on requests from the local VM/PC session to the remote VM system for processing.

Clearly, you could explicitly logon to the remote VM system (in a remote session) and perform explicit LINKs to minidisks, run programs, etc. If you do this, though, these resources are not available to applications running in the local session environment. What VMPCSERV does is to make these resources available to the VM/PC local environment as perceived local resources.

To use VMPCSERV, you must logon to the remote VM system (by selecting a Remote 3277 Session and performing the logon).

Since VMPCSERV runs as an application under your userid in a Remote 3277 Session, you **cannot** run another CMS application on the Remote 3277 Session unless you terminate the VMPCSERV application.

To start the VMPCSERV program, just enter VMPCSERV (when in a Remote 3277 Session). VMPCSERV displays the following screen:

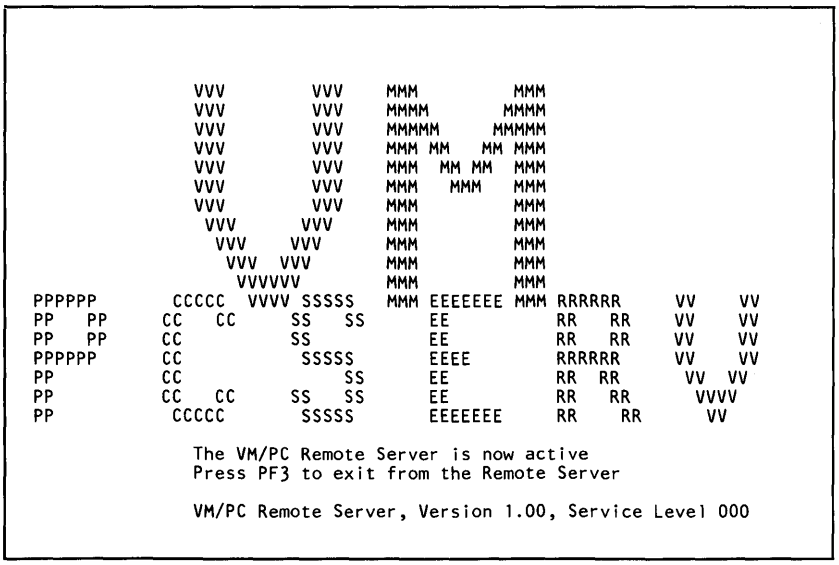


Figure 6-1. VMPCSERV Logo

To stop the VMPCSERV program, press PF3. You will then return to a normal Remote 3277 Session and can explicitly perform any remote service, but your local session will no longer be able to access any of these remote services or files.

After starting VMPCSERV in your Remote 3277 session, use the Select key to return to the Session Selection Menu. There, select a Local 3277 Session and, if you haven't done so already, logon and IPL VM/PC CMS. You now can use these Remote Services:

1. Remote minidisks
2. Remote files
3. Remote printers

Remote Minidisks

You can access remote CMS format minidisks when you are in the VM/PC CMS local environment. You can access programs and utilities which are on your remote VM system without having to copy them to a VM/PC disk. Such

programs are available for your use in the Local 3277 Session as if they were on one of the local minidisks on your VM/PC system.

To use these **remote minidisks** issue the CP command **LINK**.

- VM/PC CP searches to see if the minidisk that you are trying to link to is **local**. If it is local, the disk is linked according to the rules governing the CP LINK command.

- If VM/PC CP determines that the minidisk does not exist locally, rather than failing to perform the link, VM/PC CP sends a request to the VMPCSERV program in your Remote 3277 Session to issue your LINK request. Assuming that the minidisk exists there (and that you have provided the appropriate password) the link is performed and you are then told that you can proceed as normal.

You can always determine when a minidisk is truly local or remote by issuing a **CP QUERY DASD** command. If a minidisk is local, the device type displayed will be **FB-512**; if a minidisk is remote, the device type displayed will be **REMOTE**.

When reviewing the LINK command, in Chapter 7 of this manual, you will notice an optional keyword of REMOTE/LOCAL. (To review, the LINK command in VM/PC CP is of the form:)

```
LINK [TO] userid vaddr [AS] vaddr2 [mode]
     [[PASS=] password] [REMOTE|LOCAL]
```

You may specify REMOTE or LOCAL, but not both, on a LINK command. These keywords are provided specifically for those users who have the same userids and minidisk addresses on both their remote VM system and on the VM/PC. As the normal search order for acquiring a minidisk is to search to see if the minidisk exists locally before determining if it exists remotely, there may be times when you want to explicitly LINK to a remote minidisk and bypass your local one. For example, you may have a userid, say CASE, that has a 290 minidisk both locally and

remotely. Explicitly stating REMOTE on the the LINK command - LINK CASE 290 222 RR mypass REMOTE - will cause VM/PC to **only** perform the LINK on the remote VM system. (Obviously, if the minidisk does not exist remotely then the LINK request will fail!).

If you specify **LOCAL** on the LINK command, only a local VM/PC minidisk search will be performed. If the minidisk does not exist locally, the LINK command will fail.

If neither **LOCAL** nor **REMOTE** is specified, the minidisk request will first be performed locally, and, if not found, the request will be performed remotely.

Password specification rules are defined in the description of the LINK command in Chapter 7. The essential difference between the VM/PC minidisk password specification and that provided by your remote VM system is that you will not be prompted for a password if the LINK is performed by the Remote Services program. You must specify the password on the LINK command in this case.

Once you have performed a LINK to a minidisk, you must issue a CMS ACCESS command in order to identify the minidisk to CMS. ACCESS establishes the CMS file search sequence order by letting you assign a **file mode** to the minidisk. This command is independent of whether or not the minidisk is local or remote. Both ACCESS and RELEASE (when you want to remove a minidisk from CMS) are described in Chapter 7 of this manual. The counterpart to LINK, the DETACH command (when you no longer need the minidisk) is also described in Chapter 7.

Once the LINK and ACCESS have been performed, you are ready to access files, in VM/PC CMS, from either local or remote minidisks.

Note: Temporary disks you have accessed in your remote system will not be available to your local system.

Remote Files

Whether or not a minidisk is really local or remote, once you have performed LINK and ACCESS, you can access files on that disk using CMS and user-written programs. That is, you have the full range of CMS file commands at your disposal.

At a high level, you can invoke an editor, to edit and modify any file; you can TYPE out a given file; you can COPYFILE any file; or you can compile a program. These commands are available to you even if the files exist on a remote minidisk (in such cases we call the files **remote**).

At a low level, you have available to you all of the CMS file system macros, such as:

1. FSCB
2. FSOPEN
3. FSSTATE(W)
4. FSREAD
5. FSWRITE
6. FSPOINT
7. FSCLOSE

You can use these macros in your programs to read and write data in both remote and local files, without the application program having to be aware of where the file is actually located. Some applications which are dependent on certain CMS control blocks and internal pointers may not work. For example, applications which search internal CMS control blocks for a file's FST may not work correctly.

Also, any single read or write of data that your applications perform must be less than or equal to 65535 bytes. Therefore, you should not read or write remote files with a record length greater than 65535, or try to read or write more than 65535 bytes of a remote file with a single file system operation.

File Upload/Download

Uploading refers to the sending of programs and data from your local VM/PC session to your remote VM system. Downloading is the reverse; sending programs and data from your remote system to your local VM/PC session.

In VM/PC, there are no explicit file upload/download commands. Such data sending results from the CMS commands you issue in your local VM/PC session when the VMPCSERV program is active. To upload a file to a remote minidisk, do the following:

1. LINK to the remote minidisk in write mode

2. ACCESS the minidisk (for example as file mode R)

3. Use the CMS COPYFILE command (for example: to Upload a file DESSERT LIST, which is on your A disk, to your R disk, the remote minidisk, issue

   ```
   COPYFILE DESSERT LIST A DESSERT LIST R
   ```

   This results in the file DESSERT LIST being copied from your A disk and being created on your R disk; that is, being uploaded to your remote minidisk.

You download a file similarly, using the CMS copyfile command:

1. LINK to the remote minidisk in read or write mode

2. ACCESS the minidisk (for example as file mode R)

3. Use the CMS COPYFILE command (for example: to download the file DESSERT LIST, which is on your R disk, to your A disk, the local minidisk, issue

   ```
   COPYFILE DESSERT LIST R DESSERT LIST A
   ```

   This results in the file DESSERT LIST being copied from your R disk and being created on your A disk; that is, being downloaded to your local minidisk.

You can also use the COPYFILE command to copy files between two remote minidisks (both on your remote VM system) or between two local minidisks.

Thus, accessing files on your remote VM system is as simple as accessing local VM/PC CMS files.

Remote Spooling

In addition to providing you with remote minidisks and files, VMPCSERV also allows VM/PC to use the printers of the remote VM system. This allows you to spool your local print output to the remote VM system, when the VMPCSERV program is running in your remote VM session. In addition, you may also tag the files to send print files to other printers in your VM network.

To send a print file to your remote VM system, you:

1. Start the VMPCSERV program in your Remote 3277 Session

2. SPOOL your local printer to the correct class

3. (Optionally) issue the appropriate TAG command

4. PRINT the file.

Normally, your local printer is spooled class *. If you change this class, printer files are spooled to your remote VM system, where the VMPCSERV program issues the appropriate printer commands. When you want to use your local printer again, spool your printer class back to *. Printer output will be directed to your local printer.

In addition to specifying a class on your remote printer files, you may also spool your printer to or for another user, in the same way that you might do this on your remote VM system. Chapter 7 contains a full description of the CP SPOOL and TAG commands, which you should consult for further information.

If you start to print a file on your remote system and change the way that the printer is spooled as the file is being printed,

the lines already printed have been sent to the remote printer, and only following lines directed to the printer will be printed as you have requested with the new spool information. This is due to the way the VMPCSERV program works. Next we will take a look at how VMPCSERV works to understand how this is done.

What to Do if Something Goes Wrong

The Remote Server program relies upon your remote CMS session for its operation. If this session should be stopped due to processing errors in the requested remote services, the remote server will also be stopped. For instance, if an I/O error occurs on a remote minidisk while you are using it through the remote server, your remote CMS session may terminate to preserve the integrity of the files on that disk, and the services of the remote server are lost.

If communication between VM/PC and the Remote Server program (VMPCSERV) is interrupted or lost, you must do the following to restart communication:

1. Restart VMPCSERV at the remote session (if VMPCSERV is still running there, press PF3 to stop it, then issue VMPCSERV to start it).

2. Redo the link at the local session.

The remote server will invoke some CMS services in your remote system in such a way that no terminal output can be created by the CMS service. To do this, the remote server spools the output only to a console file, and then purges this file after the CMS service is completed. The exec which invokes the remote server closes any console file you may have so that you will not lose it when the server purges the console file. If the CMS service invoked terminates your remote CMS session, you should issue the command SPOOL CONSOLE TERM STOP. This will once again allow you to see console output in your remote session, and you can re-ipl your remote CMS session.

How VMPCSERV Works

This section is an overview of how VMPCSERV works. You do not need to know how VMPCSERV works to use it to do your work.

When you start VMPCSERV in your remote VM session, all of your CMS minidisks, except your S-disk, are released. These disk modes will be used when you request ACCESSes to be done for you on VM/PC for remote minidisks.

The VMPCSERV program then tries to establish communications with the VM/PC. The VMPCSERV program and the VM/PC communicate by using the 3277-2 Coaxial Connection which connects the VM/PC to the remote system. These communications are performed in the VMPCSERV command by doing Diagnose X'58' reads and writes of the data.

While these communications are taking place, you cannot do any other work on your remote system, since VMPCSERV is running.

When the communication is established, the VMPCSERV program displays a full-screen logo, which tells you that the VMPCSERV Remote Server is active. This logo is displayed on your Remote 3277 Session, until you stop VMPCSERV, to remind you that you have the remote server active.

The VMPCSERV program then waits for requests from VM/PC. The VMPCSERV program can perform a number of different services for you, as we have seen previously. Each type of service requested from the VM/PC is internally identified by a Service Request ID and Sub-function ID. The actual values for these identifiers are set by VM/PC when it requests a remote service for you. The VMPCSERV program recognizes what the request is for, based on these identifiers, and calls the appropriate service within VMPCSERV to service the request.

For instance, if you issued the command "LINK * 191 191" on your local session, and you did not have a locally defined 191 minidisk, VM/PC would send a request for a LINK to the VMPCSERV program. VMPCSERV would perform the

LINK on your remote VM system, and the results of the LINK command would be sent back to you. (Note that LINK * is only valid if the userid of the local and remote systems are the same).

Additionally, if you issued the command "ACCESS 191 Z" on your local CMS session and 191 was a remote minidisk, VM/PC would send an ACCESS request to the remote VM session where the VMPCSERV program would perform the actual ACCESS of the minidisk. The VMPCSERV program uses the same mode letter which you specify to perform the access. Therefore, the VMPCSERV program issues the CMS command "ACCESS 191 Z" and the remote access is done.

Likewise, a file being printed on your remote system uses the remote printer services of the VMPCSERV program. When your local program prints a line, the request is sent to the VMPCSERV program, which invokes the printer services of VMPCSERV to actually print the line on your remote printer. Thus the remote printers are accessible to programs running in your local CMS session.

All remote services are performed for you in your remote session, as requested by you from the local session. Thus if you write to a file on a remote minidisk, the local session sends a request and the data to be written to the VMPCSERV program, which actually performs the write to the file you specified.

The VMPCSERV program continues to wait for requests to be done until you stop it by pressing the PF3 key. At this time, VMPCSERV releases all of the minidisks which you have accessed from the local VM session, and detaches all virtual devices acquired during the time the remote services program was executing. VMPCSERV recorded which disks were accessed when it started execution, and restores those minidisks to your CMS environment by accessing them as you had them accessed originally. VMPCSERV then terminates processing, and your remote CMS session is restored.

Collecting Names of Files Downloaded to VM/PC

Your remote system installation may want to automatically monitor VMPCSERV for particular downloaded file traffic. For instance, they may want to prevent users from copying certain modules to the local VM/PC environment, or keep track of who is using particular files in the VM/PC environment from the host.

Whenever a file system read of record 1 is detected, the HSTFSR module (which is part of VMPCSERV) gives control to a routine named HSTACC (if there is one). A user could write an HSTACC routine to generate an accounting record with the date, time, and fileid of the file being accessed.

Here's a simple example of an HSTACC accounting routine.

```
*****************************************************************
*                                                               *
*       This routine will use CP Diagnose X'4C' to produce      *
*       an accounting record to track reading a file from the   *
*       Remote System.                                          *
*                                                               *
*****************************************************************
HSTACC   ENTER
         LA    R2,ACTTIME          Point to area for the time
         DIAG  R2,R15,X'0C'        Get the date and time of day
         MVC   ACTFNAME,0(R1)      Get the filename
         MVC   ACTFTYPE,8(R1)      Get the filetype
         MVC   ACTFMODE,16(R1)     Get the filemode
         LA    R2,ACTDATA          Get address of account data
         LA    R3,X'10'            Subcode X'10'
         LA    R4,LACTDATA         Length of user information
         DIAG  R2,R3,X'4C'         Issue diagnose for account
         DROP  R1
         SR    R15,R15
         RET
ACTDATA  DS    0D
ACTTIME  DS    CL8                 Date MM/DD/YY
         DS    CL8                 Time HH:MM:SS
ACTFNAME DS    CL8                 Name of file being read
ACTFTYPE DS    CL8                 Type of file being read
ACTFMODE DS    CL2                 Mode of file being read
LACTDATA EQU   *-ACTDATA
         REGEQU
         END   HSTACC
```

Note: ENTER and RET are macros that can be found on the CMS 101 minidisk in VMPCSERV MACLIB.

In this very simple example, when record 1 of a file is requested, HSTFSR calls the HSTACC routine to generate accounting information, which can be collected by the Remote System Installation for analysis. The actual format of the accounting data would not have to match what is done

here, as this is merely an example of what might be done. If you use system accounting services, see the publication *VM/SP System Programmers Guide* for information on user accounting requirements.

The HSTACC routine could even check the filename and filetype against a table of CMS fileids which the installation does not want copied to the local CMS environment. If read access to the file was to be restricted, a bad return code could be returned to the caller of the FSREAD routine in VMPCSERV, to prevent the copying of this particular data.

See the next section for further information.

Setting Up an HSTACC Routine: The HSTACC routine will get control whenever the first record of a Host file is read at the request of the local VM/PC system. This routine is in the main stream of VMPCSERV and must observe certain rules.

VMPCSERV is a module run in the users' area at X'20000' which already restricts the use of many system as well as other CMS functions and programs. The HSTACC routine must preserve the user environments and return them to their original state. If the environment is altered in the routine, it must be restored upon exit. When VMPCSERV terminates after PF3 is pressed, the original user environment is restored as it was before the VMPCSERV EXEC was run.

If there is a secondary entry point (HSTACCFE - accounting final exit) defined in the primary exit routine, it will be given control during VMPCSERV termination. The user can then cleanup and close spool files and undo whatever the accounting exit did. The HSTACCFE entry point will be given control at register 15. The exit must return on register 14. Register 5 will contain the actual return code of the termination of VMPCSERV. Code 3E8 means that the normal exit was taken via PF 3 key being pressed. Code 3E9 means that VMPCSERV is being terminated because of an I/O error. No terminal or disk I/O should be performed. The original user environment will not be restored with this error (3E9). All other registers have no

meaning but the data that they point to should not be destroyed by this user exit routine (for example, register 13 save area).

The HSTACC routine must not issue commands that will allow the system to interrupt the VMPCSERV processing (such as "CP SET MSG ON"). The HOST and the VM/PC Local processors are paced and required to answer each other in a timely sequence. If this sequence is interrupted, the connection will be broken and VMPCSERV will stop. It will be necessary to IPL the HOST CMS to recover the machine.

Remember if you use any spooling type functions in this exit, the local VM/PC user may also be using them. This could be confusing unless you save and restore the current values that the local processor set, prior to exiting from HSTACC each time. Spooling messages are impossible to stop and may interrupt VMPCSERV. Unless the screen is reset in the host machine on time, VMPCSERV may terminate.

Because VMPCSERV preserves the users environment and restores it at the completion of the service process, there are no disks accessed other than the system "S" disk. This is so that the local VM/PC can use all 25 mode letters other than S. It is not recommended that any file I/O be performed in this EXIT. Diagnose "4C" should be used to create accounting data, but it is also possible to accumulate the transactions in buffers, and then send them to the accounting or other machine to be processed. See the description of the second entry point, above, for more information.

HSTACC should only be used to accumulate or interrogate the names of the files that are being requested by the LOCAL processor (VM/PC). The exit will be given control with register one (R1) pointing to the filename, filetype, and filemode. HSTACC could accumulate or filter the names of the downloaded files and send formatted accounting records of them to an accounting machine or use the DIAG X'4C' directly. The Accounting machine could then reduce the data and create a list of programs that are being used by the local VM/PC.

Rules and Restrictions: The exit will be made available to the user if they follow these rules and restrictions when setting up an HSTACC routine.

1. Naming conventions are:

   - Main exit name of the routine is "HSTACC TEXT."

   - Termination exit entry point name is "HSTACCFE."

2. Install the user's exit by linking HSTACC together with other VMPCSERV text decks. No attempt is made to find the HSTACC deck on any TXTLIB or LOADLIB format libraries. Dynamic loading can not be performed. Link HSTACC together with VMPCSERV service module by either:

   Making HSTACC available on a disk when VMPCSERV is installed by the local VM/PC machine.

   -OR-

   Using the EXEC provided (GENPCSER) to load, and generate the module in the host machine first by:

   - Saving the uploaded TEXT decks that make up VMPCSERV.
   - Making the uploaded TEXT decks available when the GENPCSER EXEC is run that produces VMPCSERV module.
   - HSTACC must be in text deck form, and available (ACCESSed) at GENPCSER EXEC time.

     Note: GENPCSER EXEC will be uploaded to the remote system during install of VMPCSERV from the CMS 101 disk.

3. User exit must return on Register 14.

4. User exit does not use the current value in R13 as its save area.

5. Zero register 15 on return, else the read will fail.

6. User exit does not do any I/O to user disks (there are only the ones linked by the Local VM/PC available).

7. User exit preserves and restores any settings of CP, CMS, and spooling that the EXIT may use before each return to VMPCSERV.

8. User exit does not use any of the CP commands that will interrupt the full screen environment.

9. User exit does not issue any error messages to the terminal.

10. User does not use any commands or functions that occupy location X'20,000'.

11. The EXIT does not delay the return to VMPCSERV to interfere with the pacing of the two processors (Acknowledgement is timed).

12. Uses standard CMS Macro interfaces.

It is recommended that you use the DIAGNOSE X'4C' to send the user data to the system defined accounting machine. See the *VM/SP System Programmer's Guide* for information on the use of Diagnose X'4C' and creating accounting records.

The HSTACC routine can use PUNCHC and CP DIAGNOSE commands to set up the environment to get the records to other accounting machines. (Spool classes other than the default ones are suggested.) The routine **MUST NOT** issue any commands that will disrupt the "quiet" environment that is needed to PACE the two processors.

Contents of Parameter Lists: The HSTFSR module will pass the fileid of the file being read to the HSTACC routine in the following format:

```
 0(R1) = CL8'filename'
 8(R1) = CL8'filetype'
16(R1) = CL2'filemode'
```

HSTFSR will pass the following to HSTACC in the Registers.

```
R1  = The data is in the above format.
R13 = Systems save area.
R14 = Return pointer to VMPCSERV at Exit time.
R15 = Entry Pointer for addressability.
HSTACC must clear register 15 or the return code
will be passed back to the local VM/PC
indicating that there was an error if R15 is non-zero.
```

Chapter 7. VM/PC Commands

This chapter describes the VM/PC CP and VM/PC CMS commands.

Entering Commands

A command consists of a command name usually followed by one or more positional operands and, in many CMS commands, by an option list. CP and CMS commands described in this publication are shown in the format:

command name	[operands...] [(options...[)]]

You must use one or more blanks to separate each entry in the command line unless otherwise indicated. For an explanation of the special symbols used to describe the command syntax, see "Notational Conventions."

The Command Name

The command name is an alphameric symbol of one to eight characters. In general, the names are based on verbs that describe the function you want the system to perform. For example, you may want to find out information concerning your CMS files. In this case, you would use the LISTFILE command.

The Command Operands

The command operands are keywords and/or positional operands of one to eight, and in a few cases, one to seven alphameric characters each. The operands specify the information on which the system operates when it performs the command function.

For CMS commands:

1. Write the operands in the order they appear in the command formats, unless otherwise specified.
2. Blanks may optionally be used to separate the last operand from the option list.
3. Use a left parenthesis "(" as the beginning of an option list; it does not have to be preceded by a blank.

The Command Options

CMS command options are keywords used to control the execution of the command. The CMS command formats in "CMS Commands" show all the options for each CMS command.

The option list must be preceded by a left parenthesis; the closing parenthesis is not necessary.

For most commands, if conflicting or duplicate options are entered, the last option entered is the option in effect for the command. Exceptions to this rule are noted where applicable.

Comments in CMS Command Lines

You can enter comments on your console by using the CP * command.

You can write comments with CMS commands by entering them following the closing parenthesis of the option list. The only exception to this rule is the ERASE command, for which comments are not allowed.

Notational Conventions

The notation used to define the command syntax in this publication is:

- Truncations and Abbreviations of Commands

  Where truncation of a command name is permitted, the shortest acceptable version of the command is represented by uppercase letters. (Remember, however, that CMS commands can be entered with any combination of uppercase and lowercase letters.) The following example shows the format specification for the FILEDEF command.

  ```
  Filedef
  ```

  This format means that FI, FIL, FILE, FILED, FILEDE, and FILEDEF are all valid specifications for this command name.

Operands and options are specified in the same manner. Where truncation is permitted, the shortest acceptable version of the operand or option is represented by uppercase letters in the command format box. If no minimum truncation is noted, the entire word (represented by all uppercase letters) must be entered.

Abbreviations are shorter forms of command operands and options. Abbreviations for operands and options are shown in the description of the individual operands and options that follow the format box. For example, the abbreviation for MEMBER in the PRINT command is MEM. Only these two forms are valid and no truncations are allowed. The format box contains

```
MEMBER {name}
       {*   }
```

and the description that follows the format box is

```
MEMBER (name)
MEM    ( *  )
```

- The following symbols are used to define the command format and should never be typed when the actual command is entered.

```
underscore      _
braces         { }
brackets       [ ]
ellipsis       ...
```

- Uppercase letters and words, and the following symbols, should be entered as specified in the format box.

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

- The abbreviations "fn," "ft," and "fm" refer to filename, filetype, and filemode, respectively. The combination "fn ft <fm>" is also called the file identifier or fileid.

  When a command format box shows the characters, fn ft fm or fileid and they are not enclosed by brackets or braces, it indicates that a CMS file identifier must be entered.

  If an asterisk (*) appears beneath fn, ft, or fm, it indicates that an asterisk may be coded in that position of the fileid. The operand description describes the usage of the *.

- Lowercase letters, words, and symbols that appear in the command format box represent variables for which specific information should be substituted. For example, "fn ft fm" indicates that file identifiers such as "MYFILE EXEC A1" should be entered.

- Choices are represented in the command format boxes by stacking:

  A
  B
  C

- An underscore indicates an assumed default option. If an underscored choice is selected, it need not be specified when the command is entered.

  Example:

  The representation:

  A
  B
  C

  indicates that either A, B, or C may be selected. However, if B is selected, it need not be specified. Or, if none is entered, B is assumed.

- The use of braces denotes choices, one of which *must* be selected.

  Example:

  The representation:

  indicates that you *must* specify either A, or B, or C. If a list of choices is enclosed by neither brackets or braces, it is to be treated as if enclosed by braces.

  The use of brackets denotes choices, one of which *may* be selected.

  Example:

  The representation:

  indicates that you may enter A, B, or C, or you may omit the field.

- In instances where there are nested braces or brackets on the text lines, the following rule applies: nested operand selection is dependent upon the selection of the operand of a higher level of nesting.

  Example:

```
 Level 1   Level 2   Level 3
[filename [filetype [filemode]]]
```

  where the highest level of nesting is the operand that is enclosed in only one pair of brackets and the lowest level of nesting is the operand that is enclosed by the maximum number of brackets. Thus, in the previous example, the user has the option of selecting a file by

filename only or filename filetype only or by filename filetype filemode. The user cannot select filetype alone because filetype is nested within filename and our rule states: the higher level of nesting must be selected in order to select the next level (lower level) operand. The same is true if the user wants to select filemode; filename and filetype must also be selected.

- An ellipsis indicates that the preceding item or group of items may be repeated more than once in succession.

  Example:

  The representation:

  ```
  (options...)
  ```

  indicates that more than one option may be coded within the parentheses.

CP Commands

Format of CP Commands

The command formats are described in alphabetical order except for special characters such as the asterisk (*) and the pound sign (#), which appear first.

The command formats are presented in the following order:

1. Command Name: Identifies the name of the command

2. Description of Function: Describes the use

3. Syntax: Lists the syntax of the command with all the possible operands that you can use

4. Operand Description: Describes the function of each operand and any values that you can include in the operand

5. Usage Notes: Contains notes about special uses of the command, its operands, or combinations of commands or operands.

6. Responses: Describes the CP responses sent to the terminal, caused by execution of the command. These responses are command responses and are not to be construed as VM/PC system messages. The command responses are not prefixed and, therefore, they are not contained in Chapter 8, "VM/PC Messages" on page 8-1.

Note: The execution of certain commands may produce system messages. Refer to Chapter 8, "VM/PC Messages" on page 8-1 for information on system message formats and user action.

Summary of CP Commands

The following is a brief description of all the commands acceptable to the CP component of the VM/PC system.

Command	Usage
*	Annotate the terminal display screen
#CP	Execute a CP command while remaining in the virtual machine environment.
ATTN	Make an attention interruption pending for the virtual machine console.
BEGIN	Continue or resume execution of your virtual machine at either a specific storage location or at the address in the current PSW.
CHANGE	Alter one or more attributes of a closed spool file.
CLOSE	Terminate spooling operations on the virtual machine printer.
CP	Execute a CP command while remaining in the CMS virtual machine environment.
DEFINE	Reconfigure your virtual machine.
DETACH	Detach a device from your virtual machine.
DISPLAY	Display virtual storage on your terminal.
DUMP	Print the following on the virtual machine printer: virtual PSW, general registers, floating-point registers, and contents of specified virtual storage locations.
EXTERNAL	Simulate an external interruption for your virtual machine and return control to that machine.
IPL	Simulate IPL of the virtual machine.

Figure 7-1 (Part 1 of 2). Summary of CP Commands

Command	Usage
LINK	Provide access to a specific minidisk by a virtual machine.
LOGOFF	Disable access to CP.
ORDER	Rearrange closed spool files in a specific order.
PURGE	Remove closed spool file from system.
QUERY	Request information about virtual machine configuration and system status.
REQUEST	Make an attention interruption pending for the virtual machine console.
SET	Control various functions within the virtual machine.
SPOOL	Alter spooling control options; direct a file to the remote system.
STORE	Alter specified virtual storage locations and registers.
TAG	Specify variable information to be associated with a spool file. Interrogate the current TAG text setting
TERMINAL	Define or redefine the input and attention handling characteristics of your virtual machine console.
TRACE	Trace specified virtual machine activity at your terminal, spooled printer, or both.

Figure 7-1 (Part 2 of 2). Summary of CP Commands

CP Commands and Options

*

Use the asterisk (*) to annotate the terminal display screen data with a comment. The format of the * (comment) command is:

*	anycomment

#CP

Use the #CP command to execute a CP command while in a virtual machine command environment without first signaling attention to get to the CP command environment. The format of the #CP command is:

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

Where:

commandline
: specifies the name and operands for the CP command or commands you want to issue. You must precede the first commandline with at least one blank.

Usage Notes:

1. The pound sign (#) shown in the preceding format represents the logical line end symbol and is the default supplied by VM/PC. However, you may choose to redefine the logical linend symbol. If you have redefined the logical line end symbol, #CP is an invalid command; you must substitute your line end symbol for the pound sign when using this command.

2. For the command to operate, the following conditions must be met:

- The virtual machine must be operating with SET LINEDIT ON (a default).

- The first three characters of the edited line must be #CP (uppercase or lowercase) with the "#" representing the logical line end symbol currently defined.

- At least one blank must separate the #CP from any command line. Do not use attention interruption in any part of the line or to enter the line.

3. You can enter multiple command lines as operands of the #CP command provided that all command lines to be executed must be separated by the current logical line end symbol.

4. If you enter #CP with no operands, the virtual machine enters the CP environment.

Responses: If you enter #CP without a commandline, you receive no response.

If you enter #CP with commandlines, you receive the responses appropriate to the individual commands you entered.

ATTN

Use the ATTN command to make an attention interruption pending at your virtual console. The format of the ATTN command is:

ATTN	

Usage Notes:

1. The REQUEST command performs the same function as ATTN; the two commands can be used interchangeably.

2. The BEGIN command is not required after you issue ATTN.

3. CP ignores the ATTN and REQUEST commands if an I/O operation is in progress at the console or if other interrupts are pending for your virtual machine. This condition may result if you issue the ATTN or REQUEST command by means of DIAGNOSE X'8'.

BEGIN

Use the BEGIN command to continue or resume execution in the virtual machine at either a specified storage location or the location pointed to by the virtual machine's current program status word (PSW). The format of the BEGIN command is:

Begin	[hexloc]

Where:

hexloc
: is the hexadecimal storage location where execution is to begin.

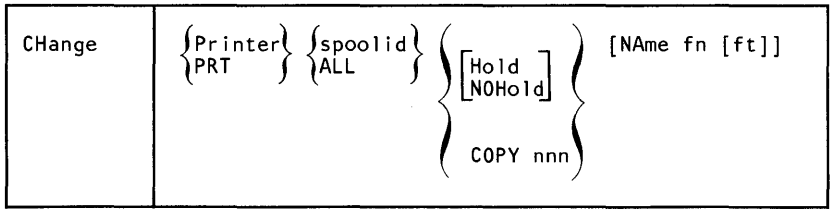
Usage Notes:

1. When BEGIN is issued without hexloc, execution begins at the storage address pointed to by the current virtual machine PSW. Unless the PSW has been altered since the CP command mode was entered, the location stored in the PSW is the location where the virtual machine stopped.

2. When BEGIN is issued with a storage location specified, execution begins at the specified storage location. The specified address replaces the instruction address in the PSW, then the PSW is loaded.

CHANGE

Use the CHANGE command to alter one or more of the attributes for a closed spool file. The format of the CHANGE command is:



Where:

PRINTER
PRT
: changes printer files.

spoolid
: designates the spoolid of file to be changed.

ALL
: changes all spool files.

HOLD
: prevents the spool file being selected from the spool queue until released. The file is released by issuing a CHANGE command with the NOHOLD operand.

NOHOLD
: releases the file from HOLD status.

COPY nnn
: specifies the number of copies of the spool file you want. The value of nnn (number of copies) must be a number from 1 to 255; leading zeros are optional.

NAME fn [ft]
: assigns identification to the file in the CMS format of filename and filetype. If ft is not specified, it is set to blanks.

Usage Notes:

1. Issue the QUERY command to determine the current attributes of the file

2. To CHANGE a file, it must have been closed but not yet selected from the spool queue.

3. When issued from a local session, the CHANGE command only applies to files that are on the local spool queue. You must issue the VM/SP CHANGE command from the remote session to change files that are on the remote spool queue.

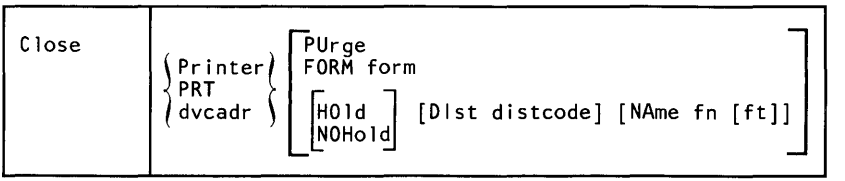
4. Spool Files whose corresponding data file cannot be found will be placed in HOLD status and cannot be changed to NOHOLD.

Response:

nnnn FILE(S) CHANGED
: This response indicates the number of files changes; if none are changed nnnn will be NO. This response will not appear if you have issued the CP SET IMSG OFF command.

CLOSE

Use the CLOSE command to indicate that the current spool data is to be collected into a spool file. The format of the CLOSE command is:



Where:

PRINTER
PRT
: closes the printer spool file.

dvcadr
: closes the specified spool device.

PURGE
: immediately purges the spool file. No spool file is created.

`FORM form`
specifies the form name for the spool file. The form name is a 1-8 alphanumeric value. If FORM is not specified, the file is created with the form name indicated by the last SPOOL command.

`HOLD`
overrides the SPOOL NOHOLD option and places the spool file in HOLD status. The file is released by issuing a CHANGE command with the NOHOLD operand.

`NOHOLD`
overrides the SPOOL HOLD option making the file immediately selectable for spool processing.

`DIST [distcode]`
specifies the distribution code the spool file is to be given.

`NAME fn [ft]`
assigns identification to the file in the CMS format of filename and filetype. If ft is not specified, it is set to blanks.

Usage Notes:

1. The HOLD/NOHOLD options take precedence over the current spool attributes.

2. The CLOSE command has no effect if the CONT is one of the current spool attributes.

Responses:

`PRT FILE spoolid SPOOLED`
This response indicates the spoolid created. This response will not appear if you have issued the CP SET IMSG OFF command.

`PRT FILE SPOOLED TO|FOR userid`
This response indicates the spool file has been routed to the Remote Server. This response will not appear if you have issued the CP SET IMSG OFF command.

CP

Use the CP command as an optional header on commands to CP in a CP environment. The format of the CP command is:

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

Where:

commandline
: specifies the name and operands for the CP command or commands you want to issue. You must precede the first commandline with at least one blank.

Usage Notes:

1. The pound sign (#) shown in the format block represents the logical line end symbol currently in effect for your virtual machine. If you have redefined the logical line end symbol, you must substitute your line end symbol for the pound sign when using this command.

2. The CP command is treated as a null line by the control program and therefore can precede any other command if one or more blanks separate CP from the other command.

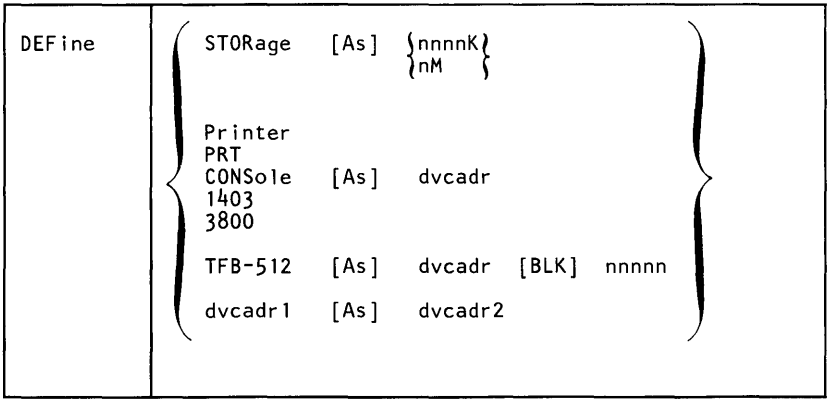
3. You can enter multiple command lines as operands of the CP command, but you must separate each command line by the logical line end (#) symbol.

Responses: If you enter CP without a commandline in the CP environment, you receive no response.

If you enter CP with commandlines, you receive the responses appropriate to the individual commands you entered.

DEFINE

Use the DEFINE command to alter your virtual machine configuration. The format of the DEFINE command is:



Where:

`STORAGE [AS] {nnnnK / nM}`

redefines the size of virtual storage allocated for the virtual machine as nnnnK (where K represents 1024 bytes) or nM (where M represents 1,048,576 bytes). If the virtual storage size is not specified as a multiple of 4k, it is rounded up to the next 4k boundary. Changing the size of virtual storage, causes the virtual machine to be reset.

`PRINTER [AS] dvcadr`
`PRT`
`1403`

redefines the virtual machine printer as a 1403 with address dvcadr.

`3800 [AS] dvcadr`

redefines the virtual machine printer as a 3800 with address dvcadr.

`CONSOLE [AS] dvcadr`

redefines the virtual machine console with address dvcadr.

`TFB-512 [AS] dvcadr [BLK] nnnnn`

adds a temporary minidisk with device address dvcadr to the virtual machine configuration. If the keyword AS is omitted, dvcadr may not be A or AS.

BLK nnnnn specifies the number of 512 bytes blocks to be allocated on the minidisk.

```
dvcadr1 [AS] dvcadr2
```
redefines the device represented by dvcadr1 as dvcadr2

Usage Notes:

1. When you alter the virtual machine configuration with the DEFINE command, the changes are temporary and are in effect for the current terminal session only. The configurator must be used to make permanent alteration to the user's configuration.

2. The maximum amount of virtual storage that can be defined is 4M.

3. If a DEFINE STORAGE is requested and sufficient DASD space is not available, the virtual storage will be defined as large as possible, and the response will indicate how much is available.

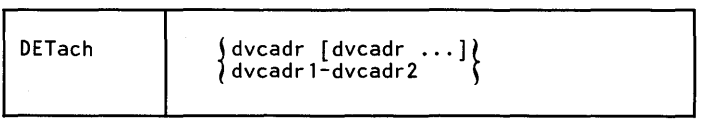
4. If the virtual machine printer type (1403/3800) is changed and a spool file is currently open, the spool file will be closed.

Responses:

```
STORAGE = nnnnK
DEV dvcadr DEFINED
```
These responses will not appear if you have issued the CP SET IMSG OFF command.

DETACH

Use the DETACH command to remove a device from the virtual machine configuration. The format of the DETACH command is:



Where:

```
dvcadr [dvcadr ...]
dvcadr1-dvcadr2
```

dvcadr is the address of the device to be removed from the virtual machine configuration. To detach multiple devices, a blank is used to separate the device addresses. To detach a range of devices, a hyphen must be inserted between the addresses specified. Multiple addresses and a range of addresses cannot be intermixed on the same command line.

Usage Notes:

1. You can detach a device that was temporarily defined by the DEFINE command, or one that is part of your permanent virtual machine configuration, except for the terminal or printer devices.

2. You can detach a device even if it is currently in use.

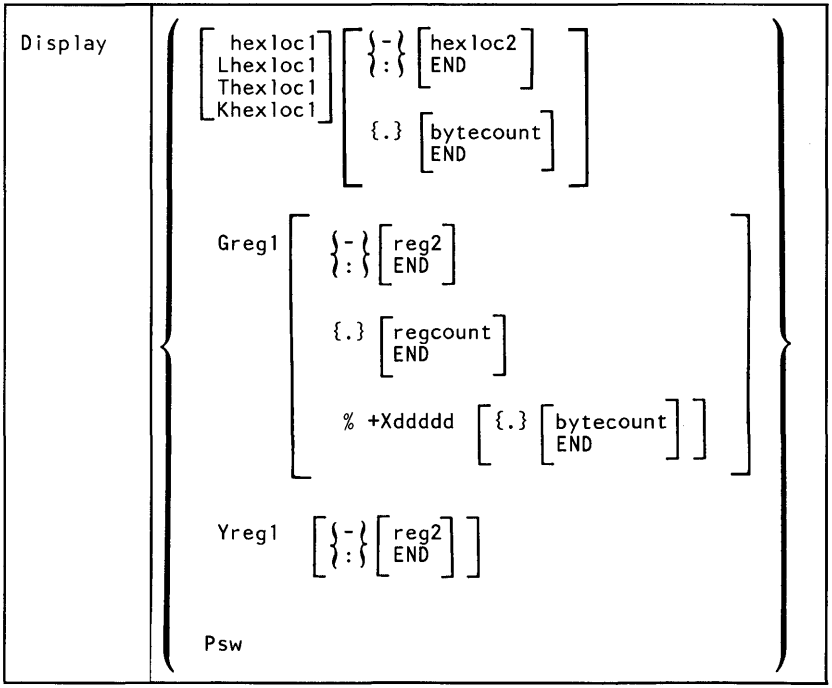
Response:

```
DEV dvcadr DETACHED
```

This response will not appear if you have issued the CP SET IMSG OFF command.

DISPLAY

Use the DISPLAY command to display the following virtual machine components. The format of the DISPLAY command is:



Where:

hexloc1
Lhexloc1
Thexloc1
Khexloc1
: is the first or only hexadecimal storage location in the virtual machine that is to be displayed. If L or no letter prefix is specified, the storage contents are displayed in hexadecimal. If T is specified, the storage contents are displayed along with the EBCDIC translation. If K is specified, the storage keys are displayed in hexadecimal.

If hexloc1 is not on a fullword boundary, it is rounded down to the next lower fullword.

If L or T is specified without any operand, or followed immediately by a blank, the contents of all virtual storage is displayed.

The address may be from 1 to 6 hexadecimal digits; leading zeros are optional.

{-} hexloc2
{:} END

is the last of a range of hexadecimal storage locations in the virtual machine to be displayed. If - or : is the first character of the operand, then hexloc1 defaults to zero. If hexloc2 is not specified, storage is displayed from hexloc1 to the end of virtual storage. If specified, hexloc2 must be greater than hexloc1 and within the virtual storage size.

The address may be from 1 to 6 hexadecimal digits; leading zeros are optional.

{.} bytecount
END

is the hexadecimal integer designating the number of bytes of storage to be displayed. The sum of hexloc1 and bytecount must be an address that does not exceed the size of the virtual machine.

The value may be from 1 to 6 hexadecimal digits; leading zeros are optional.

Greg1

is a decimal number from 0 to 15 or a hexadecimal number from 0 to F representing the first, or only, general register whose contents are to be displayed. If G is specified without a register number, the contents of all general registers are displayed.

Yreg1

is an integer (0, 2, 4, or 6) representing the first, or only, floating-point register whose contents are to be displayed. If Y is specified without a register number, the contents of all floating-point registers are displayed.

{-} reg2
{:} END

is the last of a range of registers to be displayed. If - or : is the first character of the operand, reg1 defaults to zero. If reg2 is not specified, all registers of this type from reg1 are displayed. If specified, reg2 must be equal to or greater than reg1. For

general registers, reg2 may be a decimal number from 0 to 15 or a hexadecimal number from 0 to F; for floating-point registers, reg2 may be 0, 2, 4, or 6.

`{.} regcount`
`END`
is a decimal number from 1 to 16 or a hexadecimal number from 1 to 10 designating the number of registers to be displayed. The sum of Greg1 and regcount may not exceed 16; for floating-point registers regcount may not exceed 4.

`[%[+Xddd]]`
uses reg1 as an address with the optional hexadecimal displacement ddd.

`PSW`
displays the current virtual machine PSW (program status word) as two hexadecimal words.

Usage Notes:

1. This command displays only second-level storage (storage which is real to the virtual machine).

2. Multiple operands on a single command line will not be supported.

3. To terminate the DISPLAY function while data is being displayed, press the PA1 key.

Responses: One of the following responses is displayed depending on the operand specified.

Displaying Storage Locations

xxxxxx word1 word2 word3 word4 [*EBCDIC Translation*]
response when displaying storage locations; xxxxxx is the storage location of word1. Up to 4 words are displayed on a line with the optional translation (periods represent translation of nonprintable characters). Multiple lines are used for a range of locations. If translation is requested, alignment is made to the next lower 16-byte boundary; otherwise alignment is to the next lower fullword boundary.

xxxxxx TO yyyyyy SUPPRESSED LINE(S), SAME AS ABOVE...
response when displaying storage locations that contain the same data as that displayed on the previous line; xxxxxx is the first storage location whose data display is suppressed; yyyyyy is the last location.

Displaying Storage Keys

xxxxxx TO yyyyyy KEY = kk
response when displaying storage keys; xxxxxx and yyyyyy are the starting and ending storage addresses and kk is the associated storage key.

Displaying General Registers

GPR n = reg1 reg2 reg3 reg4
response when displaying general registers; n is the register whose contents is reg1. Up to 4 registers are displayed on a line for a range of registers; multiple lines are used if required.

Displaying Floating-point Registers

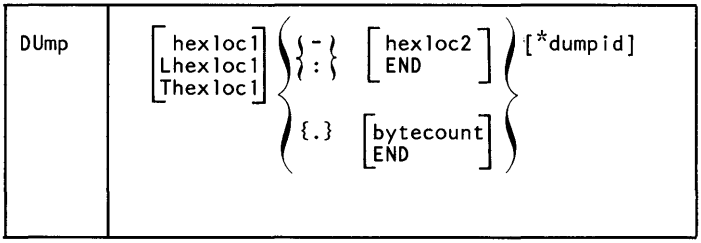
FPR n = xxxxxxxxxxxxxxxx .xxxxxxxxxxxxxxxx E xx
response when displaying floating-point registers; n is the register being displayed. The register is displayed in both the internal hexadecimal format and the E format. One register is displayed per line; multiple lines are used for a range of registers.

Displaying the PSW

PSW = xxxxxxxx xxxxxxxx
response when displaying the PSW; the PSW is displayed as two hexadecimal words.

DUMP

Use the DUMP command to print the contents of various virtual machine components. The format of the DUMP command is:



Where:

hexloc1
Lhexloc1
Thexloc1

is the first or only hexadecimal storage location in the virtual machine that is to be dumped. If L or no letter prefix is specified, the storage contents are dumped in hexadecimal. If T is specified, the storage contents are dumped along with the EBCDIC translation.

If hexloc1 is not on a boundary divisible by 32, it is rounded down to the next lower such boundary.

If L or T is specified without any operand, or followed immediately by a blank, the contents of all storage is dumped.

The address may be from 1 to 6 hexadecimal digits; leading zeros are optional.

{-} hexloc2
{:} END

is the last of a range of hexadecimal storage locations in the virtual machine to be dumped. If - or : is the first character of the operand, then hexloc1 defaults to zero. If hexloc2 is not specified, storage is dumped from hexloc1 to the end of virtual storage. If specified, hexloc2 must be greater than hexloc1 and within the virtual storage size.

The address may be from 1 to 6 hexadecimal digits; leading zeros are optional.

{.} bytecount
END
is the hexadecimal integer designating the number of bytes of storage to be dumped. The sum of hexloc1 and bytecount must be an address that does not exceed the size of the virtual machine. If this address is not on a fullword boundary, it is rounded up to the next highest fullword.

The value may be from 1 to 6 hexadecimal digits; leading zeros are optional.

*dumpid
can be entered for descriptive purposes, and must include the asterisk (*) prefix. If specified, the dumpid is placed on the first output line.

Usage Notes:

1. This command displays only second-level storage (storage which is real to the virtual machine).

2. The CP DUMP command executes in an area of storage separate from your virtual machine and does not destroy any portion of your storage.

3. Use #CP DUMP or press PA1 before entering the command to preserve the virtual machine's register's. Otherwise, the registers will be those of the VM/PC CMS command processor.

4. The DUMP command dumps the virtual PSW and registers. If only this information is required, at least 1 virtual address must be specified, such as:

```
DUMP 0
```

5. The CLOSE command must be issued to create the spool file.

6. To terminate the dump prematurely, press the PA1 key.

Responses:

`DUMPING LOC hexloc`
where hexloc is a 64K boundary address (e.g. 020000, 030000, ...).

`DUMP COMPLETE`
indicates normal completion of the DUMP command.

EXTERNAL

Use the EXTERNAL command to simulate an external interrupt to the virtual machine and return control to that machine.

The format of the EXTERNAL command is:

Where:

`code`
`40`
is the interrupt code, a hexadecimal number to be associated with the external interrupt. All codes within the range X'01' through X'FF' are valid. The default is the external interrupt button on the system console, X'40'.

Usage Note: This simulates pressing the interrupt key on the real computer console, or other functions which cause an external interrupt. Control is given to the virtual machine immediately.

IPL

Use the IPL command to perform the initial program load function for the virtual machine. The format of the IPL command is:

```
Ipl  CMS   [parm string]
```

Where:

`parm string`
passes a parameter string to CMS in the virtual machine's general registers. The string is padded on the right with blanks to make 64 bytes (which fills the 16 registers). If parm string is not specified, the general registers are cleared to zeros.

Usage Notes:

1. The IPL command will always clear virtual storage to zeros before performing the program load.

2. See the CMS documentation for definition of valid parameter strings.

3. If an automatic environment IPL is performed (either from the logon screen or the configuration file), the string 'AUTOIPL' will be loaded.

LINK

Use the LINK command to request access to a minidisk defined in another user's configuration. The format of the LINK command is:

```
LINK   [To]   userid dvcadr1  [As]  dvcadr2  [mode]
              [[PASS=]  password]  [LOCAL ]
                                   [REMOTE]
```

Where:

`userid`
is the name of the user whose configuration is to be searched for minidisk dvcadr1. Use * to indicate that your own userid is to be used.

If the keyword TO is omitted, the userid may not be T or TO.

dvcadr1
: the device address searched for in the configuration for userid.

[AS] dvcadr2
: is the device address to be temporarily added to your virtual machine configuration.

  If the keyword AS is omitted, dvcadr2 may not be A or AS.

mode
: is the access mode. Valid access modes are:

Mode	Meaning
R	Read-only (R/O) access. If the link is REMOTE, the link is not done if any other user has write access.
RR	Read-only (R/O) access. If the link is REMOTE, the link is established even if any other user has write access.
W	Write access. If the link is REMOTE, the link is not done if the mini-disk is accessed by any other user in either read or write mode.
WR	Write access with acceptable reduction to read access. If the link is REMOTE, a write access will be established unless any other user has either read or write access; in this case, read access will be established.
M	Write access. If the link is REMOTE, the link is not done if the mini-disk is accessed by any other user in write mode.
MR	Write access with acceptable reduction to read access. If the link is REMOTE, a write access will be established unless any other user has write access; in this case, read access will be established.
MW	Write access. If the link is REMOTE, write access will be established regardless of any other user access.

Caution: More than one user writing to the same virtual device can result in permanent loss of data.

If the mode is omitted, the default is R if the userid is for another user. If you are linking to one of your own disks, the default mode is the one defined in your configuration.

`[PASS=] password`
: is the one to eight character string that must match the access mode password for the dvcadr1 specified (this should not be confused with your user password).

`LOCAL`
: the LINK is to be resolved in the local system configuration file.

`REMOTE`
: the LINK is to be resolved in the remote system directory.

Usage Notes:

1. If you link to one of your own minidisks, the password is not required.

2. If REMOTE is specified, the Remote Server must be active for the link to be established.

3. If neither LOCAL nor REMOTE is specified, the local configuration file is searched. If the link is not resolved locally and the Remote Server utility is active, the link resolution will be attempted at the remote system.

Responses:

`ENTER READ PASSWORD:`
: enter the read password to obtain read access.

`ENTER WRITE PASSWORD:`
: enter the write password to obtain write access.

`DEV dvcadr2 LINKED`
: indicates that access has been established to the requested minidisk. This response will not appear if you have issued the CP SET IMSG OFF command.

LOGOFF

Use the LOGOFF command to terminate the current session. The format of the LOGOFF command is:

```
LOGoff
LOGout
```

Usage Notes:

1. This command closes any active spool files, and relinquishes access to any temporary minidisks.

Response:

The VM/PC logo reappears on the display when the logoff is complete allowing another logon to be performed.

ORDER

Use the ORDER command to place your spool files in a specific order. The format of the ORDER command is:

```
ORDer   {Printer}  spoolid1 spoolid2 ...
        {PRT    }
```

Where:

`PRINTER`
`PRT`
places printer files in the specified order.

`spoolid1 spoolid2 ...`
are the spoolids of the files to be rearranged and the order in which they are to be placed.

Usage Notes:

1. Spool files selected by the ORDER command will be placed in the specified order ahead of those files not selected. The files not selected will remain in their same relative order.

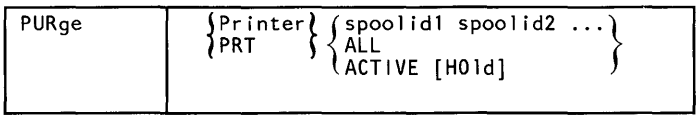
2. When issued from the local session, the ORDER command only applies to files on the local spool queue. You must issue the VM/SP ORDER command to order files that are on the remote spool queue.

Response:

`nnnn FILE(S) ORDERED`
This response indicates the number of files ordered; if none are ordered nnnn will be NO. This response will not appear if you have issued the CP SET IMSG OFF command.

PURGE

Use the PURGE command to erase your spool files before they are selected for processing, or to terminate processing of an active file. The format of the PURGE command is:



Where:

`PRINTER`
`PRT`
purges printer files.

`spoolid1 spoolid2 ...`
are the spoolids of the files to be purged.

`ALL`
purges all files for the type specified.

`ACTIVE [HOLD]`
purges the file currently being processed for the type specified. The HOLD option terminates processing of the file but does not erase the file; the file is retained and is placed in HOLD status.

Usage Note:

1. When issued from a local session, the PURGE command

only applies to files on the local spool queue. You must issue the VM/SP PURGE command from the remote session to purge files that are on the remote spool queues.

Response:

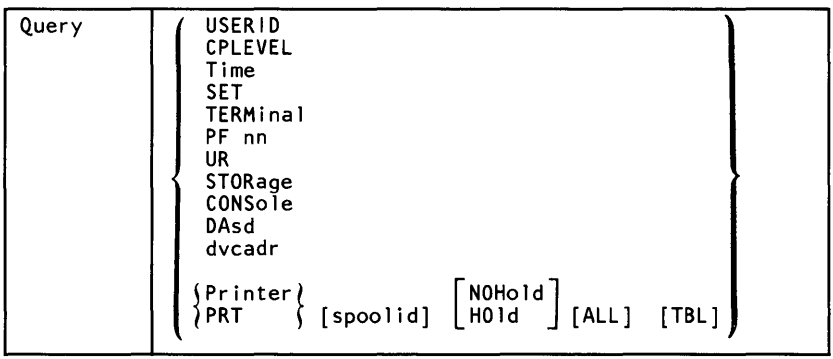
nnnn FILE(S) PURGED
: This response indicates the number of files purged; if none are purged nnnn will be NO. This response will not appear if you have issued the CP SET IMSG OFF command.

QUERY

Use the QUERY command to inquire about the following information:

- The time of day.
- Attributes of closed spool files not currently being printed.
- The current values of SET functions.
- The current values of TERMINAL functions.
- The spool attributes of the printer.
- The current setting of program function keys.

The format of the QUERY command is:



Where:

USERID
: displays the userid of the active user.

CPLEVEL
displays the VM/PC CP service level.

TIME
displays the current time and date. These values depend on the time and date that reside in DOS and may have been set prior to entering the S/370 environment.

SET
displays the status of the SET command options.

TERMINAL
displays the status of the TERMINAL command options.

PF[nn]
displays the operand of the specified program function key. The value of nn may be from 1 to 12. If nn is not specified, all function operands are displayed.

UR
displays the status of the console and printer.

STORAGE
displays the size of your virtual storage.

CONSOLE
displays the status of the console.

DASD
displays the status of all of the minidisks currently in the virtual machine configuration.

dvcadr
displays the status of the device dvcadr.

PRINTER
PRT

displays an information line for each spool file not currently being printed.

spoolid

selects spool information for a specific file on the local spool queue.

NOHOLD

selects spool information for all files that are not in HOLD status.

HOLD

selects spool information for all files that are in HOLD status.

ALL

displays additional information for spool files.

TBL

displays additional information about spool file created for the 3800 printer.

Responses:

Query Userid

```
userid
```

Query CPLEVEL

```
VM/PC, VERSION n.nn, SERVICE LEVEL nnn
```

Query Time

```
TIME IS hh:mm:ss ON weekday mm/dd/yy
```

The current time and date are displayed based on the values set when DOS was loaded.

Query Set

```
                                        ┌ON  ┐
    ┌ON ┐            ┌ON ┐               │OFF │          ┌ON ┐
RUN └OFF┘,   LINEDIT └OFF┘,   EMSG      │CODE│,   IMSG  └OFF┘
                                        └TEXT┘
```

Query Terminal

```
LINEND {c|OFF}, LINEDEL {c|OFF}, CHARDEL {c|OFF}, ESCAPE {c|OFF}
HILIGHT {ON|OFF}, PA2 {FULL|HALF}, MODE {VM|CP}, WRAP {ON|OFF}, APL {ON|OFF}
```

Query PF[nn]

```
PFnn {IMMED|DELAYED|RETRIEVE|UNDEFINED} [pfdata1#...]
```

Query Storage

```
STORAGE=nnnnK
```

Query DASD

```
DASD dvcadr {FB-512 drive|REMOTE} {R|W} [nnnnn BLK] [userid dvcadr2|**TEMP**]
```

This information line is displayed for each minidisk in your virtual machine configuration.

Where:

`dvcadr`
is the address of the minidisk.

`FB-512`
indicates that the mini-disk resides locally.

`REMOTE`
indicates that the mini-disk resides on the remote system.

`W`
`R`
indicates the read/write status of the minidisk.

`drive`
is the Personal Computer disk/diskette drive on which the mini-disk resides.

nnnnn
is the number of blocks allocated to the minidisk.

userid dvcadr2
indicates the userid and device address linked to.

TEMP
indicates that this is a temporary disk.

Query dvcadr

```
PRT dvcadr CL c contatr holdatr COPY nnn READY FORM form
    dvcadr TO|FOR userid DIST distcode FLASHC ccc LINECT lll sepatr
    dvcadr FLASH ovly CHAR ctab  ctab ctab ctab    MDFY cmod FCB pcp
```

Where:

c
is the spool class.

contatr
is the continuation attribute, CONT or NOCONT.

holdatr
is the hold attribute, HOLD or NOHOLD.

nnn
is the number of copies to be produced.

form
is the form type to be used when the file is printed.

distcode
is the distribution code

ccc
is the number of copies to be superimposed with the overlay frame.

lll
is the number of lines per page.

sepatr
is the separator page attribute, SEP or NOSEP.

ovly
is the name of the forms overlay frame to be used.

ctab
: is the name of the character arrangement table to be used.

cmod
: is the name of the copy modification module to be used on output text.

pcpi
: is the name of the FCB module to be used in the vertical formatting of a page.

  CONSOLE dvcadr

  for keyboard/display

  see Query Dasd for minidisk format.

Query Printer:

```
ORIGINID FILE CLASS RECORDS CPY HOLD FORM
userid   id   a typ norecs  nnn stat form

ORIGINID FILE CLASS RECORDS CPY HOLD DATE  TIME     NAME     TYPE     DIST
userid   id   a typ norecs  nnn stat mm/dd hh:mm:ss name     type     dist

ORIGINID FILE CLASS RECORDS CPY HOLD FLASH  CHARS  FCB   MDFY  FLSHC
userid   id   a typ norecs  nnn stat ovly   ctab   plpi  cmod  ccc
```

REQUEST

Use the REQUEST command to make an attention interrupt pending at your virtual console. The format of the REQUEST command is:

```
REQuest
```

Usage Notes:

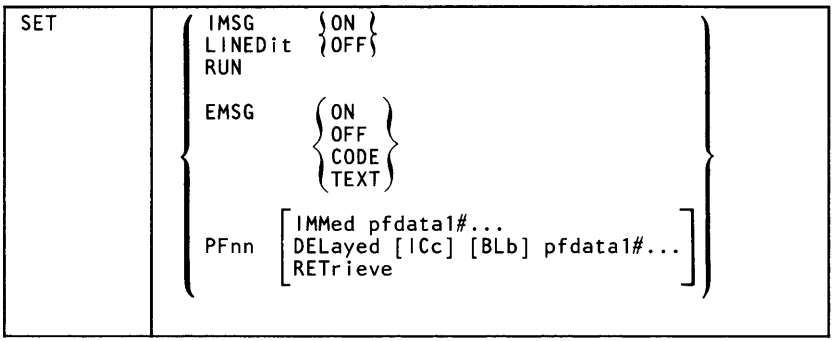
1. The REQUEST command performs the same functions as ATTN and the two commands can be used interchangeably.

2. CP ignores the REQUEST command if any I/O operation is in progress at the console or if other

interrupts are pending. This condition may result if the user issues the REQUEST command by means of DIAGNOSE X'8'.

SET

Use the SET command to control various functions of your virtual machine. The format of the SET command is:



Where:

IMSG {ON / OFF}
: controls whether certain informational messages issued by the CP CHANGE, CLOSE, DEFINE, DETACH, ORDER, PURGE and SPOOL commands are displayed or not. The descriptions of the CP commands tell which messages are affected. If ON is specified, the messages are displayed; if OFF, they are not. When you log on, IMSG is set ON.

LINEDIT {ON / OFF}
: controls whether line editing is active or not. ON specifies that keyboard input is to be scanned for line editing characters; OFF, that no line editing is to be performed. When you log on, LINEDIT is set ON.

RUN {ON / OFF}
: controls the termination of the CP command environment. ON specifies that the CP command environment automatically ends after the execution

of the CP command entered; OFF specifies that the CP command environment continues until terminated with the BEGIN command.

```
EMSG { ON   }
     { OFF  }
     { CODE }
     { TEXT }
```

controls error message handling. ON specifies that both error code and text are to be displayed; TEXT, that only the text is to be displayed; CODE, that only the code is to be displayed; OFF that no error messages are to be displayed. When you log on, EMSG is set TEXT.

```
PFnn [IMMED  pfdata1#...                          ]
     [DELAYED  [ICc]  [BLb]  pfdata1#...]
```

defines the function to be performed for the specified program function key. If specified with no operand, the key will be considered to be undefined.

In PFnn, nn is a number from 1 to 12 that corresponds to a key on the keyboard. The operand defines the function(s) to be performed when you press that key.

IMMED indicates that the function(s) is(are) to be performed immediately after pressing the key. DELAYED indicates that the pfdata#... is to be written into the display input area, and will not be executed until you press the ENTER key. If neither is specified, DELAYED will be assumed.

ICc defines the character c which will be replaced with an insert cursor control character; only the first occurrence of c will be replaced. If ICc is not specified with the DELAYED option, the cursor will be placed at the end of the data.

BLb defines the character b which will be replaced by blanks; all occurrences of b will be replaced by blanks.

`pfdata1#...`
defines the command(s) that constitute the function to be performed. If more than one command is to be executed, the commands must be separated by a logical line end character (defaults to #).

`PFnn RETRIEVE`
specifies that input data is to be retrieved from the terminal activity history buffer when the function key is pressed.

In PFnn, nn is a number from 1 to 12 that corresponds to a function key on the keyboard.

If the cursor is located in the input area of the screen when the function key is pressed, the previously entered input is written into the input area on the screen. If the function key is pressed again, without pressing the ENTER key, the input previous to that just placed into the input area is written into the input area. When the least recent input data available in the activity history buffer is in the input area, pressing the function key will cause the most recent input data to be displayed again.

If the cursor is located on a line displaying previously entered input when the function key is pressed, that input is written into the input area (previously entered input may be displayed by scrolling). Then, leaving the cursor in the input area and pressing the function key, the input previous to the one just written to the input area will be displayed.

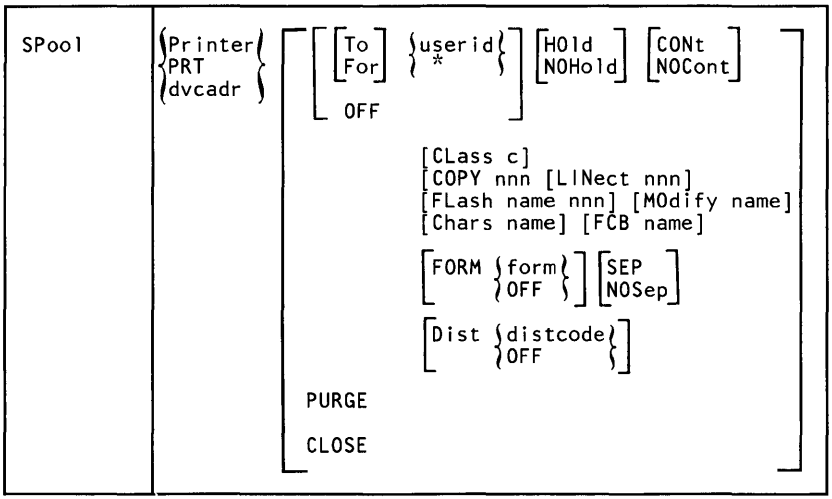
Usage Notes:

1. The ICc and BLb options for PFnn are intended to be used when the command is executed via Diagnose 8 (for example, from a CMS EXEC).

2. For the RETRIEVE function, the amount of input contained in the activity history buffer depends on the amount of both input and output (however, it will generally be the case that large amounts of output to the display will be the factor limiting the number of input lines remaining in the buffer).

SPOOL

Use the SPOOL command to alter the attributes that a spool file will be given when it is closed. You can direct spool files to remote locations by using the SPOOL command in conjunction with the TAG command. The format of the SPOOL command is:



Where:

PRINTER
PRT
: closes the printer spool file.

dvcadr
: modifies attributes for printer device.

TO userid
: specifies the userid to receive the spool file. This option can only be specified if CLASS is not *.

FOR {userid / *}
: specifies the userid under which the spool file is to be created. The userid becomes the owner of the spool file. This option can only be specified if CLASS is not *.

HOLD
places the spool file in hold status when the file is closed. This means that the system cannot select the file for processing until it is released by issuing a CHANGE command for the file. If HOLD is specified, NOHOLD may not be specified.

NOHOLD
indicates that a spool file will be selectable for immediate processing when it is closed. If NOHOLD is specified, HOLD may not be specified.

CONT
indicates that continuous spooling is to be performed (i.e. that CLOSE command requests are to be ignored). If CONT is specified, NOCONT may not be specified.

NOCONT
indicates that continuous spooling is to be reset. If NOCONT is specified, CONT may not be specified.

COPY nnn
specifies the number of copies to be processed when the spool file is closed.

LINECT nnn
specifies the number of lines per page on the output form. VM/PC CP uses this to format pages for the DUMP and TRACE commands.

CLASS c
specifies the spool class for the device. If the value of c is A-Z or 0-9, the spool file will be routed to the Remote Server. If the value of c is *, the spool file will be routed to the local printer.

FLASH name nnn
signifies that a forms overlay contained in the 3800 printer is to be superimposed onto the output. The name, one to four characters, identifies the overlay to be used. The number, nnn, specifies the first nnn copies are to superimposed; the remaining copies made will remain unflashed. If nnn is specified as 0, the default, no superimposing will be done.

MODIFY name
: assigns a copy modification module to be used when printing a file on the 3800. This function makes it possible to alter the text by either preventing printing of certain information or by adding labels to the output. The name must be one to four characters; if omitted, the file will print without modification.

CHARS name
: specifies the name of the character arrangement table to be used when printing the file; the name must be one to four characters.

FCB name
: controls the vertical spacing of the output on a page; the name must be one to four characters.

FORM {form / OFF}
: specifies the form name for spool files when they are closed. The form must be one to eight characters. OFF indicates that the default is to be used.

SEP
: puts separator page prior to spool file.

NOSep
: does not put a separator page prior to spool file. This is the default.

DIST {distcode / OFF}
: specifies the distribution code the spool file is to have when it is closed. OFF indicates that the default is to be used.

OFF
: performs the same function as 'CLASS *'.

CLOSE
: closes the spool file regardless of the CONT setting. CLOSE does not affect the setting of any spool attributes. If CLOSE is specified, PURGE may not be specified.

PURGE
closes and purges the spool file regardless of the CONT setting. PURGE does not affect the setting of any spool attributes. If PURGE is specified, CLOSE may not be specified.

Usage Notes:

1. The destination commit point occurs at the time the spool file is opened. This means that when the first line of a spool file is received and the file is opened, the destination (local or remote) is fixed for that file. If a SPOOL command is issued altering the destination and a spool file is open, it will be closed prior to the destination change taking effect.

2. Output classes are not supported for the local printer because the local output priority and forms selection can be performed with the HOLD option.

3. Local printing will be performed using the 132 characters/line font unless the 'CHARS 80' option has been specified to indicate that the 80 character/line font is to be used.

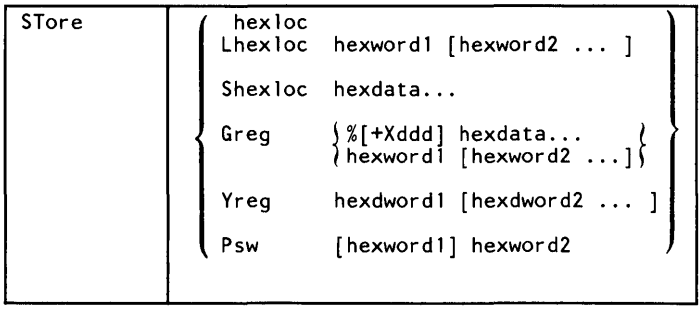
4. EXEC's running in CMS can stack the spool characteristics via "Query dvcadr" and parse out the LINECT operand. This allows lines/page to be controlled at a single point, the SPOOL command.

Response:

Same as CLOSE command if the CLOSE option is used to close a spool file (see "CLOSE" on page 7-14).

STORE

Use the STORE command to alter the contents of registers and storage of the virtual machine. The format of the STORE command is:



Where:

`hexloc`
`Lhexloc hexword1 [hexword2 ... ]`
stores the specified data in successive fullword locations starting at the fullword specified by hexloc. The smallest unit that can be stored is a fullword.

If hexloc is not on a fullword boundary, it is rounded down to the next lower fullword. If the value of an operand contains less than a fullword (8 hexadecimal digits), it is right-adjusted in the word and padded on the left (high-order digits) with zeros before it is stored.

`Shexloc hexdata...`
stores data at the address specified by hexloc without word alignment. The smallest unit that can be stored is one byte (2 hexadecimal digits). If an odd number of hexadecimal digits is specified, none of the data is stored, and an error message is generated.

`Greg %[+Xddd] hexdata...`
stores data at the address specified by in reg, plus the optional displacement ddd, without word alignment. The smallest unit that can be stored is one byte (2 hexadecimal digits). If an odd number of hexadecimal digits is specified, none of the data is stored, and an error message is generated.

`Greg hexword1 [hexword2 ...]`
stores the specified data in successive general registers starting at the register specified by reg. The reg operand must be a decimal value from 0 to 15 or

a hexadecimal digit from 0 to F If the value of an operand contains less than a fullword (8 hexadecimal digits), it is right-adjusted in the word and padded on the left (high-order digits) with zeros before it is stored.

`Yreg hexdword1 [hexdword2 ...]`
stores the specified data in successive floating-point registers starting at the register specified by reg. The reg operand must be an integer with one of the following values: 0, 2, 4, or 6. If the value of an operand contains less than a doubleword (16 hexadecimal digits), it is left-adjusted in the doubleword and padded on the right (low-order digits) with zeros before it is stored.

`PSW [hexword1] hexword2`
stores the specified data into the first and second words of the virtual machine PSW. If only hexword2 is specified, it is stored into the second word of the PSW. If either value, hexword1 or hexword2, is less than a fullword (8 hexadecimal digits), it is right-adjusted and padded on the left (high-order digits) with zeros before it is stored.

Usage Notes:

1. This command stores into only second-level storage (storage which is real to the virtual machine).

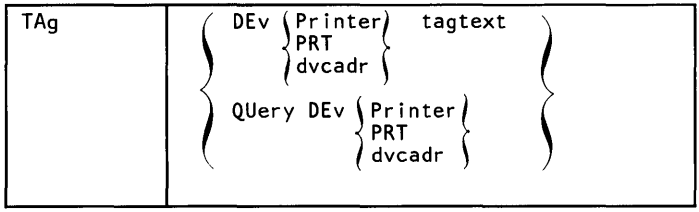
2. Multiple operands on a single command line will not be supported.

Response:

STORE COMPLETE

TAG

Use the TAG command to associate remote routing information with a spool file when it is spooled to the Remote Server. The format of the TAG command is:



Where:

DEV {PRINTER / PRT / dvcadr}
: specifies that the printer output is to have the tag information associated with it or that the printer tag information is to be queried.

tagtext
: specifies the tag information. It can be up to 134 characters including imbedded blanks. The contents of the tagtext depends on the requirements of the transmission network; it is user's responsibility that the addressing and control information is meaningful.

QUERY DEV {PRINTER / PRT / dvcadr}
: displays the tag information associated with the printer.

Usage Notes:

1. If the tagtext is not specified, the tag information area is cleared.

2. When the printer is spooled CLASS *, the tag information is ignored.

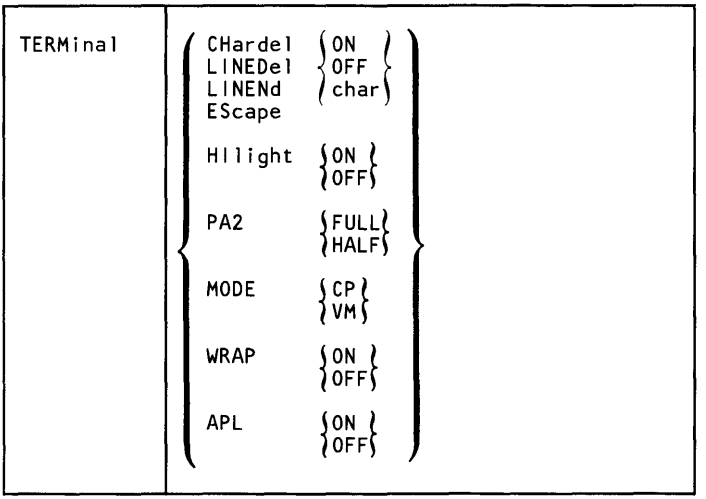
Responses:

PRT dvcadr TAG:
tagtext...
: The tagtext is displayed in response to a query for the printer and the tag has been set.

PRT dvcadr TAG NOT SET
: is displayed in response to a query for the printer and the tag has not been set.

TERMINAL

Use the TERMINAL command to control various functions of the keyboard/display. The format of the TERMINAL command is:



Where:

CHARDEL {ON | OFF | char}

defines the logical character delete symbol. If ON is specified, the default symbol (@) becomes the delete symbol. If OFF is specified, no logical character delete editing is done. If char is specified, that character becomes the logical character delete symbol; the character chosen should not be common to the normal data stream being entered. When you log on, CHARDEL ON is in effect.

LINEDEL {ON | OFF | char}

defines the logical line delete symbol. If ON is specified, the default symbol (^) becomes the delete symbol. If OFF is specified, no logical line delete editing is done. If char is specified, that character becomes the logical line delete symbol; the character

chosen should not be common to the normal data stream being entered. When you log on, LINEDEL ON is in effect.

LINEND { ON / OFF / char }

defines the logical line end symbol. If ON is specified, the default symbol (#) becomes the logical line end symbol. If OFF is specified, no logical line end editing is done. If char is specified, that character becomes the logical line end symbol; the character chosen should not be common to the normal data stream being entered. When you log on, LINEND ON is in effect.

ESCAPE { ON / OFF / char }

defines the logical escape symbol. If ON is specified, the default symbol (") becomes the logical escape symbol. If OFF is specified, no logical escape editing is done. If char is specified, that character becomes the logical escape symbol; the character chosen should not be common to the normal data stream being entered. When you log on, ESCAPE ON is in effect.

HILIGHT { ON / OFF }

when HILIGHT is set ON, input from the keyboard is high-lighted on the display; when set OFF, input is displayed at normal intensity. When you log on, HILIGHT OFF is in effect.

PA2 { FULL / HALF }

defines the amount that the activity screen is scrolled when the PA2 key is pressed with the cursor in the input area. If FULL is specified, the entire screen is cleared when the PA2 key is pressed. If HALF is specified, and there is no information on the lower half of the activity screen, the normal action of clearing the screen is performed; if there is information on the lower half of the activity screen, the line in the middle of the output area is scrolled to the top of the output area on the screen.

For both the FULL and HALF options, when the PA2 key is pressed, if the cursor resides on one of the lines in the output area, that line is scrolled to the top of the output area.

In all cases, all lines removed from the screen are scrolled into the terminal activity history buffer.

When you log on, PA2 FULL is in effect.

MODE {VM / CP}

controls the terminal attention environment. CP specifies that all attentions are to be handled by VM/PC CP. VM specifies that the first attention is to be reflected to the virtual machine; following attentions, prior to the virtual machine performing a VM READ, will the handled by VM/PC CP. When you log on, MODE VM is in effect.

WRAP {ON / OFF}

controls the display of information lines longer than the width of the screen.

ON specifies that more than 1 screen line is to be used to display the entire information line (information will wrap from one screen line to the next). An attempt will be made to split the information at an appropriate break-point so that words will not span 2 screen lines. Specifying ON also disables the scrolling commands for windowing the screen left and right.

OFF specifies that an information line is to take only 1 screen line and information beyond the width of the screen must be displayed with the left and right windowing command in the scroll area. If an information line is longer than a screen line, an ellipsis (...) will be displayed on the right-hand portion of the screen to indicate that more information is there.

When you log on, WRAP ON is in effect.

APL $\left\{\begin{matrix}\text{ON}\\\text{OFF}\end{matrix}\right\}$

is provided for command compatibility only; the APL character set is not available.

ON enables the Diagnose 54 alteration of the PA2 key function from screen clear to external interrupt.

OFF restores normal PA2 key screen clear functions.

When you log on, APL OFF is in effect.

Usage Notes:

1. The terminal settings you specify are only in effect for the duration of the session.

2. Although you can define line-editing symbols, the LINEDIT operand of the SET command determines whether line-editing functions will be performed or not.

3. Only one operand at a time may be specified on the TERMINAL command.

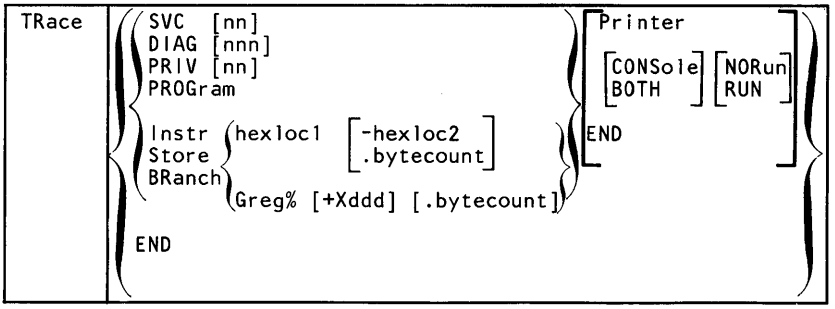
4. You cannot use any of the letters A-Z, or numbers 0-9 as the char in the CHARDEL, LINEDEL, LINEND, and ESCAPE operands.

TRACE

Use the TRACE command to trace various virtual machine activity on either the display, or printer, or both. Tracing can be continuous, RUN option, or step-by-step, NORUN option.

If you issue more than one TRACE command, the operands are cumulative. The RUN and NORUN operands, however, can be specified with different TRACE functions and do not cause conflict.

The format of the TRACE command is:



Where:

SVC [nn]
: traces virtual machine SVC interrupts. If nn is specified, the SVC whose hexadecimal interrupt code is nn is traced.

DIAG [nnn]
: traces virtual machine diagnose instructions. If nnn is specified, the diagnose instruction whose hexadecimal function code is nnn is traced.

PRIV [nn]
: traces virtual machine privileged instructions. If nn is specified, the privileged instruction whose hexadecimal operation code is nn is traced.

PROGRAM
: traces virtual machine program interrupts.

INSTR
: indicates that address comparison is be performed on storage accesses for instruction fetch.

STORE
: indicates that address comparison is be performed on accesses which cause virtual storage to be altered.

BRANCH
: traces virtual machine interrupts, PSW instructions, and successful branches.

hexloc1
: is the first or only hexadecimal storage location in the virtual machine where execution is to be halted.

-hexloc2
: is the last storage location defining a range of virtual addresses to be used for comparison. If an address compare occurs, including on hexloc2, execution is halted.

Greg% [+Xddd
: the first, or only, virtual address at which to halt execution is the address contained in reg, plus the optional displacement ddd.

.bytecount
: is a hexadecimal value of the number of bytes to be included within a range of virtual addresses to be used for comparison; the sum of hexloc1 plus bytecount determines the first address that is beyond the range. If an address compare occurs within the range of addresses execution is halted.

PRINTER
: directs tracing output to the printer. The destination of the output depends on the SPOOL and TAG values currently set.

CONSOLE
: directs tracing output to the display.

BOTH
: directs tracing output to both the printer and display.

NORUN
: halts program execution after the trace output is displayed and enters the VM/PC CP command environment.

RUN
: continues program execution after the trace output has been displayed.

END
: terminates the specified or all tracing activity.

Usage Notes:

1. A table of 16 specific codes for each of SVC, DIAG, and PRIV operands is kept. When the operand is specified without the nn option, the table is cleared and

all interrupts or instructions are traced for that operand. The codes specified in these operands are cumulative until the respective table is full; an entry may not be deleted from the table, but the table's contents can be cleared by re-specifying the operand without the nn option.

2. To resume operation of the virtual machine after instruction execution has been halted with the NORUN option, the BEGIN command must be entered.

3. The TRACE command can be used as an address stop function by using the NORUN option.

4. Stores to a location which do not cause the storage to change will not be detected by storage alteration trace.

5. Storage alteration trace is limited to any four (4) consecutive bytes of storage.

6. If trace output is being recorded on the printer, a CLOSE command must be entered to create the spool file.

7. When recording trace output on the printer, the trace output is intermixed with other data sent to the printer.

Responses: The following symbols are used in the responses received from TRACE:

mnem	mnemonic for the instruction
iiiiiiii	instruction (2, 4, or 6 bytes)
bm	branch mask
cc	condition code
code	interrupt code

TRACE ACTIVATED
tracing function activated

TRACE TERMINATED
tracing function terminated

```
INSTR FETCH AT xxxxxx  mnem  iiiiiiii  CC=cc
```
The instruction at virtual storage location xxxxxx satisfied the instruction fetch comparison and sequential processing continues.

```
INSTR FETCH AT xxxxxx  mnem  iiiiiiii  ===> yyyyyy
```
The instruction at virtual storage location xxxxxx satisfied the instruction fetch comparison, and the next instruction will be fetched from storage location yyyyyy.

```
STG ALTER AT xxxxxx BY INSTR AT yyyyyy  mnem
iiiiiiii  CC=cc
```
The instruction at virtuual storage location yyyyyy satisfied the storage alteration comparison at xxxxxx.

```
PROG INTERRUPT code AT xxxxxx  mnem  iiiiiiii ===>
yyyyyy
```
A program interrupt occurred at location xxxxxx and the resulting PSW swap will fetch the instruction at yyyyyy.

```
SVC INTERRUPT code AT xxxxxx  menm  iiiiiiii ===>
yyyyyy
```
A supervisor call interrupt occurred at location xxxxxx and the resulting PSW swap will fetch the instruction at yyyyyy.

CMS Commands

The CMS command language allows you to create, modify, debug, and, in general, manipulate a collection of files.

Character Set Usage

You can enter CMS commands using a combination of characters from six different character sets. The following figure shows the contents of the character sets.

Character Set	Names	Symbols
Separator	Blank	
National	Dollar Sign Pound Sign At Sign	$ # @
Alphabetic	Uppercase Lowercase	A - Z a - z
Numeric	Numeric	0 - 9
Alphameric	National Alphabetic Numeric	$,#,@ A - Z a - z 0 - 9
Special		All other characters

Figure 7-2. Character Sets and Their Contents

CMS Command Search Order

When you enter a command line in the CMS environment, CMS has to locate the command to execute. If you have EXEC or MODULE files on any of your accessed disks, CMS treats them as commands; also, they are known as user-written commands.

As soon as the command name is found, the search stops and the command is executed. The search order is:

1. Search for a file with filetype EXEC on any currently accessed disk. CMS uses the standard search order (A through Z).

2. Search for a valid name on any currently accessed disk, according to current SYNONYM file definitions in effect.

3. Search for a nucleus extension command if the high order byte of register 1 is not equal to X'03' or X'04'.

4. Search for a command in the transient area. Commands which may be in the transient area are:

ACCESS	LISTFILE	RENAME
COMPARE	MODMAP	SET
FILEDEF	PRINT	SYNONYM
GENDIRT	QUERY	TYPE
GLOBAL	RELEASE	

5. Search for a nucleus-resident command. The nucleus-resident CMS commands are:

CP	GENMOD	START
DEBUG	INCLUDE	STATE
ERASE	LOAD	STATEW
ESTATE	LOADMOD	
ESTATEW	NUCXLOAD	

6. Search for a file with filetype MODULE on any currently accessed disk.

7. Search for a valid abbreviation or truncation of a nucleus extension.

8. Search for a valid abbreviation or truncation of a command in the transient area.

9. Search for a valid abbreviation or truncation of a command in the nucleus.

10. Search for a valid abbreviation or truncation of any other CMS command

11. Search for a CP command.

12. Search for a valid abbreviation or truncation of a CP command.

For example, if you create a command module that has the same name as a CMS nucleus-resident command, your command module cannot be executed, since CMS locates the nucleus-resident command first, and executes it. When a user-written command has the same name as a CMS command module abbreviation, certain error messages may indicate the CMS command name, rather than the program name.

You can enter CMS commands when you are running CMS in your virtual machine, the terminal is idle, and the virtual machine can accept input.

When CMS is processing a previously entered command, the keyboard remains unlocked for additional command input. Note that in these circumstances the command you enter is stacked in the terminal input buffer and is not executed until the command that is currently being executed completes. If more commands are entered than CP can handle, a NOT ACCEPTED message is displayed at the display terminal.

In addition to the commands listed in the figures that follow, there are three commands called Immediate commands that are handled in a different manner from the others. They may be entered while another command is being executed, and they are executed immediately. The Immediate commands are:

- HT - Halt typing
- HX - Halt execution
- RT - Resume typing

Summary of CMS Commands

Command	Usage
ACCESS	Identify direct access space to a CMS virtual machine, create extensions and relate the disk space to a logical directory.
CMDCALL	Converts EXEC 2 extended plist function calls to CMS extended plist command calls.
COMPARE	Compare records in CMS disk files.
COPYFILE	Copy CMS disk files according to specifications.
CP	Enter CP commands from the CMS environment.
DEBUG	Enter DEBUG subcommand environment.
DEFAULTS	Set or display default options for the command: FILELIST
DROPBUF	Eliminate a program stack buffer.
ERASE	Delete CMS disk files.
ESTATE	Verify the existence of a CMS disk file.
ESTATEW	Verify the existence of a CMS disk file on a read/write CMS disk.
EXEC	Execute special procedures made up of frequently used sequences of commands.
EXECIO	Do I/O operations between a device and the program stack.
EXPORT	Make a copy of a CMS file as a Personal Computer file.
FILEDEF	Define an OS ddname and relate that ddname to any device supported by CMS.
FILELIST	List information about CMS disk files, with the ability to edit and issue commands from the list.
FINIS	Close an open file.

Figure 7-3 (Part 1 of 3). Summary of CMS Commands

Command	Usage
FORMAT	Prepare disks in CMS fixed block format.
GENDIRT	Fill in auxiliary module directories.
GENMOD	Generate executable CMS files (MODULE files).
GLOBAL	Identify specific CMS libraries to be searched for macros, copy files, or missing subroutines.
GLOBALV	Set, maintain, and retrieve a collection of named variables.
IDENTIFY	Display or stack userid, date, time, time zone, and day of the week.
IMPORT	Make a copy of a Personal Computer file on a CMS minidisk.
INCLUDE	Bring additional TEXT files into storage and establish linkages.
LISTFILE	List information about CMS disk files.
LOAD	Bring TEXT files into storage for execution.
LOADMOD	Bring a single MODULE file into storage.
MACLIB	Create or modify CMS macro libraries.
MAKEBUF	Create a new program stack buffer.
MODMAP	Display the load map of a MODULE file.
NUCXDROP	Delete specified nucleus extensions.
NUCXLOAD	Load a nucleus extension.
NUCXMAP	Identify existing nucleus extensions.
PRINT	Spool a specified CMS file to the system printer.
QUERY	Request information about a CMS virtual machine.
RELEASE	Make a disk and its directory inaccessible to a CMS virtual machine.
RENAME	Change the name of a CMS file or files.

Figure 7-3 (Part 2 of 3). Summary of CMS Commands

Command	Usage
SENTRIES	Determine the number of lines currently in the program stack.
SET	Establish, set, or reset CMS virtual machine characteristics.
SORT	Arranges a specified file in ascending EBCDIC order according to sort fields.
START	Begin execution of programs previously loaded.
STATE	Verify the existence of a CMS disk file.
STATEW	Verify the existence of a CMS disk file on a read/write CMS disk.
SYNONYM	Invoke a table containing synonyms you have created for CMS and user-written commands.
TXTLIB	Generate and modify text libraries.
TYPE	Display all or part of a CMS file at the terminal.
UPDATE	Make changes in a program source file as defined by control cards in a control file.
XEDIT	Invoke the VM/PC Editor to create or modify a disk file.

Figure 7-3 (Part 3 of 3). Summary of CMS Commands

CMS Commands and Options

This section contains reference information for the CMS commands used by general users. Each command description indicates the command format, operands and options; it also lists error messages and return codes the command issues. Usage notes are provided, where applicable.

The formats of the DEBUG, XEDIT, and EXEC commands are also listed; for details on the DEBUG subcommands or EXEC control statements, see *VM/SP CMS Command and Macro Reference.*

For details on the XEDIT subcommands and macros, see *VM/SP System Product Editor Command and Macro Reference.*

For usage information on XEDIT subcommands and macros, see *VM/SP System Product Editor User's Guide.*

For more detailed usage information on CMS commands, see the *VM/SP CMS User's Guide.*

ACCESS

Use the ACCESS command to identify a disk to CMS, establish a filemode letter for the files on the disk, and set up a file directory in storage. The specifications you make with the ACCESS command determine the entries in the user file directory. The format of the ACCESS command is:

ACcess	[cuu mode[/ext [fn [ft [fm]]]] [(options...[)]]] [101 A *] [* *] options: ERASE

Where:

cuu
: makes available the disk at the specified virtual device address. The default value is 101.

  Valid addresses are 001 through FFF.

mode
: assigns a one-character filemode letter to all files on the disk being accessed. This field must be specified if cuu is specified. The default value is A.

ext
: indicates the mode of the parent disk. Files on the disk being accessed (cuu) are logically associated with files on the parent disk; the disk at cuu is considered a read-only extension. A blank must not precede or follow the slash (/).

`fn [ft [fm]]`
defines a subset of the files on the specified disk. Only the specified files are included in the user file directory and only those files can be read. An asterisk coded in any of these fields indicates all filenames, filetypes, or filemode numbers (except 0) are to be included. (See Usage Notes 3 and 4.) To specify a filemode use a letter and a number, for example: B1.

Options:

`ERASE`
specifies that you want to erase all of the files on the specified disk. This option is only valid for read/write disks.

Usage Notes:

1. If you have defined disk addresses 100, 101, and 10E in the VM/PC configuration file, or if they are defined before you IPL CMS, these disks are accessed as the S-, A-, and Y-disks respectively. Following an IPL of CMS, you must issue explicit ACCESS commands to access other disks. Ordinarily, you have access only to files with a filemode number of 2 on the system disk.

2. Associated with each CMS disk is a file directory, which contains an entry for every CMS file on the disk. The user file directory created in storage by the ACCESS command contains entries for only those files that you can reference.

   If you use the CP LINK command to link to a new minidisk, issue an ACCESS command each time. Do this so that you obtain the appropriate file directory.

3. The filename, filetype, and filemode fields can only be specified for disks that are accessed as read-only extensions. For example:

   ```
   access 195 b/a * assemble
   ```

gives you read-only access to all the files with a filetype of ASSEMBLE on the disk at virtual address 195. The command:

```
access 100 z/a * * z1
```

gives you access to all files on the system disk (100) that have a filemode number of 1.

When you access any disk in read-only status, files with a filemode number of 0 are not accessed.

4. You can also identify a set of files on a disk by referring to a filename or filetype prefix. For example:

   ```
   access 192 c/a abc*
   ```

   accesses only those files in the disk at virtual address 192 whose filenames begin with the characters ABC. The command line:

   ```
   access 192 c/a * a* c2
   ```

   gives you access to all files whose filetypes begin with an A and that have a filemode number of 2.

5. You can force a read/write disk into read-only status by accessing it as an extension of another disk or of itself; for example:

   ```
   access 101 a/a
   ```

   forces your A-disk into read-only status.

6. When a disk is made a read-only extension of another disk, commands that typically require or allow you to specify a filemode may search extensions of the specified disk. The exception to this is the LISTFILE command. For a detailed description of read-only extensions, see the *VM/SP CMS User's Guide.*

7. If you enter the ERASE option by mistake, you can recover from the error as long as you have not yet written any new files onto the disk. (That is, you have

not yet caused CMS to rewrite the file directory.) Reissue the ACCESS command without the ERASE option.

8. If two or more disks have been accessed in CMS, and CP DEFINE commands are executed that swap virtual addresses, then a subsequent RELEASE command may write the file directory on the wrong disk; for example:

```
(CMS)  ACCESS 193 C
(CMS)  ACCESS 198 E
(CP)   DEFINE 193 293
(CP)   DEFINE 198 193
(CMS)  RELEASE C
```

   This sequence of commands will write the file directory from 193 to 198 since the CP definitions are unknown to CMS.

9. To free an accessed disk, refer to the CMS RELEASE Command.

Responses:

`DMSACC723I mode (cuu) read/only`
: This message is displayed if the disk is accessed read-only.

`DMSACC724I cuu1 REPLACES mode(cuu2)`
: Before execution of the command, the disk represented by cuu2 was the "mode" disk. The disk, cuu1, is now assigned that filemode letter. This message is followed by message DMSACC726I.

`DMSACC725I cuu ALSO = 'mode' DISK`
: The disk specified by cuu is the mode disk and an ACCESS command was issued to assign it another filemode letter.

`DMSACC726I 'cuu mode' RELEASED`
: The disk being accessed at virtual address cuu as a read/write disk is already accessed at a different mode. It is released from that mode. Or, a disk currently accessed at mode is being replaced.

Other Messages and Return Codes:

```
DMSACC003E INVALID OPTION 'option'  RC=24
DMSACC017E INVALID DEVICE ADDRESS 'cuu'  RC=24
DMSACC048E INVALID MODE 'mode'  RC=24
DMSACC059E 'cuu' ALREADY ACCESSED AS READ/WRITE 'mode' DISK  RC=36
DMSACC060E FILE(S) 'fn [ft [fm]]' NOT FOUND. DISK 'mode(cuu)' WILL
           NOT BE ACCESSED  RC=28
DMSACC070E INVALID PARAMETER 'parameter'  RC=24
DMSACC109S VIRTUAL STORAGE CAPACITY EXCEEDED  RC=104
DMSACC112S DISK 'mode(cuu)' has invalid directory format  RC=100
DMSACC113S mode (cuu) NOT ATTACHED  RC=100
```

CMDCALL

Use the CMDCALL command in an EXEC 2 procedure to allow certain CMS commands (ERASE, LISTFILE, RENAME, and STATE) to display the message 'FILE NOT FOUND'. CMDCALL accomplishes this by converting the EXEC 2 extended plist function calls to CMS extended plist command calls. The format of the CMDCALL command is:

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

For a complete description of the CMDCALL command, refer to the *VM/SP CMS Command and Macro Reference.*

COMPARE

Use the COMPARE command to compare two CMS disk files of of fixed- or variable-length format on a record-for-record basis and to display dissimilar records at the terminal. The format of the COMPARE command is:

COMpare	fileid1 fileid2 [(option...[)]] Option: COL[mm[-]nn] [1 lrecl]

For a complete description of the COMPARE command, refer to the *VM/SP CMS Command and Macro Reference.*

CONWAIT

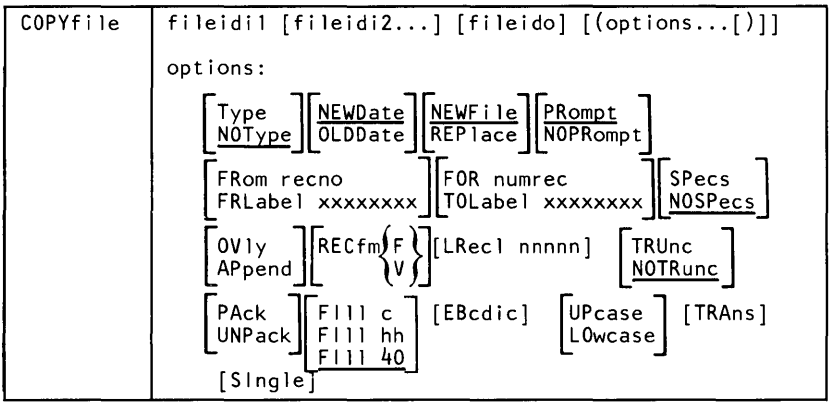
Use the CONWAIT command to cause a program to wait until all pending terminal I/O is complete. The format of the CONWAIT command is:

CONWAIT	

Usage Note: The CONWAIT command synchronizes input and output to the terminal; it ensures that the output console stack is cleared before the program continues execution. Also, you can ensure that a read or write operation is finished before you modify an I/O buffer.

COPYFILE

Use the COPYFILE command to copy and/or modify CMS disk files. The manner in which the file identifiers are entered determines whether or not one or more output files are created. The format of the COPYFILE command is:



For a complete description of the COPYFILE command, refer to the *VM/SP CMS Command and Macro Reference.*

CP

Use the CP command to transmit commands to the VM/SP control program environment without leaving the CMS environment. The format of the CP command is:

CP	commandline

Where:

`commandline`
is any CP command valid for your CP command

privilege class. If this field is omitted, you are placed in the CP environment and may enter CP commands without preceding each command with CP. To return to CMS, issue the CP command BEGIN.

Usage Notes:

1. You must use the CP command to invoke a CP command:
   - Within an EXEC procedure
   - If the implied CP (IMPCP) function is set to OFF for your virtual machine
2. To enter a CP command from the CMS environment without CMS processing the command line, use #CP.
3. When you enter an invalid CP command following the CP command, you receive a return code of -1. In an EXEC, this return code is +1.

Responses: All responses are from the CP command that was issued; the CMS ready message follows the response.

DEBUG

Use the DEBUG command to enter the debug environment from the CMS environment. In the debug environment you can use a variety of DEBUG subcommands that allow you to test and debug your programs. For tutorial information, including examples, see the *VM/SP CMS User's Guide.* The format of the DEBUG command is:

DEBUG	

Usage Notes:

1. The debug environment is also entered as a result of an external interruption or the result of a breakpoint (address stop) encountered during program execution.

2. Once you are in the debug environment, you can enter only DEBUG subcommands and CP commands via the #CP function.

3. To return to the CMS environment, enter the DEBUG subcommand RETURN.

Responses:

`DMSDBG728I DEBUG ENTERED`
This message indicates that you are in the debug environment.

DEFAULTS

Use the DEFAULTS EXEC procedure to set up default options for the FILELIST command. Each time you invoke FILELIST, the options specified in the DEFAULTS command are in effect. However, the options specified with each invocation of FILELIST override the ones set up in the DEFAULTS command. Thus, you can customize the options by using DEFAULTS, yet override them when you desire. DEFAULTS can also be used to display the current default options for the FILELIST command.

The format of the DEFAULTS command is:

DEFAULTS	[Set FILELIST options... List]

Where:

`Set`
specifies that default options are to be set up for the FILELIST command.

`List`
specifies that the current default options for the FILELIST command are to be displayed.

`options`
is one or more options associated with the FILELIST command, as shown below.

The FILELIST command and the options that can be specified as defaults are listed below. Valid abbreviations

for both the command name and the keyword options are indicated by uppercase letters. Mutually exclusive options are listed one under the other.

Command Name	Options	
Filelist	Profile fn	Filelist
	Profile PROFFLST	NOFilelist

Usage Notes: The DEFAULTS command uses the GLOBALV command, which maintains a LASTING GLOBALV file on your A-disk. This file contains the options specified in a DEFAULTS command. However, **do not edit the LASTING GLOBALV file** to change the options. Use the DEFAULTS command, instead. For more information on GLOBALV files, see the description of the GLOBALV command.

Responses: The following is a list of your default options for the FILELIST command:

```
option...
```

To change these default options enter:

```
'DEFAULTS Set FILELIST opt1 [opt2..]'.
```

The following default options have been set:

```
FILELIST option...
```

To change any default options enter:

```
DEFAULTS Set FILELIST opt1 [opt2..]
```

Error Messages and Return Codes:

```
DMSWDF014E INVALID KEYWORD 'function'.  RC=24
DMSWDF029E INVALID PARAMETER 'parm' IN THE OPTION 'option' FIELD.
           RC=24
DMSWDF637E MISSING VALUE FOR THE PROFILE OPTION. RC=24.
DMSWDF641E NO {options|command} SPECIFIED.  RC=24
DMSWDF642E DEFAULTS DOES NOT ACCEPT {'COMMAND' command|'OPTION'
           OPTION FOR 'COMMAND'}. RC=24
DMSWDF653E ERROR EXECUTING 'GLOBALV', RC = nn.  RC=40
DMSSTT062E INVALID CHARACTER 'character' IN FILEID 'fn ft fm'.  RC=20
```

DESBUF

Use the DESBUF command to clear the console and program stack input and output buffers. The format of the DESBUF command is:

DESBUF	

Usage Notes: Note that DESBUF clears the output buffers as well as the input buffers. Use the CONWAIT command before DESBUF to halt program execution until all output lines are displayed at the terminal.

Warning: Be careful when using the DESBUF command because the input and output console and program stack buffers are used to communicate information between programs.

DROPBUF

Use the DROPBUF command to eliminate the most recently created program stack buffer. The format of the DROPBUF command is:

DROPBUF	n

Where:

n
: indicates the number of the first program stack buffer you want to drop. CMS drops the indicated buffer and all buffers created after it. If n is not specified, only the most recently created buffer is dropped.

Usage Note: Note that you can specify a number with DROPBUF. For example, if you issue:

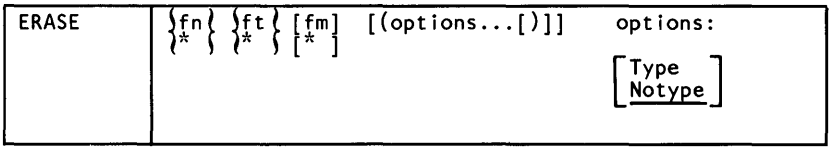
```
DROPBUF 4
```

CMS eliminates program stack buffer 4 and all program stack buffers created after it. Thus, if there were presently six program stack buffers, CMS would eliminate program

stack buffers 6, 5, and 4. If you issued DROPBUF without specifying n, only program stack buffer 6 would be eliminated.

ERASE

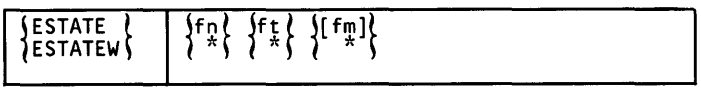
Use the ERASE command to delete one or more CMS files from a read/write disk. The format of the ERASE command is:



For a complete description of the ERASE command, refer to the *VM/SP CMS Command and Macro Reference.*

ESTATE/ESTATEW

Use the ESTATE command to verify the existence of a CMS file on any accessed disk; use the ESTATEW command to verify the existence of a CMS file on any accessed read/write disk. In most instances, the commands ESTATE and ESTATEW are equivalent to STATE and STATEW. However, if the file being verified has more than 65535 items or more than 65535 data blocks, then the ESTATE/ESTATEW commands should be used. Use of the STATE/STATEW commands will result in error message DMSSTT253E FILE 'fn ft fm' CAN NOT BE HANDLED WITH SUPPLIED PLIST. The use of ESTATE/ESTATEW will not produce this message, and is therefore preferred if the existence of such a file may be verified. The formats of the ESTATE and ESTATEW commands are:



Where:

`fn`
: is the filename of the file whose existence is to be verified. If fn is specified as *, the first file found satisfying the rest of the fileid is used.

ft
: is the filetype of the file whose existence is to be verified. If ft is specified as *, the first file found satisfying the rest of the fileid is used.

fm
: is the filemode of the file whose existence is to be verified. If fm is omitted, or specified as *, all your disks are searched.

Usage Notes:

1. If you issue the ESTATEW command specifying a file that exists on a read-only disk, you receive error message DMSSTT002E.

2. When you code an asterisk in the fn or ft fields, the search for the file is ended as soon as any file satisfies any of the other conditions. For example, the command:

   ```
   estate * file
   ```

   executes successfully if any file on any accessed disk (including the system disk) has a filetype of FILE.

3. You can invoke the ESTATE/ESTATEW command from the terminal, from an EXEC file, or as a function from a program. If ESTATE/ESTATEW is invoked as a function or from an EXEC file that has the message output suppressed, the message DMSSTT002E FILE 'fn ft fm' NOT FOUND is not issued.

Response: The CMS ready message indicates that the specified file exists.

Error Messages and Return Codes:

```
DMSSTT002E FILE 'fn ft fm' NOT FOUND  RC=28
DMSSTT048E INVALID MODE 'mode'  RC=24
DMSSTT054E INCOMPLETE FILEID SPECIFIED  RC=24
DMSSTT062E INVALID CHARACTER 'char' IN FILEID 'fn ft'  RC=20
DMSSTT069E DISK 'mode' NOT ACCESSED  RC=36
DMSSTT070E INVALID PARAMETER 'parameter'  RC=24
```

EXEC

Use the EXEC command to execute one or more CMS commands or EXEC control statements contained in a specified CMS EXEC or EXEC2 file. The format of the EXEC command is:

[EXec]	fn [args...]

For a complete description of the EXEC command, refer to the *VM/SP CMS Command and Macro Reference.*

EXECIO

Use the EXECIO command to:

- Write lines from the program stack to a CMS disk file or to a virtual printer.
- Cause execution of CP commands and recover resulting output.

In some cases output data to be written may be supplied directly on the EXECIO command line.

The format of the EXECIO command is:

EXECIO	(syntax below)

```
                                                    OPTIONS-SEE BELOW
EXECIO   {lines} / DISKR fn ft [fm[linenum]] [([FINIs] [a][b]]          \
         {  *  } |                                                       |
                 | CP     [(                         [a][b][d][)]]       |
                 | DISKW fn ft fm [linenum                               |
                <        [recfm [lrecl]]]  [([FINIs]  [b][c][d][)]]      >
                 |                                                       |
                 | PRINT [([CC  {code} ]             [b][c][d][)]]       |
                 |       [      {DATA} ]                                 |
                 \ EMSG  [(                          [b][c][d][)]]       /

Option formats:

(a)                  (b)                  (c)            (d)

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

Zone {n1 n2}         NOTYPE
     {1  * }

LIFO
FIFO

SKip
```

Note: Parsing of the EXECIO command differs from that of other CMS commands in that it involves handling of strings that may contain embedded blanks, parenthesis, other special characters, and words of more than eight characters. Therefore, if a right 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.

Since VM/PC does not support virtual readers or punches, these devices are not valid in the EXECIO command. Except for that, the VM/PC EXECIO command is functionally equivalent to the VM/SP command.

When EXECIO is first called, it is loaded into CMS free storage and installed as a nucleus extension. Further calls to EXECIO normally use this copy of EXECIO, thus eliminating the need to reload the EXECIO module.

For a complete description of the EXECIO command, refer to the *VM/SP CMS Command and Macro Reference.*

EXPORT

Use the EXPORT command to make a copy of a CMS file as a Personal Computer file. The format of the EXPORT command is:

EXPort	fn ft fm d:filename.ext [(options...[)]] where options may be: [REP or REPLACE] [ASCII or NOASCII] [EOL or NOEOL]

Where:

`fn`
is the filename of the CMS file to export.

`ft`
is the filetype of the CMS file to export.

`fm`
is the filemode of the CMS file to export.

Note: You must specify a fn, ft, and fm when using the EXPORT command.

`d:filename.ext`
is the filespec of the Personal Computer file to be created. The drive and filename must be specified. The filename extension is optional. No special characters (?,*) are allowed in the filespec.

`REP or REPLACE`
Specifies that an existing Personal Computer file with the fileid specified will be replaced with the specified CMS file.

`ASCII or NOASCII`
ASCII specifies that the file is to be translated from EBCDIC to ASCII during the EXPORT. NOASCII specifies that the file is not to be translated from EBCDIC to ASCII during the EXPORT. ASCII is the default.

EOL or NOEOL

EOL specifies that end-of-line characters (carriage return/line feed) are to be inserted into the target file to delimit the lines of the file. NOEOL specifies that the target file is to be created without inserting end-of-line characters to delimit lines of the file. EOL is the default.

Notes:

1. The EXPORT command may create a temporary file called "EXPORT.CMS" during the export of the CMS file. Any file which you have on the target disk called "EXPORT.CMS" will be erased if the EXPORT command is used.

2. The CMS file to be exported must have a logical record length of 65535 or less if the EOL option is to be used. If the LRECL of the file exceeds this, it cannot be exported with the EOL option.

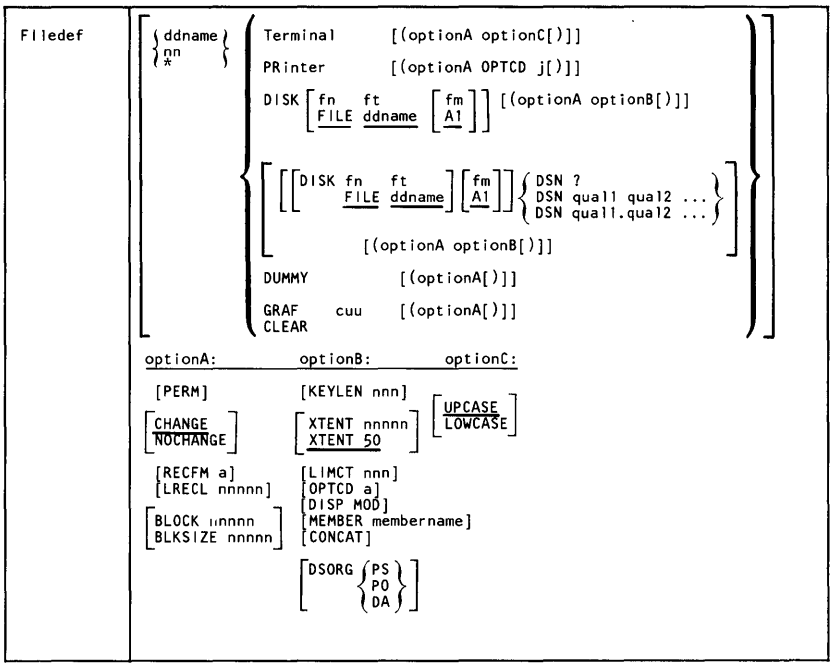
3. A file which has been exported, translated to ASCII, and subsequently imported may differ from the original file, due to the translate tables used in the file transformations. The NOASCII options should be used to avoid problems caused by this transformation. Appendix E, "ASCII to EBCDIC Conversion" on page E-1 lists the EBCDIC-ASCII conversion tables which are used in this translation.

Messages and Return Codes:

```
DMSEXP002E File 'fn ft fm not found   RC=28
DMSEXP003E Invalid option 'option'   RC=24
DMSEXP024E File 'd:filename.ext' already exists --
             specify 'replace' RC=28
DMSEXP042E No fileid(s) specified   RC=24
DMSEXP048E Invalid mode 'mode'   RC=24
DMSEXP054E Incomplete fileid specified   RC=24
DMSEXP062E Invalid * in fileid   RC=20
DMSEXP070E Invalid parameter 'parameter'   RC=24
DMSEXP104E Error 'nn' reading file 'fn ft fm' from disk   RC=100
DMSEXP622E Insufficient storage available   RC=41
DMSEXP800E Invalid export of a CMS file to a system file   RC=100
DMSEXP804E Invalid PC fileid 'd:filename.ext' specified  RC=20
DMSEXP806E Export terminated, insufficient disk
             space available  RC=32
DMSEXP807S Drive d not ready. Make drive ready and press enter
DMSEXP808S File cannot be exported with the 'EOL' option  RC=20
DMSEXP907E I/O error on file 'd:filename.ext'    RC=256
DMSSTT048E Invalid mode 'mode'   RC=24
DMSSTT062E Invalid character 'char' in fileid 'fn ft fm'   RC=20
DMSSTT069E Disk 'mode' not accessed   RC=36
```

FILEDEF

Use the FILEDEF command to establish data definitions for OS ddnames, or to override default file definitions made by the assembler and the OS language processors. The format of the FILEDEF command is:



Since VM/PC does not support virtual readers, punches, or tapes, these devices are not valid in the FILEDEF command. Except for this, the VM/PC FILEDEF command is functionally equivalent to the VM/SP command.

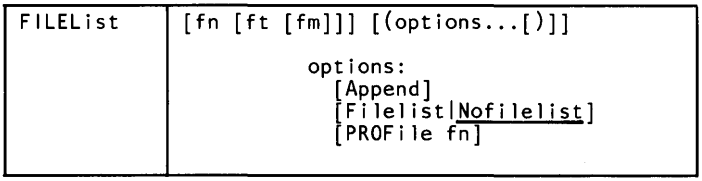
For a complete description of the FILEDEF command, refer to the *VM/SP CMS Command and Macro Reference.*

FILELIST

Use the FILELIST EXEC procedure to display a list of information about CMS files residing on accessed disks. In the FILELIST environment, information that is normally

provided by the LISTFILE command (with the DATE option) is displayed under the control of the System Product editor. You can use XEDIT subcommands to manipulate the list itself. You can also issue CMS commands against the files directly from the displayed list.

The format of the FILELIST command is:



The VM/PC command FILELIST is functionally equivalent to the VM/SP command FILELIST, except that PF 1 will perform a sort of the list of files based on the filenames in the list.

For a complete description of the FILELIST command, refer to the *VM/SP CMS Command and Macro Reference.*

FINIS

Use the FINIS command to close one or more files. The format of the FINIS command is:

Where:

`fn`
is the filename of the file to be closed. If you code an asterisk (*) in this field, all filenames are closed.

`ft`
is the filetype of the file to be closed. If you code an asterisk (*) in this field, all filetypes are closed.

`fm`
is the filemode of the file to be closed. If you code an asterisk (*) in this field, all disks are searched for the specified file. If this field is omitted, A1 is assumed.

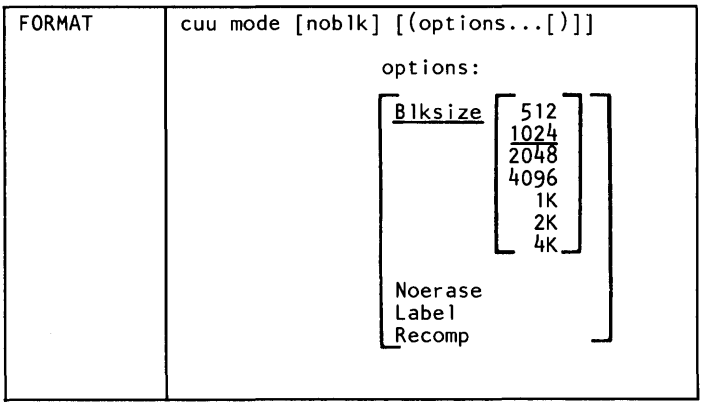
Usage Note: Use FINIS when your program does not close a file during its execution. CMS commands close files automatically at the end of their execution. (An 'EXEC' file is considered to be a single CMS command, independent of its content.)

FORMAT

Use the FORMAT command to:

- Initialize a virtual disk (minidisk) for use with CMS files
- Count or reset the number of blocks on a virtual disk
- Write a label on a virtual disk

The format of the FORMAT command is:



Where:

cuu
: is the virtual device address of the virtual disk to be formatted.

  Valid addresses are 001 through FFF.

mode
: is the filemode letter to be assigned to the specified device address. Valid filemode letters are A through Z. This field must be specified. If any other disk is accessed as mode, it is released.

noblk
is the number of FB-512 blocks to be made available for use. If the number specified exceeds the actual number of blocks on the disk, then all the blocks on the disk are made available for use.

Options:

BLKSIZE
specifies the physical DASD block size of the CMS minidisk. The block sizes 1024, 2048, and 4096 may alternately be specified as 1K, 2K, and 4K, respectively.

NOERASE
specifies that the permanently formatted FB-512 blocks are not to be cleared to zeros. If not specified, the FB-512 blocks will be cleared.

LABEL
writes a label on the disk without formatting the disk. The CMS disk label is written on block1 of an FB-512 device. A prompting message requests a six-character disk label (fewer than six characters are left-justified and blanks padded).

RECOMP
changes the number of FB-512 blocks on the disk that are available to the user. This number becomes the actual number of minidisk FB-512 blocks, or the number specified by noblk, whichever is less. If noblk is not specified, the maximum number of blocks initially formatted on the disk is made available to the user.

Usage Notes:

1. When you do not specify either the RECOMP or LABEL option, the disk area is initialized by writing records containing binary zeros on each block. Any previous data on the disk is erased. For example:

```
format 101 a  1000
```

initializes 1000 blocks of the disk located at virtual address 101 in CMS format. The command:

```
format 102 b  800 (recomp)
```

changes the number of blocks available at virtual address 102 to 800, but does not erase any existing CMS files. To change only the label on a disk, you can enter:

```
format 103 c (label)
```

Respond to the prompting message with a six-character label.

2. When formatting FB-512 devices, enough blocks of the minidisk area must be formatted to support the CMS disk structure, or message DMS216E will be displayed, and the FORMAT request will be terminated. The number of FB-512 blocks which must be formatted for minidisks of 512, 1K, 2K, and 4K CMS blocksize is 6, 12, 24, and 48, respectively.

3. If the FORMAT command with the RECOMP option fails and CMS issues message DMSFOR214W, "CANNOT RECOMPUTE WITHOUT LOSS OF DATA. NO CHANGE.," query your A-disk to determine the number of unallocated blocks. If the number of blocks seems adequate, it is possible that some of the allocated space is at the end of the disk, and is thus not available to the FORMAT command. Issue the command:

   ```
   COPY * * A = = = (REP
   ```

   followed by the FORMAT command with the RECOMP option.

4. Choosing an appropriate BLKSIZE to format a disk depends upon its intended use. A 4K BLKSIZE will optimize the I/O if the disk is to contain large files with no missing records (dense). A BLKSIZE of 1K is more appropriate when creating many small files or sparse files. For example, PL/I regional files are sparse and they may allocate more space on a 4K disk than on a 1K disk, therefore, the smaller BLKSIZE is preferable.

   The larger the blocksize of the disk, the greater the amount of storage required for input/output buffers.

Each buffer that the system needs must be a contiguous block of system storage. The size of this area of storage being the blocksize of the disk. Programs that dynamically allocate storage based upon machine size may use up all of the available storage. This may not allow the system enough storage to allocate buffers for its use. Consequently, a program needing a 4K disk that uses all of the available storage may be unable to get I/O buffers if they are not already allocated. For more information on CMS storage management, refer to the *VM/SP System Logic and Problem Determination Guide Vol. 2 - CMS.*

Responses:

```
DMSFOR603RFORMAT WILL ERASE ALL FILES ON DISK
           'mode(cuu)'.
           DO YOU WISH TO CONTINUE? (YES|NO):
```

You have indicated that a disk area is to be initialized: all existing files are erased. This message gives you the option of canceling the execution of the FORMAT command. Reply yes or no.

```
DMSFOR605R ENTER DISK LABEL:
```

You have requested that a label be written on the disk. Enter a one- to six-character label.

```
DMSFOR705I DISK REMAINS UNCHANGED.
```

The response to message DMSFOR603R was NO or a null line was entered.

```
DMSFOR732I 'nnnnnnnnnn' FB-512 BLOCKS FORMATTED
           ON DISK 'mode(cuu)'
```

The format operation is complete.

```
DMSFOR733I FORMATTING DISK 'mode'
```

The disk represented by mode letter 'mode' is being formatted.

```
LABEL CUU M STAT CYL TYPE BLKSIZE FILES BLKS USED-(%) BLKS LEFT BLK TOTAL
label cuu m  R/W nnn type blksize  nnnnn      nnnn- %       nnn     nnnnnn
```

This message provides the status of a disk when you use the RECOMP option. The response is the same as when you issue the QUERY command with the DISK operand.

Other Messages and Return Codes:

```
DMSFOR003E INVALID OPTION 'option'  RC=24
DMSFOR005E NO 'option' SPECIFIED  RC=24
DMSFOR017E INVALID DEVICE ADDRESS 'cuu'  RC=24
DMSFOR028E NO DEVICE SPECIFIED  RC=24
DMSFOR037E DISK 'mode[(cuu)]' IS READ/ONLY  RC=36
DMSFOR048E INVALID MODE 'mode'  RC=24
DMSFOR069E DISK 'mode' NOT ACCESSED  RC=36
DMSFOR070E INVALID PARAMETER 'parameter'  RC=24
DMSFOR113S DEVICE 'cuu' NOT ATTACHED  RC=100
DMSFOR114S 'cuu' IS AN UNSUPPORTED DEVICE TYPE
           OR REQUESTED BLKSIZE IS NOT SUPPORTED
           FOR THE DEVICE  RC=88
DMSFOR125S PERMANENT UNIT CHECK ON DISK 'mode(cuu)' RC=100
DMSFOR126S ERROR {READ|WRIT}ING LABEL ON DISK 'mode(cuu)' RC=100
DMSFOR214W CANNOT RECOMPUTE WITHOUT LOSS OF DATA. NO CHANGE  RC=8
DMSFOR216E INSUFFICIENT BLOCKS ON DISK TO SUPPORT
           CMS DISK STRUCTURE  RC=100
```

GENDIRT

Use the GENDIRT command to fill in a CMS auxiliary directory. The auxiliary directory contains the name and location of modules that would otherwise significantly increase the size of the resident directory, thus increasing search time and storage requirements. By using GENDIRT to fill in an auxiliary directory, the file entries for the given command are loaded only when the command is invoked. The format of the GENDIRT command is:

GENDIRT	directoryname [targetmode]

For a complete description of the GENDIRT command, refer to the *VM/SP CMS Command and Macro Reference* and *VM/SP System Programmer's Guide.*

GENMOD

Use the GENMOD command to generate a module file on a CMS disk. The format of the GENMOD command is:

Genmod	[fn [MODULE [fm / A1]]] [(options...[)]] options: [FROM entry1] [TO entry2] [MAP / NOMAP] [STR / NOSTR] [OS / ALL] [SYSTEM]

Where:

fn
: is the filename of the MODULE file being created. If fn is not specified, the file created has a filename equal to that of the first entry point in the LOAD MAP.

fm
: is the filemode of the MODULE file being created. If fm is not specified, A1 is assumed.

Options: If conflicting options are specified, the last one entered is used.

FROM entry1
: specifies an entry point or a control section name that represents the starting virtual storage location from which the copy is generated.

TO entry2
: specifies an entry point or a control section name that represents the ending virtual storage location from which the copy is generated.

MAP
: copies system loader table entries for the generated module into a load map record which is included at the end of the MODULE file. The record can contain as many as 3276 load map entries. The MODMAP command can be issued to display the load map.

NOMAP
: specifies that a load map is not to be contained in the MODULE file.

Note: If a module is generated with the NOMAP option, that module cannot later be loaded and started with the CMS LOADMOD and START commands. When NOMAP is specified, the information produced is not sufficient for the START command to execute properly. However, a module generated with the NOMAP option can later be invoked as a command; that is, it can be invoked if its filename is entered.

STR
: invokes the CMS storage initialization routine when the MODULE file is subsequently loaded (see the LOADMOD command description). This routine frees any storage remaining from a previous program. STR is the default setting if the MODULE is to be loaded at the beginning of available user storage.

  If you have issued CMS SET RELPAGE ON, STR causes CMS storage initialization to release the remaining pages of storage.

  Note: If a program running in the user area calls a transient routine that was generated with the STR option, the user area storage pointers will be reset. This reset condition could cause errors upon return to the original program (for example, when OS GETMAIN/FREEMAIN macros are issued in the user program).

NOSTR
: indicates that, when the MODULE is loaded, free storage pointers are not reset for any storage currently in use. NOSTR is the default setting if the MODULE file is to be loaded at a location other than the default load address.

SYSTEM
: indicates that when the MODULE file is subsequently loaded, it is to have a storage protect key of zero.

OS
indicates that the program may contain OS macros and, therefore, should be executed only when OS Simulation is available.

ALL
indicates that the program:

- Contains CMS macros and must be capable of running regardless of whether OS Simulation is available or not
- Contains no OS macros

Usage Notes:

1. The GENMOD command is usually invoked following the LOAD command, and possibly the INCLUDE command. For example, the sequence:

```
load myprog
genmod testprog
```

loads the file MYPROG TEXT into virtual storage and creates a nonrelocatable load module named TESTPROG MODULE. TESTPROG may now be invoked as a user-written command from the CMS environment.

2. Before the file is written, undefined symbols are set to location zero and the common reference control section is initialized. The undefined symbols are not retained as unresolved symbols in the MODULE file. Therefore, once the MODULE file is generated, those references cannot be resolved and may cause unpredictable results during execution.

3. If you load a program into the transient area you should issue the GENMOD command with the STR option. Be careful if the program uses OS GETMAIN or FREEMAIN macros because your program, plus the amount of storage obtained via GETMAIN, cannot

exceed two pages (8192 bytes). It is recommended that you do not use GETMAIN macros in programs that execute in the transient area.

4. A transient module (loaded with the ORIGIN TRANS option) that was generated with the SYSTEM option is written on disk as a fixed-length record with a maximum length of 8192 bytes.

5. If you are using FORTRAN under CMS, use FROM MAIN as an option to avoid unpredictable results.

6. If FROM is not specified on the GENMOD command, the starting virtual storage location (entry point) of the module is either the address of fn (if it is an external name) or the entry point determined according to the hierarchy discussed in Usage Note 4 of the LOAD command description in *VM/SP Command and Macro Reference*. This is not necessarily the lowest address loaded. If you have any external references before your START or CSECT instructions, you must specify the 'FROM entry1' operand on the GENMOD command to load your program properly.

7. If you are using PL/I under CMS, use FROM PLISTART as an option to avoid unpredictable results.

8. In order to generate a CMS module which can be loaded and executed as a relocatable nucleus extension, the RLDSAVE option should be used on the LOAD command. GENMOD will then append the relocation data to the module file for use during the loading of the module as a nucleus extension. For example, the sequence

```
load myprog (rldsave gen testprog
```

   will load the file MYPROG TEXT into virtual storage and create a load module named TESTPROG MODULE which will have its relocation data saved as a record in the MODULE file.

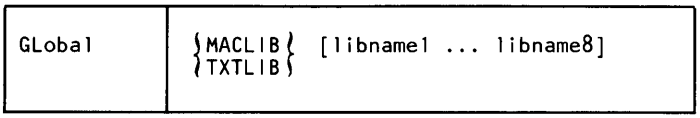
Responses: None.

Messages and Return Codes:

```
DMSMOD003E INVALID OPTION 'option'  RC=24
DMSMOD005E NO {FROM|TO} ENTRY SPECIFIED  RC=24
DMSMOD021E ENTRY POINT 'name' NOT FOUND  RC=40
DMSMOD032E INVALID FILETYPE 'ft'  RC=24
DMSMOD037E DISK 'mode' IS READ/ONLY  RC=36
DMSMOD040E NO FILES LOADED  RC=40
DMSMOD070E INVALID PARAMETER 'parameter'  RC=24
DMSMOD084E INVALID USE OF 'FROM' AND 'TO' OPTIONS  RC=24
DMSMOD105S ERROR 'nn' WRITING FILE 'fn ft fm' ON DISK  RC=100
DMSSTT048E INVALID MODE 'mode'  RC=24
DMSSTT069E DISK 'mode' NOT ACCESSED  RC=36
```

GLOBAL

Use the GLOBAL command to identify which CMS, or OS libraries are to be searched for macros, copy files, or subroutines when processing subsequent CMS commands. The format of the GLOBAL command is:



Where:

`MACLIB`
: precedes the specification of macro libraries that are to be searched for macros and copy files during the execution of language processor commands.

`TXTLIB`
: precedes the specification of text libraries to be searched for missing subroutines when the LOAD or INCLUDE command is issued, or when a dynamic load occurs (that is, when an OS SVC 8 is issued).

  Note: Subroutines that are called by dynamic load should (1) contain only VCONs that are resolved within the same text library member or (2) be resident in storage throughout the processing of the original CMS LOAD or INCLUDE command. Otherwise, the entry point is unpredictable.

`libname1...`
: are the filenames of up to eight libraries. Filetypes must be MACLIB or TXTLIB, accordingly. The libraries are searched in the order in which they are

named. If no library names are specified, the command cancels the effect of any previous GLOBAL command.

Usage Notes:

1. A GLOBAL command remains in effect for an entire CMS session unless it is explicitly canceled or reissued. If a program failure forces you to IPL CMS again, you must reissue the GLOBAL command.

2. There are no default libraries; if you wish to use the same libraries during every terminal session, place the GLOBAL command(s) in your PROFILE EXEC.

3. To find out what libraries have been specified, issue the QUERY command with the MACLIB, TXTLIB, or LIBRARY operands. (The LIBRARY operand requests a display of all libraries.)

4. For information on creating and/or manipulating CMS libraries, see the discussion of the MACLIB and TXTLIB commands.

Responses: None.

Messages and Return Codes:

```
DMSGLB002W FILE 'fn ft' NOT FOUND  RC=28
DMSGLB014E INVALID FUNCTION 'function'  RC=24
DMSGLB047E NO FUNCTION SPECIFIED  RC=24
DMSGLB108S MORE THAN 8 LIBRARIES SPECIFIED  RC=88
```

GLOBALV

The GLOBALV (GLOBAL Variables) command addresses two primary needs: 1) the need for several EXECs to share a common set of values; 2) the need to retain those values, either temporarily or permanently, for subsequent use.

Sharing

Values are often given names, describing what they represent, for easy reference. Although the values often vary, their names usually do not. The GLOBALV command processor builds and maintains group(s) of named, variable values in free storage for shared use by EXECs. EXECs

"share" a value by referring to it by a common name. When requested, GLOBALV retrieves a variable(s) from the group(s) and places it in the program stack for subsequent use by the requesting EXEC.

GLOBALV supports use of more than one group. This allows for grouping distinct variables that are either related or often used together, which facilitates both more efficient retrieval and more selective use. The "global variable group(s)," built by GLOBALV from a set of CMS GLOBALV type files on the user's A-disk and extensions, exist throughout an IPL, unless explicitly purged or re-initialized.

Retaining

When variables are defined or changed, the user decides whether the variables or changes are to last:

1. For the current IPL only
2. Throughout an entire session (normally, from LOGON to LOGOFF)
3. Permanently, i.e. across sessions

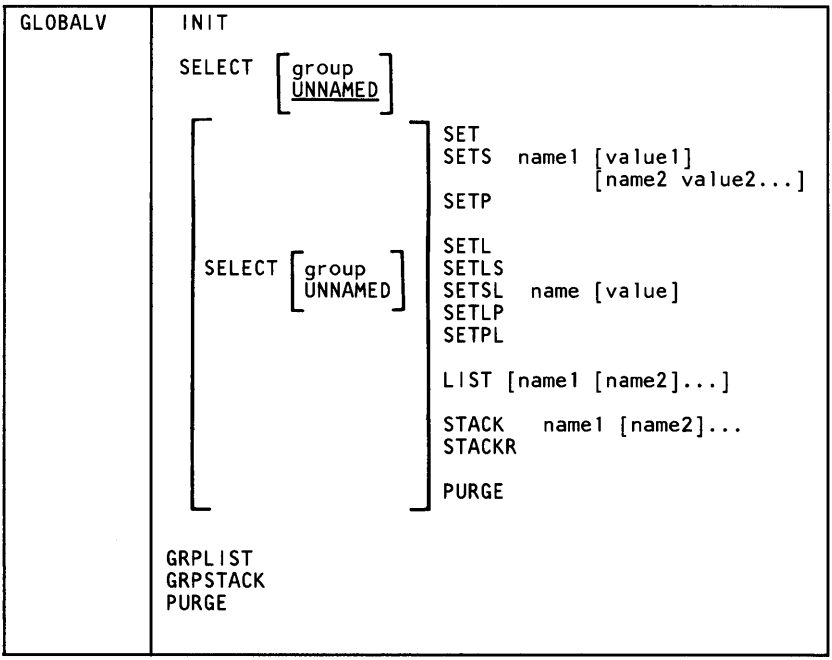
Variables defined for the current IPL only are retained in storage. Those required longer than a single IPL are retained in CMS files on the user's A-disk from where they are put in storage. The CMS filenames are SESSION GLOBALV (for values required throughout the session), and LASTING GLOBALV (for values that are to last permanently). These two files and a third A-disk file (INITIAL GLOBALV) are the source from which the GLOBALV command processor creates and initializes the variable(s) in storage. The INITIAL file is normally created by the user as an alternative way of defining a large number of variables for an IPL.

The CMS GLOBALV disk files may be of fixed or variable format. Fixed format facilitates creation of files by users (via editing). It accommodates variables whose names and values do not exceed eight bytes each. The GLOBALV command processor uses variable format which allows for varying length variable names and values. In addition,

variable format includes a special field which, when used, identifies the group name into which the variable will be grouped.

The GLOBALV command processor manages requests to define or set (SET...) variables both in storage and in the LASTING and SESSION GLOBALV files on the user's A-disk.

The format of the GLOBALV command is:



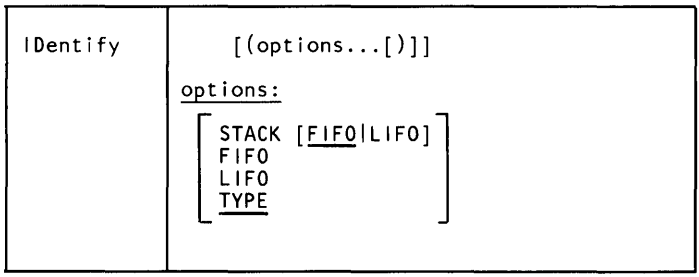
Note: Although this command may be used in CMS EXECs, it is designed for use with EXEC 2 EXECs. Restrictions/ precautions are listed in the *CMS Command and Macro Reference.*

For a description of the GLOBALV command, refer to *VM/SP CMS Command and Macro Reference.*

IDENTIFY

Use the IDENTIFY command to display or stack the following information: your userid, the date, time, and day of the week.

The format of the IDENTIFY command is:



Options:

STACK FIFO|LIFO
: specifies that the information should be placed in the program stack rather than displayed at the terminal. The information is stacked either FIFO (first in first out) or LIFO (last in first out). The default order is FIFO.

FIFO
: specifies that the information should be placed in the program stack rather than displayed at the terminal. The information is stacked FIFO. The options STACK, STACK FIFO, and FIFO are all equivalent.

LIFO
: specifies that the information should be placed in the program stack rather than displayed at the terminal. The information is stacked LIFO. This option is equivalent to STACK LIFO.

TYPE
: specifies that the information should be displayed at the terminal. This is the default option.

Responses: The following information is displayed or stacked:

```
userid AT * VIA * date time zone day
```

Where:

`userid`
is the userid of your virtual machine.

`*`
is a token placeholder representing the RSCS node id, which is not supported in VM/PC.

`*`
is a token placeholder representing the userid of the RSCS virtual machine, which is not supported in VM/PC.

`date`
is the local date, in the form mm/dd/yy.

`time`
is the local time, in the form hh:mm:ss.

`zone`
is the word LOCAL.

`day`
is the day of the week.

Implementation Notes: The userid and node are from the CP QUERY USERID command. The date and time are from the CP QUERY TIME command.

IDENTIFY keeps some of its information in storage, such as userid. To change any of that information, you must issue NUCXDROP IDENTIFY and then reissue the IDENTIFY command.

Error Messages and Return Codes:

```
DMSIDE003E INVALID OPTION 'nn'  RC=24
DMSIDE070E INVALID PARAMETER 'nn'  RC=24
```

IMPORT

Use the IMPORT command to make a copy of a Personal Computer file on a CMS minidisk. The format of the IMPORT command is:

IMPort	d:filename.ext fn ft fm [(options...[])] options: [REP or REPLACE] [LRECL nnnn] [ASCII or NOASCII] [EOL or NOEOL]

Where:

`d:filename.ext`
is the filespec of the Personal Computer file to be imported. The drive and filename must be specified. The filename extension is optional. Remember to include the extension when you refer to a file that has an extension, otherwise VM/PC will be unable to locate the file. No special characters (?,*) are allowed in the filespec.

`fn`
is the filename of the CMS file to create.

`ft`
is the filetype of the CMS file to create.

`fm`
is the filemode of the CMS file to create.

Note: You must specify a fn, ft, and fm when using the IMPORT command.

`REP or REPLACE`
Specifies that an existing CMS file with the fileid specified will be replaced with the specified Personal Computer file.

`LRECL nnnn`
Specifies the logical record length at which the Personal Computer file is to be read, and at which the CMS file will be written (default is 80).

ASCII or NOASCII

ASCII specifies that the file is to be translated from ASCII to EBCDIC during the IMPORT. NOASCII specifies that the file is not to be translated from ASCII to EBCDIC during the IMPORT. ASCII is the default.

EOL or NOEOL

EOL specifies that the source file is to be scanned for end-of-line characters (carriage return/line feed) which will be interpreted as end-of-line delimiters in creating the CMS file. NOEOL specifies that the source file is not to be scanned for end-of-line characters. In this case, the file created will be an F-format file. EOL is the default.

Notes:

1. If the EOL option is specified, the Personal Computer DOS file will be scanned up to the LRECL value for end-of-line characters. If no end-of-line characters are found before the LRECL value is reached, the error message:

   ```
   Import terminated, file 'd:filename.ext' truncated
   ```

   is displayed, and the import is not done. The logical record length (either the default or user specified LRECL) must exceed the length of the largest line in the Personal Computer DOS file.

2. If the NOEOL option is specified to import a file, the CMS file will be an F-format file, with a logical record length equal to the LRECL used to import the file (either the default, 80, or a user specified LRECL).

   If end-of-file is detected prior to reading in this amount, then the CMS file created will have a logical record length equal to the amount read. Thus, if the Personal Computer file has 38 bytes, and is imported without specifying a LRECL value, the CMS file produced will have 38 bytes.

   If the last read of the Personal Computer file results in less than the LRECL amount of data, it will be padded with X'00' to the LRECL length, except in the case just

noted. If a PC file with 100 bytes is to be imported, the resultant CMS file will have 2 records of 80 bytes each, with the second record padded with x'00' to a length of 80.

3. If the EOL option is used, the maximum LRECL value which may be specified is 65535. If the NOEOL option is used, the only restriction on the LRECL value is the size of your virtual machine, since an I/O buffer this size must be acquired.

4. A file which has been exported, translated to ASCII, and subsequently imported may differ from the original file, due to the translate tables used in the file transformations. The NOASCII options should be used to avoid problems caused by this transformation. Appendix E, "ASCII to EBCDIC Conversion" on page E-1 lists the EBCDIC-ASCII conversion tables which are used in this translation.

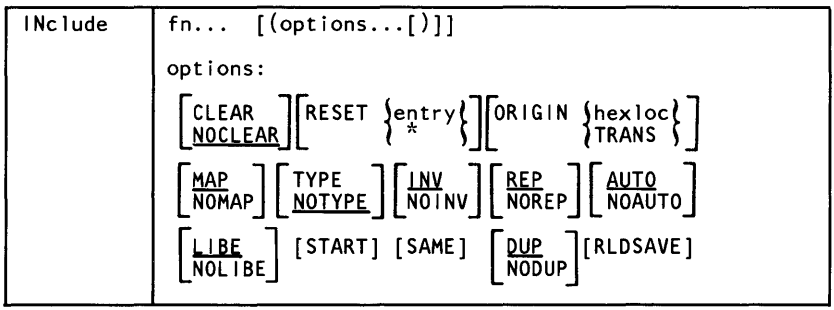
Messages and Return Codes:

```
DMSIMP002E File 'd:filename.ext' not found   RC=28
DMSIMP003E Invalid option 'option'   RC=24
DMSIMP024E File 'fn ft fm' already exists -- specify
           'replace'   RC=28
DMSIMP029E Invalid parameter 'parameter' in the option
           'option' field   RC=24
DMSIMP037E Disk 'mode' is read/only   RC=36
DMSIMP042E No fileid(s) specified   RC=24
DMSIMP048E Invalid mode 'mode'   RC=24
DMSIMP054E Incomplete fileid specified   RC=24
DMSIMP062E Invalid * in fileid   RC=20
DMSIMP070E Invalid parameter 'parameter'   RC=24
DMSIMP105E Error 'nn' writing file 'fn ft fm' on disk   RC=100
DMSIMP622E Insufficient storage available   RC=41
DMSIMP804E Invalid PC fileid 'd:filename.ext' RC=20
DMSIMP805E Import terminated, file 'd:filename.ext'
           truncated  RC=100
DMSIMP807S Drive d not ready. Make drive ready and press enter
DMSIMP907E I/O error on file 'd:filename.ext'  RC=256
DMSSTT048E Invalid mode 'mode'   RC=24
DMSSTT062E Invalid character 'char' in fileid 'fn ft fm'   RC=20
DMSSTT069E Disk 'mode' not accessed   RC=36
```

INCLUDE

Use the INCLUDE command to read one or more TEXT files (containing relocatable object code) from disk and to load them into virtual storage, establishing the proper linkages between the files. A LOAD command must have been previously issued for the INCLUDE command to produce desirable results. For information on the CMS

loader and the handling of unresolved references, see the description of the LOAD command. The format of the INCLUDE command is:

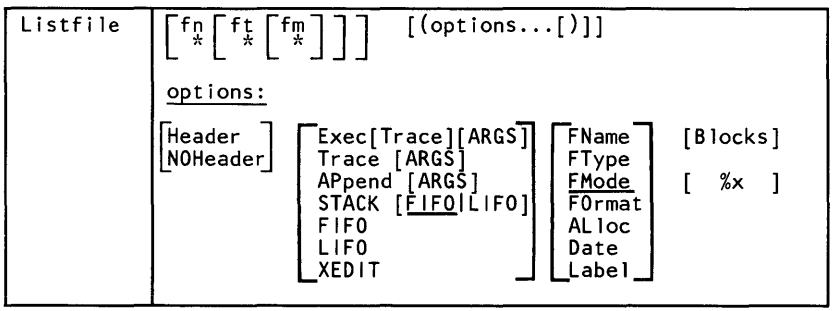


The RLDSAVE option controls whether the CMS loader saves relocation information in an internal buffer as the text files are loaded. If this information is saved, the GENMOD command will append it to the CMS module when it is created. This information is used by the NUCXLOAD command, if the module is loaded as a nucleus extension, to relocate the module to the area where it has been loaded.

For a complete description of the INCLUDE command, refer to *VM/SP CMS Command and Macro Reference.*

LISTFILE

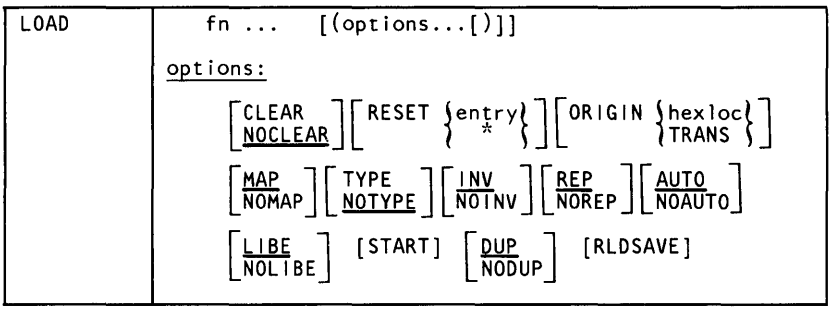
Use the LISTFILE command to obtain specified information about CMS files residing on accessed disks. The format of the LISTFILE command is:



For a complete description of the LISTFILE command, refer to the *VM/SP CMS Command and Macro Reference.*

LOAD

Use the LOAD command to read one or more CMS TEXT files (containing relocatable object code) from disk and to load them into virtual storage, establishing the proper linkages between the files. The format of the LOAD command is:

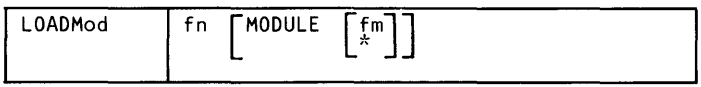


The RLDSAVE option controls whether the CMS loader saves relocation information in an internal buffer as the text files are loaded. If this information is saved, the GENMOD command will append it to the CMS module when it is created. This information is used by the NUCXLOAD command, if the module is loaded as a nucleus extension, to relocate the module to the area where it has been loaded.

For a complete description of the LOAD command, refer to the *VM/SP CMS Command and Macro Reference.*

LOADMOD

Use the LOADMOD command to load a MODULE file into storage. The file must be in the format which is created by the GENMOD command. The format of the LOADMOD command is:



Where:

`fn`
is the filename of the file to be loaded into storage. The filetype must be MODULE.

fm
: is the filemode of the module to be loaded. If not specified, or specified as an asterisk, all your disks are searched for the file.

Usage Note:

You can use the LOADMOD command when you want to debug a CMS MODULE file. After the file is loaded, you may set address stops or breakpoints before you begin execution with the START command; for example:

```
loadmod prog1
cp trace i 210ae.4
start
```

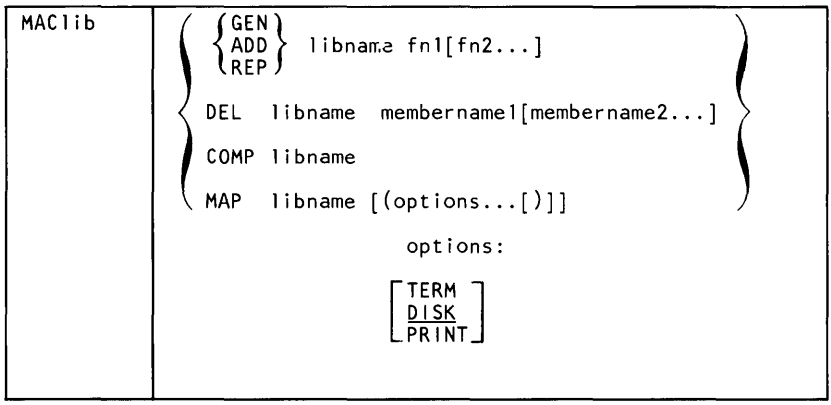
Responses: None.

Messages and Return Codes:

```
DMSMOD001E NO FILENAME SPECIFIED  RC=24
DMSMOD002E FILE 'fn ft' NOT FOUND  RC=28
DMSMOD032E INVALID FILETYPE 'ft'  RC=24
DMSMOD070E INVALID PARAMETER 'parameter'  RC=24
DMSMOD104S ERROR 'nn' READING FILE 'fn ft fm' FROM DISK  RC=100
DMSMOD109S VIRTUAL STORAGE CAPACITY EXCEEDED  RC=104
DMSMOD116S LOADER TABLE OVERFLOW  RC=104
DMSSTT048E INVALID MODE 'mode'  RC=24
DMSSTT062E INVALID CHARACTER 'character' IN FILEID 'fn ft fm' RC=20
```

MACLIB

Use the MACLIB command to create and modify CMS macro libraries. The format of the MACLIB command is:



For a complete description of the MACLIB command, refer to the *VM/SP CMS Command and Macro Reference.*

MAKEBUF

Use the MAKEBUF command to create a new buffer within the program stack. The format of the MAKEBUF command is:

MAKEBUF	

Usage Notes:

1. When you issue a MAKEBUF command, CMS returns as a return code the number of the program stack buffer just created. If you issue a MAKEBUF command in an EXEC that has the &ERROR statement in effect, the MAKEBUF return code causes the &ERROR statement to execute.

2. Use the WAITRD function to read lines from the buffers the MAKEBUF command creates. WAITRD first reads lines from the most recently created buffer. When the most recent buffer is exhausted, WAITRD reads the next most recent buffer. When all program stack buffers are exhausted, WAITRD reads from the terminal input buffer.

MODMAP

Use the MODMAP command to display the load map associated with the specified MODULE file. The format of the MODMAP command is:

MODmap	fn

For a complete description of the MODMAP command, refer to the *VM/SP CMS Command and Macro Reference.*

NUCXDROP

Use the NUCXDROP command to cancel nucleus extensions and release the storage occupied by the corresponding program. The NUCXDROP command uses the NUCEXT function which is described in detail in the Function section of the publication *VM/SP CMS Command and Macro Reference.*

The command format is:

NUCXDROP	{name1 name2.... {* }

For a complete description of the NUCXDROP command, refer to the *VM/SP CMS Command and Macro Reference.*

NUCXLOAD

Use NUCXLOAD to install nucleus extensions. The command loads either an ADCON-free, serially reusable, or relocatable module into free storage and installs it as a nucleus extension. The nucleus extension is invoked by issuing the name of the nucleus extension. The NUCXLOAD command uses the NUCEXT function which is described in detail in the Function section of the publication *VM/SP CMS Command and Macro Reference.*

The format of the command is:

NUCXLOAD	command [fname [ftype [fmode]]] [(options[)]] options: [SYstem] [SErvice] [Push]

Where:

`command`
is the name of the command to be installed as a nucleus extension.

`fname`
is the filename of the module to be so installed. If this item is omitted, it will be defaulted to "command."

`ftype`
is the filetype of the module to be so installed. If this item is omitted, it will be defaulted to "MODULE."

`fmode`
is the filetype of the module to be so installed. If this item is omitted, it will be defaulted to "*."

`options`
may be any of the following (minimum abbreviation in capitals):

`SYstem`
the program will be loaded in nucleus-type free storage and will not be dropped if an abnormal end occurs.

`SErvice`
the program will accept service calls.

`Push`
no check will be made for an existing nucleus extension of the same name.

Return codes:

```
 10 - Invalid argument list on command
 21 - Length mismatch when reading module
 22 - Invalid format for RLD record
 23 - A Y-CON RLD entry was found which could not be relocated
 31 - Insufficient preallocated storage available to load module
 32 - Storage not available for header record buffer
 33 - Storage not available to load the module
 34 - Storage not available for RLD record buffer
 41 - Nucleus Extension of same name already exists
4xx - FSSTATE error ("xx" is FSSTATE return code)
5xx - Error reading module header record ("xx" is FSREAD return code)
6xx - Error reading module code ("xx" is FSREAD return code)
7xx - Error reading RLD record ("xx" is FSREAD return code)
8xx - Error from NUCEXT ("xx" is NUCEXT return code)
```

NUCXMAP

Use the NUCXMAP command to get information about the currently defined nucleus extensions. NUCXMAP displays on the console or stacks a list of the nucleus extensions. The NUCXMAP command uses the NUCEXT function which is described in detail in the Function section of the *VM/SP CMS Command and Macro Reference.*

The command format is:

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

For a complete description of the NUCXMAP command, refer to the *VM/SP CMS Command and Macro Reference.*

PRINT

Use the PRINT command to print a CMS file on the spooled printer. The format of the PRINT command is:

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

        options:

        [CC [HEADer]]          [TRC   ]
        [NOCC       ] [UPCASE] [NOTRC ]

        [LINECOUN  nn][MEMBER { *          }][HEX]
                      [55]    { membername }
```

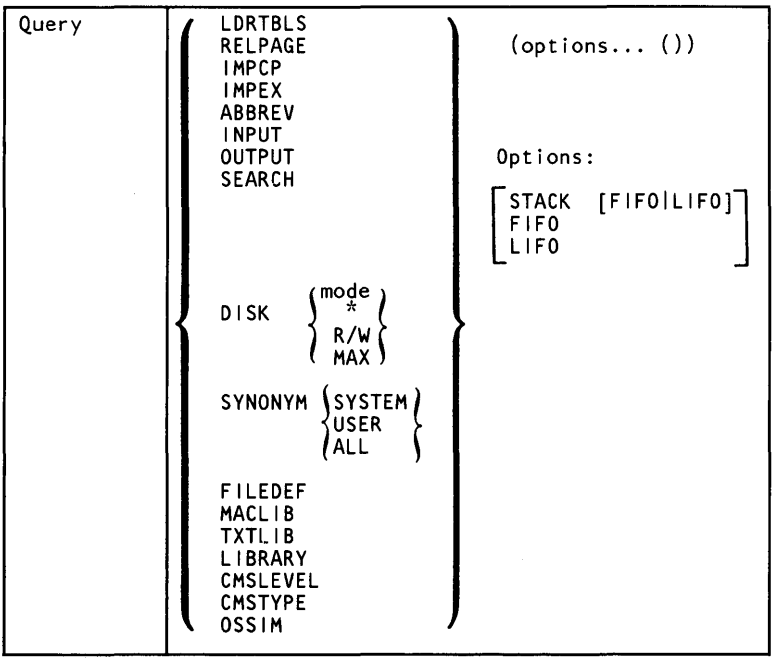
For a complete description of the PRINT command, refer to the *VM/SP CMS Command and Macro Reference.*

QUERY

Use the QUERY command to gather information about your CMS virtual machine. You can determine:

- The state of virtual machine characteristics that are controlled by the CMS SET command
- File definitions (set with the FILEDEF command) that are in effect
- The status of accessed disks

The format of the QUERY command is:



Operands for Functions that Can Be Controlled Via the SET Command:

LDRTBLS
displays the number of loader tables.

Response: LDRTBLS = nn

RELPAGE
indicates whether pages of storage are to be released or retained after certain commands complete execution.

Response:

```
RELPAGE = {ON }
          {OFF}
```

Where:

ON
releases pages.

OFF
retains pages.

IMPCP
displays the status of implied CP command indicator.

Response:

```
IMPCP = {ON }
        {OFF}
```

Where:

ON
indicates that CP commands can be entered from the CMS environment.

OFF
indicates that you must use the CP command or the #CP function to enter CP commands from the CMS environment.

IMPEX
displays status of implied EXEC indicator.

Response:

```
IMPEX = {ON }
        {OFF}
```

Where:

ON
: indicates that EXEC files can be executed by entering the filename of the file.

OFF
: indicates that the EXEC command must be explicitly entered to execute EXEC files.

ABBREV
: displays the status of the minimum truncation indicator.

Response:

```
ABBREV = {ON }
         {OFF}
```

Where:

ON
: indicates that truncations are accepted for CMS commands.

OFF
: indicates that truncations are not accepted.

INPUT
: displays the contents of any input translate table in effect.

Response:

```
INPUT  a1  xx1
       .   .
       .   .
       .   .
       an  xxn
```

If you do not have an input translate table in effect, the response is:

```
NO USER DEFINED INPUT TRANSLATE TABLE IN USE
```

OUTPUT
: displays the contents of any output translate table in effect.

Response:

```
OUTPUT  xx1  a1
        .    .
        .    .
        .    .
        xxn  an
```

If you do not have an output translate table defined, the response is:

```
NO USER DEFINED OUTPUT TRANSLATE TABLE IN USE
```

Operands for CMS Disk Status Functions:

`SEARCH`
displays the search order of all disks currently accessed.

Response:

```
label  cuu  mode {R/O}
                {R/W}
  .     .     .    .
  .     .     .    .
  .     .     .    .
```

Where:

`label`
is the label assigned to the disk when it was formatted; or, if it is an OS or DOS disk, the volume label.

`cuu`
is the virtual device address.

`mode`
is the filemode letter assigned to the disk when it was accessed.

$\left\{\begin{matrix}\text{R/O}\\\text{R/W}\end{matrix}\right\}$
indicates whether read/write or read-only is the status of the disk.

`DISK mode`
displays the status of the single disk represented by "mode."

Response:

```
LABEL CUU M STAT  CYL TYPE BLKSIZE  FILES BLKS USED-(%) BLKS LEFT BLK TOTAL
label cuu m {R/O} FB  type blksize   nnnn      nnnn-nn       nnnn     nnnnn
           {R/W}
```

Where:

`label`
is the label assigned to the disk when it was formatted.

`cuu`
is the virtual device address.

`m`
is the access mode letter.

`{R/O} STAT`
indicates whether read/write or read-only

`{R/W}`
is the status of the disk.

`cyl`
is the number of cylinders available on the disk. For an FB-512 device, this field contains the notation 'FB' rather than the number of cylinders.

`type`
is the device type of the disk or the designation drive id: PC if the minidisk is on a local Personal Computer disk.

`blksize`
is the CMS disk block size when the minidisk was formatted.

`nnnn FILES`
is the number of CMS files on the disk.

`nnnn BLKS USED`
indicates the number of CMS disk blocks in use.

`nn`
indicates the percentage of blocks in use.

`nnnn BLKS LEFT`
indicates the number of disk blocks left. This is a high approximation because control blocks are included.

nnnnn BLK TOTAL
indicates the total number of disk blocks.

If the disk with the specified mode is not accessed, the response is:

```
DISK 'mode' NOT ACCESSED
```

DISK *
displays the status of all CMS disks.

Response: Is the same as for QUERY DISK mode; one line is displayed for each accessed disk.

DISK R/W displays the status of all CMS disks that have been accessed in the Read/Write mode.

Response: Is of the same format as QUERY DISK mode; one header is displayed followed by one line for each accessed CMS Read/Write disk.

DISK MAX displays the status of the CMS disk accessed in Read/Write mode having the most available space.

Response: Is of the same format as QUERY DISK mode; a header and one line are displayed for the CMS Read/Write disk with the most available space.

Other Functions: "SYNONYM SYSTEM" displays the CMS system synonyms in effect.

Response:

```
SYSTEM    SHORTEST
COMMAND   FORM
-------   ------------------
command   minimum truncation
   .            .
   .            .
   .            .
```

If no system synonyms are in effect, the following message is displayed at the terminal:

NO SYSTEM SYNONYMS IN EFFECT

SYNONYM USER
displays user synonyms in effect.

Response:

```
SYSTEM  USER      SHORTEST
COMMAND SYNONYM   FORM (IF ANY)
------- -------   -----------------
command synonym   minimum truncation
   .       .             .
   .       .             .
   .       .             .
```

If no user synonyms are in effect, the following message is displayed at the terminal:

NO USER SYNONYMS IN EFFECT

SYNONYM ALL
displays all synonyms in effect.

Response: The response to the command QUERY SYNONYM SYSTEM is followed by the response to QUERY SYNONYM USER.

FILEDEF
displays all file definitions in effect.

Response:

```
[ddname device [fn [ft]]
   .      .      .   .
   .      .      .   .
   .      .      .   .
```

If no file definitions are in effect, the following message is displayed at the terminal:

NO USER DEFINED FILEDEF'S IN EFFECT

MACLIB
displays the names of all files, with a filetype of MACLIB, that are to be searched for macro definitions (that is, all MACLIBs specified on the last GLOBAL MACLIB command, if any).

Response: `MACLIB = libname...`

If no macro libraries are to be searched for macro definitions, the response is:

```
MACLIB = NONE
```

TXTLIB
: displays the names of all files, with a filetype of TXTLIB, that are to be searched for unresolved references (that is, all TXTLIBs specified on the last GLOBAL TXTLIB command, if any).

Response: `TXTLIB = libname...`

If no TXTLIBs are to be searched for unresolved references, the following message is displayed at the terminal:

TXTLIB = NONE

LIBRARY
: displays the names of all library files with filetypes of MACLIB and TXTLIB that are to be searched.

Response:

```
MACLIB =  { libname... }
          { NONE       }

TXTLIB =  { libname... }
          { NONE       }
```

CMSLEVEL returns the feature or program product, release, and the service level of CMS.

Response: Displays the VM/PC Service Level.

For example: VM/PC Version 1.00, Service Level 000

Usage Notes:

1. You may specify only one QUERY parameter at a time.

2. If the implied CP (IMPCP) function is in effect and you enter an invalid QUERY parameter, you may receive the message DMKCQR026E - OPERAND MISSING OR INVALID.

3. If an invalid QUERY parameter is specified from an EXEC and the implied CP (IMPCP) function is in effect, then the return code is -0003.

4. When the STACK option is specified, the header is included in the program stack.

CMSTYPE returns the current setting of CMSTYPE, either:

```
CMSTYPE = RT
    or
CMSTYPE = HT
```

OSSIM displays whether the OS Simulation environment is available for use.

STACK
: causes the results of the QUERY command to be placed in the program stack instead of being displayed at the terminal. The information is stacked either FIFO (first-in first-out) or LIFO (last-in first-out). The default order is FIFO.

  If CMS passes the command to CP, then the response from CP is also put in the program stack. If CP precedes the QUERY command, CMS does not stack the results. The STACK option is valid only when issued from CMS.

FIFO
: (first-in first-out) is the default option for STACK. FIFO causes the results of the QUERY command to be placed in the program stack instead of being displayed at the terminal. The information is stacked FIFO. The options STACK, STACK FIFO, and FIFO are all equivalent.

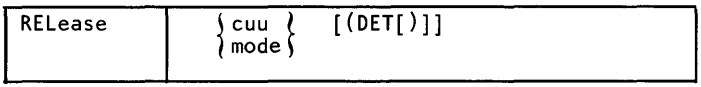
LIFO
: (last-in first-out) causes the results of the QUERY command to be placed in the program stack instead of being displayed at the terminal. The information is stacked LIFO. This option is equivalent to STACK LIFO.

Error Messages and Return Codes:

```
DMSQRY005E NO 'option' SPECIFIED  RC=24
DMSQRY014E INVALID FUNCTION 'function'  RC=24
DMSQRY026E INVALID PARAMETER 'parameter' FOR 'function' FUNCTION  RC=24
DMSQRY047E NO FUNCTION SPECIFIED  RC=24
DMSQRY070E INVALID PARAMETER 'parameter'  RC=24
```

RELEASE

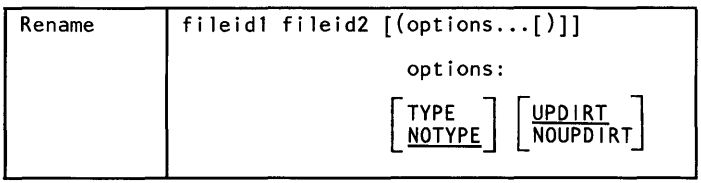
Use the RELEASE command to free an accessed disk and make the files on it unavailable. The format of the RELEASE command is:



For a complete description of the RELEASE command, refer to the *VM/SP CMS Command and Macro Reference.*

RENAME

Use the RENAME command to change the fileid of one or more CMS files on a read/write CMS disk. The format of the RENAME command is:



For a complete description of the RENAME command, refer to the *VM/SP CMS Command and Macro Reference.*

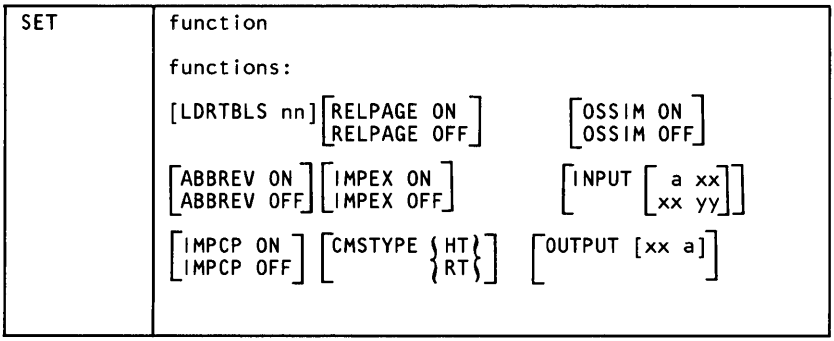
SENTRIES

Use the SENTRIES command to determine the number of lines currently in the program stack. When you issue a SENTRIES command, CMS returns the number of lines in the program stack (but not the console input buffer) as a return code. The format of the SENTRIES command is:

Usage Notes: If you issue a SENTRIES command in an EXEC that has the &ERROR statement in effect, a nonzero SENTRIES return code causes the &ERROR statement to execute.

SET

Use the SET command to establish, turn off, or reset a particular function in your CMS virtual machine. Only one function may be specified per SET command. The format of the SET command is:



Where functions:

`LDRTBLS nn`

defines the number (nn) of pages of storage to be used for loader tables. By default, a virtual machine having up to 384K of addressable real storage has two pages of loader tables; a larger virtual machine has three pages. Each loader table page has a capacity of 204 external names. During LOAD and INCLUDE command processing, each unique external name encountered in a TEXT deck is entered in the loader table. The LOAD command clears the table before reading TEXT files; INCLUDE does not. This number can be changed with the SET LDRTBLS nn command provided that: (1) nn is a decimal number between 1 and 127, inclusive, and (2) the virtual machine has enough storage available to allow nn pages to be used for loader tables. If these two conditions are met, nn pages are set aside for loader tables. If you plan to change the number of pages allocated for loader tables, you should deallocate storage at the high end of storage so that the storage for the loader tables may be obtained from that area. Usually, you can deallocate storage by releasing one or more of the disks that were accessed.

RELPAGE ON
releases page frames of storage and sets them to binary zeros after the following commands complete execution: COPYFILE, COMPARE, MACLIB, SORT, TXTLIB, UPDATE, and the program product language processors supported by VM/PC.

RELPAGE OFF
does not release pages of storage after the commands listed in the RELPAGE ON description complete execution. Use the SET RELPAGE OFF function when debugging or analyzing a problem so that the storage used is not released and can be examined.

OSSIM ON
loads the OSSIM module into CMS free storage to provide the services of OS Simulation. This command is normally used in the SYSPROF exec (system profile exec) which is normally executed when CMS is started.

OSSIM OFF
may be used to unload the OS Simulation services of CMS.

INPUT a xx
translates the specified character "a" to the specified hexadecimal code "xx" for characters entered from the terminal.

INPUT xx yy
allows you to reset the hexadecimal code "xx" to the specified hexadecimal code "yy" in your translate table.

Note: If you issue SET INPUT and SET OUTPUT commands for the same characters, issue the SET OUTPUT command first.

INPUT
returns all characters to their default translation.

OUTPUT xx a
translates the specified hexadecimal representation "xx" to the specified character "a" for all "xx" characters displayed at the terminal.

OUTPUT
returns all characters to their default translation.

Note: Output translation does not occur for SCRIPT files when the SCRIPT command output is directed to the terminal, nor when you use the CMS editor on a display terminal in display mode.

`ABBREV ON`
accepts system and user abbreviations for system commands. The SYNONYM command makes the system and user abbreviations available.

`ABBREV OFF`
accepts only the full system command name or the full user synonym (if one is available) for system commands.

For a discussion of the relationship of the SET ABBREV and SYNONYM commands, refer to the SYNONYM command description.

`IMPEX ON`
treats EXEC files as commands; an EXEC file is invoked when the filename of the EXEC file is entered.

`IMPEX OFF`
does not consider EXEC files as commands. You must issue the EXEC command to execute an EXEC file.

`IMPCP ON`
passes command names that CMS does not recognize to CP; that is, unknown commands are considered to be CP commands.

`IMPCP OFF`
generates an error message at the terminal if a command is not recognized by CMS.

`CMSTYPE HT`
suppresses CMS terminal display within an EXEC. All CMS terminal display from an EXEC, except for CMS error messages with a suffix letter of 'S' or 'T', is suppressed until the end of the EXEC file or until a SET CMSTYPE RT command is executed.

`CMSTYPE RT`
resumes CMS terminal display which has been suppressed as a result of a previous SET CMSTYPE HT command.

Note: &STACK HT and SET CMSTYPE HT have the same effect when interpreted by the CMS EXEC processor. Similarly, &STACK RT and SET CMSTYPE RT are equivalent for the CMS EXEC processor. However, when using EXEC 2, the commands &STACK HT and &STACK RT cause the characters "HT" and "RT" to be placed in the program stack and do not affect the console output. These characters must be used by a program or cleared from the stack. Otherwise, you will receive an "UNKNOWN CP/CMS COMMAND" error message when they are read from the program stack.

Usage Notes:

1. If you issue the SET command specifying an invalid function and the implied CP function is in effect, you may receive message DMKCFS026E OPERAND MISSING OR INVALID.

2. If an invalid SET command function is specified from an EXEC and the implied CP function is in effect, then the return code is -0003.

Responses: None. To determine or verify the setting of a function, use the QUERY command.

Messages and Return Codes:

```
DMSLIO002I FILE 'fn' TXTLIB NOT FOUND  RC=0
DMSSET014E INVALID FUNCTION 'function'  RC=24
DMSSET026E INVALID PARAMETER 'parameter' FOR 'function' FUNCTION  RC=
DMSSET031E LOADER TABLES CANNOT BE MODIFIED  RC=40
DMSSET047E NO FUNCTION SPECIFIED  RC=24
DMSSET048E INVALID MODE 'mode'  RC=24
DMSSET050E PARAMETER MISSING AFTER 'function'  RC=24
DMSSET061E NO TRANSLATION CHARACTER SPECIFIED  RC=24
DMSSET070E INVALID PARAMETER 'parameter'  RC=24
DMSSET098W CMS OS SIMULATION NOT AVAILABLE  RC=4
DMSSET100W SYSTEM NAME 'name' NOT AVAILABLE  RC=4
```

SORT

Use the SORT command to read fixed-length records from a CMS input file, arrange them in ascending EBCDIC order according to specified sort fields, and create a new file containing the sorted records. The format of the SORT command is:

SORT	fileid1 fileid2

For a complete description of the SORT command refer to the *VM/SP CMS Command and Macro Reference* publication.

For an example of using the SORT command, see "CMS SORT Command" on page 5-24.

START

Use the START command to begin execution of CMS programs that were previously loaded. The format of the START command is:

START	[entry [args...]] [*] [(option [)]]	option: NO

For a complete description of the START command, refer to the *VM/SP CMS Command and Macro Reference.*

STATE/STATEW

Use the STATE command to verify the existence of a CMS file on any accessed disk; use the STATEW command to verify the existence of a CMS file on any accessed read/write disk. The formats of the STATE and STATEW commands are:

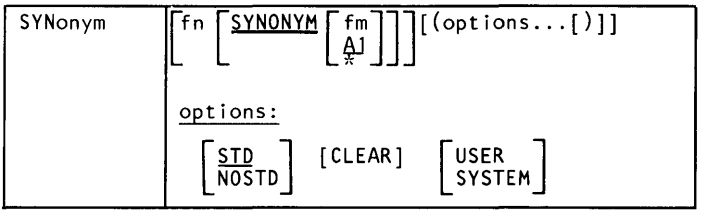
{STATE / STATEW}	{fn / *} {ft / *} {[fm] / *}

For a complete description of the STATE/STATEW command, refer to the *VM/SP CMS Command and Macro Reference.*

SYNONYM

Use the SYNONYM command to invoke a table of synonyms to be used with, or in place of, CMS and user-written command names. You create the table yourself using an editor. The form for specifying the entries for the table is described under "The Synonym Table."

The names you define can be used either instead of or in conjunction with the standard CMS command truncations. However, no matter what truncations, synonyms, or truncations of the synonyms are in effect, the full real name of the command is always accepted. The format of the SYNONYM command is:



Where:

`fn`
is the filename of the file containing your synonyms table.

`fm`
is the filemode of the file containing your synonyms; if omitted, your A-disk and its extensions are searched. If you specify fm, you must enter the keyword, SYNONYM. If you specify fm as an asterisk (*), all disks are searched for the specified SYNONYM file.

Options:

`STD`
specifies that standard CMS abbreviations are accepted.

`NOSTD`
standard CMS abbreviations are not to be accepted. (The full CMS command and the synonyms you defined can still be used.)

`CLEAR`
removes any synonym table set by a previously entered SYNONYM command.

`USER`
specifies that the user synonym table is to be modified.

SYSTEM
specifies that the system synonym table is to be modified.

Usage Notes:

1. If you enter the SYNONYM command with no operands, the system synonym table and the user synonym table (if they exist) are listed.

2. The SET ABBREV ON or OFF command, in conjunction with the SYNONYM command, determines which standard and user-defined forms of a particular CMS command are acceptable.

3. EXEC procedures having a synonym defined for them can be invoked by its synonym if implied EXEC (IMPEX) function is on. However, within an EXEC procedure, only the EXEC filename can be used. A synonym is not recognized within an EXEC since the synonym tables are not searched during EXEC processing.

The Synonym Table: You create the synonym table using an editor. The table must be a file with the filetype SYNONYM. The file consists of 80-byte fixed-length records in free-form format with columns 73-80 ignored. The format for each record is:

```
systemcommand usersynonym count
```

Where:

systemcommand
is the name of the CMS command or MODULE or EXEC file for which you are creating a synonym.

usersynonym
is the synonym you are assigning to the command name. When you create the synonym, you must follow the same syntax rules as for commands; that is, you must use the character set used to create commands, the synonym may be no longer than eight characters, and so on.

count
is the minimum number of characters that must be entered for the synonym to be accepted by CMS. If omitted, the entire synonym must be entered (see the following example).

A table of command synonyms is built from the contents of this file. You may have several synonym files but only one may be active at a time. For example, if the synonym file named MYSYN contains:

```
COPYFILE  CPY
```

then, after you have issued the command:

```
synonym mysyn
```

the synonym CPY can be entered as a command name to execute the COPYFILE command. It cannot be truncated since no count is specified. If MYSYN SYNONYM contains:

```
ACCESS GETDISK 3
```

then, the synonyms GET, GETD, GETDI, GETDIS, or GETDISK can be entered as the command name instead of ACCESS.

If you have an EXEC file named TDISK, you might have a synonym entry:

```
TDISK TDISK 2
```

so that you can invoke the EXEC procedure by specifying the truncation TD.

The Relationship between the SET ABBREV and SYNONYM Commands: The default values of the SET and SYNONYM commands are such that the system synonym abbreviation table is available unless otherwise specified.

The system synonym abbreviation table for the FILEDEF command states that FI is the minimum truncation. Therefore, the acceptable abbreviations for FILEDEF are: FI, FIL, FILE, FILED, FILEDE, and FILEDEF. The

system synonym abbreviation table is available whenever both SET ABBREV ON and SYNONYM (STD) are in effect.

If you have a synonym table with the file identification USERTAB SYNONYM A, that has the entry:

```
FILEDEF USENAME 3
```

then, USENAME is a synonym for FILEDEF, and acceptable truncations of USENAME are: USE, USEN, USENA, USENAM, and USENAME. The user synonym abbreviation table is available whenever both SET ABBREV ON and SYNONYM USERTAB are specified.

No matter what synonyms and truncations are defined, the full real name of the command is always in effect.

A figure follows that lists the forms of the system command and user synonyms available for the various combinations of the SET ABBREV and SYNONYM commands.

Responses: When you enter the SYNONYM command with no operands, the synonym table(s) currently in effect are displayed.

```
SYSTEM      USER        SHORTEST
COMMAND     SYNONYM     FORM (IF ANY)
-------     -------     ------------
   .           .           .
   .           .           .
   .           .           .
```

This response is the same as the response to the command QUERY SYNONYM ALL.

```
DMSSYN7111 NO SYSTEM SYNONYMS IN EFFECT
```

This response is displayed when you issue the SYNONYM command with no operands after the command SYNONYM (NOSTD) has been issued.

Other Messages and Return Codes:

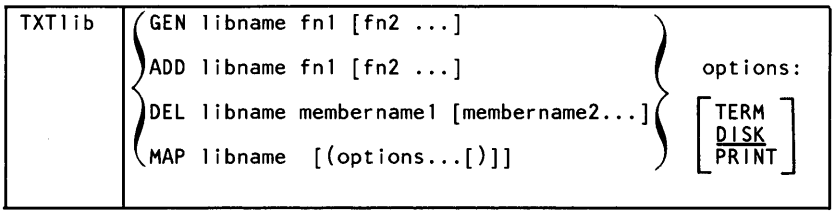
```
DMSSYN002E FILE 'fn ft fm' NOT FOUND  RC=28
DMSSYN003E INVALID OPTION 'option'  RC=24
DMSSYN007E FILE 'fn ft fm' NOT FIXED, 80 CHAR RECORDS  RC=32
DMSSYN032E INVALID FILETYPE 'ft'  RC=24
DMSSYN048E INVALID MODE 'mode'  RC=24
DMSSYN056E FILE 'fn ft fm' CONTAINS INVALID RECORD FORMATS  RC=32
DMSSYN066E 'option AND 'option' ARE CONFLICTING OPTIONS   RC=24
DMSSYN104S ERROR 'nn' READING FILE 'fn ft fm' FROM DISK  RC=100
```

Options	Acceptable Command Forms	Comments
SET ABBREV ON SYN USERTAB (STD	FI FIL . . . FILEDEF USE USEN . . . USENAME	The ABBREV ON option of the SET command and the STD option of the SYNONYM command make the system table available. The user synonym, USENAME, is available because the synonym table (USERTAB) is specified on the SYNONYM command. The truncations for USENAME are available because SET ABBREV ON was specified with the USERTAB also available.
SET ABBREV OFF SYN USERTAB (STD	FILEDEF USENAME	The user-defined synonym, USENAME, is permitted because the user synonym table (USERTAB) is specified on the SYNONYM command. No system or user truncations are permitted.
SET ABBREV ON SYN USERTAB (NOSTD	FILEDEF USE USEN . . . USENAME	The system synonym table is unavailable because the NOSTD option is specified on the SYNONYM command. The user synonym, USENAME, is available because the user synonym table (USERTAB) is specified on the SYNONYM command and the truncations of USENAME are permitted because SET ABBREV ON is specified with USERTAB also available.
SET ABBREV OFF SYN USERTAB (NOSTD	FILEDEF USENAME	The system synonym table is made unavailable either by the SET ABBREV OFF command or by the SYN (NOSTD command. The synonym, USENAME, is permitted because the user-defined synonym table (USERTAB) is specified on the SYNONYM command. The truncations for USENAME are not permitted because the SET ABBREV OFF option is in effect.
SET ABBREV ON SYN (CLEAR STD	FI FIL . . . FILEDEF	The user-defined table is now unavailable. The system synonym table is available because both the ABBREV ON option of the SET command and the STD option of the SYNONYM command are specified.
SET ABBREV OFF SYN (CLEAR STD SET ABBREV ON SYN (CLEAR NOSTD SET ABBREV OFF SYN (CLEAR NOSTD	FILEDEF	Because CLEAR is specified on the SYNONYM command, the synonym and its truncations are no longer available. Either the SET ABBREV OFF command or the SYNONYM (NOSTD command make the system synonym table unavailable.

Figure 7-4. System and User-Defined Truncations

TXTLIB

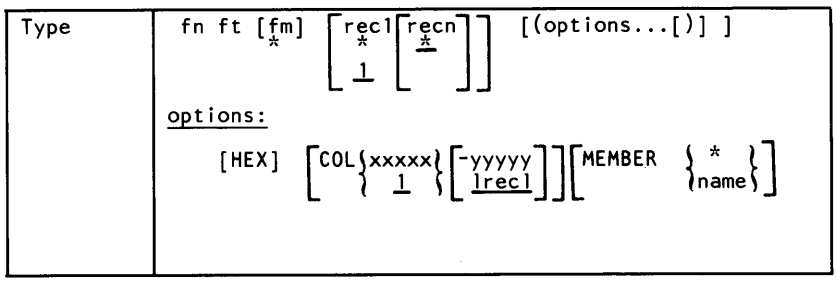
Use the TXTLIB command to update CMS text libraries. The format of the TXTLIB command is:



For a complete description of the TXTLIB command, refer to the *VM/SP CMS Command and Macro Reference.*

TYPE

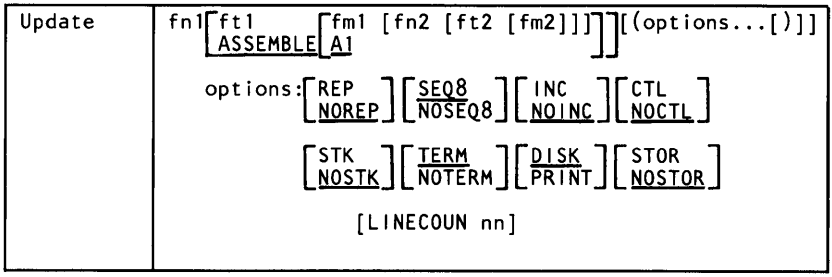
Use the TYPE command to display all or part of a CMS file at the terminal in either EBCDIC or the hexadecimal representation of the EBCDIC code. The format of the TYPE command is:



For a complete description of the TYPE command, refer to the *VM/SP CMS Command and Macro Reference.*

UPDATE

Use the UPDATE command to modify program source files. The UPDATE command accepts a source input file and one or more files containing UPDATE control statements and updated source records; then it creates an updated source output file, an update log file indicating what changes, if any, were made, and an update record file if more than a single update file is applied to the input file. The format of the UPDATE command is:



The LINECOUN option controls the number of lines per print page which the UPDATE command produces. The default value is 55. Except for this option, the VM/PC UPDATE command is functionally equivalent to the VM/SP command.

For a complete description of the UPDATE command, refer to the *VM/SP CMS Command and Macro Reference.*

XEDIT

Use the XEDIT command to invoke the VM/PC editor to create, modify, and manipulate CMS disk files. Once the VM/PC editor has been invoked, you may execute XEDIT subcommands and use the EXEC 2 macro facility.

You can return control to the CMS environment by issuing the XEDIT subcommand FILE, QUIT or QQUIT.

For complete details on XEDIT subcommands and macros, see the publication *VM/SP System Product Editor Command and Macro Reference.*

The format of the XEDIT command is:

XEDIT	[fn [ft [fm]]] [(options...[)]] options: [Width nn] [NOSCreen] [PROFile macroname] [NOPROFil] [NOCLear] options valid only in update mode: [Update] [NOUpdate] [Seq8] [NOSeq8] [Ctl fnl] [NOCtl] [Merge] [Incr nn] [SIDcode string]

When XEDIT is first called, it is loaded into CMS free storage and installed as a nucleus extension. Further calls to XEDIT normally use this copy of XEDIT, thus eliminating the need to reload the XEDIT module.

The VM/PC command XEDIT is functionally equivalent to the VM/SP command XEDIT, except that the default setting for the PF 1 key is the command 'TOP'.

For a complete description of the XEDIT command, refer to the *VM/SP CMS Command and Macro Reference.*

Immediate Commands

You can issue an Immediate command from the terminal only after causing an attention interruption by pressing the PA1 key, then entering the CP command ATTN or pressing the PA1 key a second time. These commands are processed as soon as they are entered.

The HT and RT Immediate commands are also recognized when they are stacked in an EXEC procedure, and the HT Immediate command can be appended to a CMS command

preceded by a logical line end symbol (#). Any program execution in progress is suspended until the Immediate command is processed.

None of the Immediate commands issue responses.

HT: Use the HT command to suppress all terminal output generated by any CMS command or your program that is currently executing. The format of the HT command is:

HT	

Usage Notes:

1. Program execution continues. When the ready message is displayed, normal terminal output resumes. Use the RT command to restore typing or displaying.

2. CMS error messages having a suffix letter of S or T cannot be suppressed.

HX: Use the HX command to stop the execution of any CMS command or program, close any open files or I/O devices, and return to the CMS command environment. The format of the HX command is:

HX	

Usage Notes:

1. HX clears all file definitions made via the FILEDEF command, including those entered with the PERM option.

2. The HX command is executed when the next SVC or I/O interruption occurs; therefore a delay may occur between keying HX and the return to CMS. All terminal output generated before HX is processed is displayed before the command is executed.

3. HX does not clear user storage.

RT: Use the RT command to restore terminal output from an executing CMS command or one of your programs that was previously suppressed by the HT command. The format of the RT command is:

RT	

Usage Note: Program execution continues, and displaying continues from the current point of execution in the program. Any terminal output that is generated after the HT command is issued and up to the time the RT command is issued is lost. Execution continues to normal program completion.

Chapter 8. VM/PC Messages

How To Use This Chapter

This chapter lists messages alphabetically by the first words of the message.

Note: Some messages begin with a replacement value, such as:

```
userid NOT IN REMOTE DIRECTORY
```

When the message is issued, 'userid' is replaced with an actual userid. Since we cannot tell what this userid will be, we cannot alphabetize the message. All messages that start with a replacement value are listed at the beginning of the Message Explanation section.

Identifying Issuing Component

This Chapter contains messages issued by the following components of VM/PC:

- VM/PC CMS - These messages are listed here followed by the identifier (DMSxxxx). The xxxx is replaced by a number, like 001E, if the message has a number. (See Note that follows.)

- VM/PC CP - These messages are listed here followed by the identifier (DMKxxxx). The xxxx is replaced by a number, like 001E.

- CP Termination Messages - These messages are listed here followed by the identifier (DMKxxxT). The xxx is replaced by a number.

- VMPCSERV - These messages are listed here followed by the identifier (SERV).

Note: Not all possible VM/PC CMS Messages are explained here. A complete list of CMS messages is given in

the section "List of All VM/PC CMS Messages" on page 8-26. That list identifies those that are included in the "Message Explanations" section.

Reading a Sample Message Explanation

Here's a sample message:

```
'cuu' replaces 'mode (cuu)'    (DMS724I)
```

The message begins with a "Replacement Value" (cuu is replaced with a number like 101 when the message is issued) so it is not alphabetically listed. The information in parentheses that follows the message, (DMS724I), indicates that CMS issues the message and its number is 724I.

Following these, there is an explanation and an indication of how to recover, if necessary.

Note: VM/PC displays all messages without numbers. If you want message numbers to be displayed with the the messages, issue the command:

```
set emsg on
```

See "SET" on page 7-38 for more information on this command. You would want to issue this command to get message numbers displayed if you were having a problem and were seeking service help.

Message Explanations

Messages are listed alphabetically by their first word, except for the first set of messages grouped under "Replacement Values." These messages begin with a value that changes based upon what you're doing or what your userid is, etc.

If you receive a message that starts with your userid name, or a command name you just issued, or a device number, check the group of messages explained under the "Replacement Values" heading.

REPLACEMENT VALUES

command COMMAND NOT COMPLETED; REMOTE SERVICES NOT AVAILABLE (DMK049E)
The command indicated could not be completed because you have not started the Remote Services Program (VMPCSERV).

'cuu mode' released (DMS726I)
The specified disk has been released since you have asked that another disk be accessed for that mode.

'cuu' already accessed as read/write 'mode' disk (DMS059E)
You are trying to access the specified disk in read-only mode, but you already have it accessed in read/write mode. You cannot have the disk accessed as a read/write disk and as a read/only disk. Release the disk with the RELEASE command and reissue the ACCESS command.

'cuu' replaces 'mode (cuu)' (DMS724I)
The disk being accessed replaces the disk specified in your search order. The disk it replaces is released.

cuu also = 'mode'-disk (DMS725I)
The disk being accessed is also accessed as the 'mode' disk.

dvcadr NOT DEFINED; SPACE NOT AVAILABLE (DMK091E)
The temporary device has not been defined because sufficient blocks could not be found to allocate it.

mode (cuu) is not available from the remote server (DMS803E)
The remote server has been restarted, and no longer has a link for you to the requested disk. Link to the disk again and re-try the ACCESS.

mode (cuu) read-only (DMS723I)
The specified disk is accessed in read-only mode.

name ENVIRONMENT DOES NOT EXIST (DMK044E)
The control program cannot find the environment that you have entered.

name ENVIRONMENT EXCEEDS STORAGE (DMK170E)
The environment you entered is larger that your virtual machine storage size. Either enter another environment that will fit within that storage size or re-define your virtual machine storage size.

'nnn' FB-512 blocks formatted on disk 'mode(cuu)' (DMS732I)
This message tells you how many blocks have been formatted on the specified disk.

'option' and 'option' are conflicting options (DMS066E)
The specified options are mutually exclusive. Reissue the command specifying only one of the options.

'option' option specified twice (DMS065E)
The option was specified more than once in the command line. Reissue the command specifying the option only once.

userid dvcadr NOT LINKED; ALREADY DEFINED (DMK110E)
The device you entered was not linked because your virtual machine configuration already has a device with the same address.

userid dvcadr NOT LINKED; ALREADY LINKED (DMK105E)
The device you entered was not linked because you already have a link to that device.

userid dvcadr NOT LINKED; DOES NOT EXIST (DMK109E)
The device you entered was not linked because the minidisk, although defined in the configuration file, has not been allocated.

userid dvcadr NOT LINKED; NO PASSWORD (DMK113E)
The device you entered was not linked because the device definition in the configuration file does not contain a password for the link mode you requested.

userid dvcadr NOT LINKED; NOT IN CONFIGURATION FILE (DMK107E)
The device you entered was not linked because it could not be found in the configuration file.

userid dvcadr NOT LINKED; PASSWORD INCORRECT (DMK114E)
The device you entered was not linked because the password you entered was incorrect for the link mode you requested.

userid dvcadr NOT LINKED; REMOTE SERVICES NOT AVAILABLE (DMK075E)
You entered a LINK command that was to be processed remotely but the Remote Services Program was not active.

userid dvcadr NOT LINKED; SPACE NOT AVAILABLE (DMK119E)
The device you requested write access to was not linked because it was not previously allocated and sufficient space was not available to perform the allocation.

userid dvcadr NOT LINKED; VOLUME NOT MOUNTED (DMK108E)
The device you entered was not linked because the diskette drive that it was defined on did not have a diskette in it or the drive does not exist.

userid NOT IN REMOTE DIRECTORY (DMK053E)
You entered a userid that could not be found in the VM directory at the remote location.

Special Characters

?CP: command (DMK001E)
You did not enter a valid CP command.

B

BLKS MISSING OR INVALID (DMK024E)
You did not enter the number of blocks or entered it incorrectly. It is incorrect if it is more than four characters or is not a decimal number.

C

Cannot recompute without loss of data. No change (DMS214W)
The number of blocks specified in the FORMAT command cannot contain all the data on the disk. Erase some files, or re-issue the FORMAT command with more blocks specified.

CHARS MISSING OR INVALID (DMK245E)
You did not enter the CHARS operand or it is longer than four characters.

CLASS MISSING OR INVALID (DMK028E)
You did not enter a class value or entered one that is not in the range A through Z or 0 through 9.

CMS subset (DMS)
This is the CMS subset prompt. You enter subset mode by entering the SUBSET command, either explicitly, or implicitly by issuing the XEDIT subcommand CMS. To return to the environment you were in previously (either normal CMS command mode or the program from which you entered subset mode), issue the CMS subset command RETURN.

Comparing 'fn ft fm' with 'fn ft fm' (DMS179I)
This is an informational message to tell you the fileids being compared.

COMPLETION INTERRUPT FOR INACTIVE DEVICE (DMK005T)
A Control Program I/O scheduling error has occurred.

CONFLICTING OPTION - option (DMK013E)
You entered the same option twice in a command, or you entered an option that is incompatible with a previous one, or you entered an option that is incompatible with the system.

COPY MISSING OR INVALID (DMK030E)
You did not enter the number of copies or entered it incorrectly. It is incorrect if it contains nondecimal digits or is greater than 99.

CP ENTERED; DISABLED WAIT PSW 'psw' (DMK450W)
The virtual machine loaded a disabled wait PSW, identified by 'psw'. You must re-IPL the environment.

CP ENTERED; PAGING ERROR (DMK410W)
An irrecoverable data error occurred on the virtual storage file. You must re-IPL the environment.

CP ENTERED; PROGRAM INTERRUPT LOOP (DMK453W)
A program interrupt occurred at the address specified in the virtual machine program new PSW.

CP MESSAGE FILE NOT FOUND (DMK015T)
The control program could not find the control program message file. Run the configurator and display the system configuration to find the filespec for the control program message file.

CP NUCLEUS FILE NOT FOUND (DMK014W)
The control program could not find the control program nucleus file. Run the configurator and display the system configuration to find the filespec for the control program nucleus.

CP PROGRAM CHECK (DMK009T)
A program check has occurred in the 370 portion of the control program.

D

DEV dvcadr DOES NOT EXIST (DMK040E)
You have entered a device address that is not part of your current virtual machine configuration.

DEV dvcadr NOT DEFINED; ALREADY DEFINED (DMK092E)
The device address you entered is already defined in your virtual machine's configuration.

DEV dvcadr NOT DEFINED; MAXIMUM NUMBER OF DEVICES DEFINED (DMK095E)
The device you entered was not defined because you already have defined the maximum number of devices in your virtual machine configuration that the system will support.

A virtual configuration can have one console, one printer, and 26 minidisks.

DEV dvcadr NOT DETACHED; CP OWNED (DMK123E)
The device you entered is owned by the Control Program and cannot be detached. Your console and printer are owned by CP but are shared with your virtual machine.

DEV dvcadr NOT LINKED; MAXIMUM NUMBER OF DEVICES DEFINED (DMK117E)
The device you entered was not linked because you already have defined the maximum number of devices in your virtual machine configuration that the system will support.

A virtual configuration can have one console, one printer, and 26 minidisks.

DEVICE TYPE MISSING OR INVALID (DMK035E)
You did not enter a CP spool device type or entered it incorrectly. Valid device types are PRT and all abbreviations of PRINTER.

Disk 'cuu' is read/only (DMS037E)
The specified command requires a read/write disk. You should access the specified disk in read/write mode and reissue the original command.

Disk 'mode (cuu)' has invalid directory format (DMS112S)
You have requested that a disk which is not in CMS format be accessed. The access has not been performed. Probably the disk has never been formatted. You must format the disk before you can access it.

DISK 'mode<(cuu)>' IS FULL (DMS107S)
There is not enough space on the specified disk to write the file. You may wish to erase some files from the disk, get a larger disk, or expand the size of the disk.

Disk <'mode' | 'cuu'> not accessed (DMS069E)
The specified disk has not been accessed. Reissue the command after accessing the specified disk.

DISTCODE MISSING OR INVALID (DMK032E)
You did not enter a distribution code or it was longer than eight characters.

Drive d not ready. Make drive ready and press enter (DMS807S)
VM/PC has detected that the specified drive is not ready when I/O was to be done to the drive. Make the drive ready, and press the enter key to continue.

DVCADR MISSING OR INVALID (DMK022E)
You have not entered a device address or entered one incorrectly. It is incorrect if it is longer than three characters or contains nonhexadecimal data.

E

Enter disk label: (DMS605R)
Format will write the specified label on the disk being formatted. You can enter a one to six character label for the disk. If you enter less than six characters, the label is left justified and padded with blanks. If you enter a null line, the system displays the message Disk remains unchanged.

Erase * * <fm | *> not allowed (DMS071E)
You cannot erase all files on a disk or all files on all disks. You can enter asterisks for the filename and filetype, but you must specify the filemode letter and number. Correct and reissue the command.

Error 'nn' reading file 'fn ft fm' from disk (DMS104S)
An unrecoverable error occurred while reading the file from disk. "nn" indicates the nature of the error and may be one of the following:

Code	Meaning
1	The specified file was not found.
2	The buffer area is not within user storage limits.
3	A permanent disk read error occurred. This may occur if you link to and access another user's disk, and try to read a file that was re-filed by its owner after you issued the ACCESS command. Reissue the command and try to read the file again.
5	The number of records is less than zero or more than 32768.
7	The fixed/variable flag in the file status table is not F or V.
8	The given storage area was smaller than the actual size of the records read. (This error is

valid if reading the first portion of a large record into a small buffer. It does not cause the function to terminate.)

9	The file is already active for writing and must be closed before it can be read.
11	Only one record can be read for a variable-length file. In this case, the number of records is greater than 1.
12	An unexpected end of file occurred (the record number specified exceeds the number of records in the file).
13	A variable-length file has an invalid displacement in the active file table.
14	An invalid character has been detected in filename.
15	An invalid character has been detected in filetype.
22	Virtual storage capacity has been exceeded.
25	Insufficient virtual storage is available.
26	Requested item number is negative, or item number plus number of items exceeds file system capacity.
27	An attempt was made to update a variable length item with one of a different length.

Error 'nn' writing file 'fn ft fm' on disk (DMS105S)

An unrecoverable error occurred while writing the file on disk. "nn" indicates the nature of the error and may be one of the following:

Code	Meaning
2	The virtual storage address is 0.
4	The first character mode is invalid.
5	The second character mode is invalid.
6	The number of records in the file is too large; it cannot exceed 65,533.
7	An attempt has been made to skip over an unwritten variable length item.
8	The number of bytes was not specified.
9	The file is already active for reading.
10	The maximum number of CMS files has been reached.
11	The fixed/variable flag is not F or V.
12	The disk is not a CMS read/write disk.

13	The disk is full (recoverable error).
14	The number of bytes to be written is not integrally divisible by the number of records to be written.
15	The length of this record is not the same as that of the previous record.
16	The fixed/variable flag is not the same as that of the previous record.
17	A variable-length record is greater than 65535 bytes.
18	The number of records is greater than 1 for variable-length file.
19	The maximum number of data blocks per file (16060) has been reached.
20	An invalid character has been detected in filename.
21	An invalid character has been detected in filetype.
22	Virtual storage capacity has been exceeded.
25	Insufficient virtual storage is available.
26	Requested item number is negative, or item number plus number of items exceeds file system capacity.
27	An attempt was made to update a variable length item with one of a different length.

Error nnnnn when accessing minidisk xxx as mode y (SERV)
As the VMPCSERV program terminates and tries to restore the CMS environment which existed prior to your using the remote server program, an error occurred as your CMS minidisks are being restored. The return code displayed is from the ACCESS command issued by VMPCSERV.

Error nnnnn when accessing minidisk xxx as mode z (SERV)
As the VMPCSERV program terminates and tries to restore the CMS environment which existed prior to your using the remote server program, an error occurred as your CMS minidisks are being restored.

Error in exec file fn, line nnn - message (DMS072E)
The EXEC interpreter has found an error in the specified file, at line "nnn." Correct the EXEC file based on the text of the error message, and re-issue the command.

Error in fn ft fm, line nnn - message (DMS085E)
The EXEC2 interpreter has found an error in the specified file, at line "nnn." Correct the EXEC file based on the text of the error message, and re-issue the command.

Error occurred when allocating a service (SERV)
You should retry VMPCSERV and contact your technical co-ordinator if the program continues to fail.

Error occurred when trying setup/remove I/O interrupt handler, return code: nnn (SERV)
You should retry VMPCSERV and contact your technical co-ordinator if the program continues to fail.

Error occurred when trying to define a service (SERV)
You should retry VMPCSERV and contact your technical co-ordinator if the program continues to fail.

ERROR READING CP MESSAGE FILE (DMK016T)
The control program had an irrecoverable data error while reading the control program message file. Restore the message file from VM/PC installation diskette 2.

ERROR READING VM/PC CONFIGURATION FILE (DMK013T)
The control program had an irrecoverable data error while reading the configuration file. Run the configurator to correct any data problems.

Error when trying to display the logo (SERV)
You should retry VMPCSERV and contact your technical co-ordinator if the program continues to fail.

Execution begins... (DMS740I)
You have requested the execution of a program under CMS.

Export terminated, insufficient disk space available (DMS806E)
The EXPORT of the file failed because there was no space left on the drive to create the PC file. You should make enough space available on the drive to create the file, and retry the operation.

FATAL ERROR FROM REMOTE SERVICES (DMK020T)
The control program had an irrecoverable error while performing remote services.

FCB MISSING OR INVALID (DMK246E)
You did not enter the FCB operand or it is longer than four characters.

File 'fn ft fm' already exists -- specify 'replace' (DMS024E)
The specified file already exists. Reissue the command with the REPLACE option if you wish to overlay the existing copy of the file.

File 'fn ft fm' is not fixed length (DMS034E)
The specified command will only work on fixed format files. You can change the record format of the file using the COPYFILE command with the RECFM F option, and re-issue the original command.

File 'fn ft fm' not found (DMS002W)
The specified file was not found on the accessed disk(s). Either the file does not exist on this disk, or the fileid was misspelled.

File(s) 'fn ft fm' not found. Disk 'mode(cuu)' will not be accessed (DMS060E)
The files requested were not on the specified disk or, if the disk contains no files and you accessed it as an extension, your access fails. Check to see that the fileid specified is valid and reissue the command.

File cannot be exported with the 'EOL' option (DMS808S)
The file to be exported has a logical record length greater than 65535, and cannot be exported with the 'EOL' option. Either use the NOEOL option of EXPORT, or use the CMS COPYFILE command to change the logical record length of the CMS file prior to retrying the EXPORT.

Files do not compare (DMS209W)
The two files you requested to be compared did not match on a line for line, column for column basis.

FLASH MISSING OR INVALID (DMK243E)
You did not enter the FLASH operand or you entered it incorrectly. It is incorrect if it contains nondecimal digits or is larger than 255.

FNAME FTYPE MISSING OR INVALID (DMK029E)
You did not enter a filename or filename and filetype, or you entered it incorrectly. It is incorrect if the filename is larger than 24 characters, or if the filetype is larger than eight characters when the filename is eight characters or less.

FORM NUMBER MISSING OR INVALID (DMK024E)
You did not enter a form number after the FORM keyword or entered one longer than eight characters.

Format will erase all files on disk 'mode(cuu)'. Do you wish to continue? (yes | no): (DMS603R)
You have requested a FORMAT of the cuu disk. This prompt allows you to change your mind, in case you have made a typing error, or decided that the FORMAT should not be done at this time. You should reply yes if you wish the format to continue or the FORMAT will stop without changing the disk.

Formatting disk 'mode' (DMS733I)
The FORMAT command is formatting the specified disk.

H

HEXLOC hexloc EXCEEDS STORAGE (DMK160E)
The hexadecimal location you entered exceeds your virtual machine storage size.

I/O ERROR OCCURRED ON IPL (DMK232E)
An irrecoverable data read error occurred during the IPL of the environment you entered.

I/O REQUESTED AND DEVICE RETURNED BUSY STATUS (DMK004T)
A CP I/O scheduling error has occurred.

I/O REQUESTED AND RDEVBLOK IS FLAGGED AS BUSY (DMK003T)
A CP I/O scheduling error has occurred.

Identical fileids (DMS019E)
The fileids specified are identical. Reissue the command and specify two different fileids.

Insufficient storage to start communications (SERV)
The VMPCSERV program acquires buffers dynamically to use in communications with the local session. You should increase your remote virtual machine size and restart VMPCSERV.

Invalid character 'character' in fileid 'fn ft fm' (DMS062E)
The fileid specified was not valid. Check to see that the fileid specified is valid and reissue the command.

Invalid CMS command (DMS)
The command you entered is not valid.

INVALID DEVICE TYPE - dvcadr (DMK006E)
The device type of the indicated device is incorrect for the function you requested.

Invalid export of a CMS file to a system file (DMS800E)
You have requested an EXPORT of a CMS file directly to a file used by the VM/PC system. EXPORT will not allow this, and the EXPORT is not performed. If you wish to do this, you must EXPORT the file to a temporary PC file, exit VM/PC and rename the exported file to the system fileid which you want.

Invalid filetype 'ft' (DMS032E)
The filetype entered was not valid for the command. Reissue the command with a valid filetype specified.

INVALID HEXDATA - hexdata (DMK005E)
The hexadecimal data you entered either contained nonhexadecimal characters or was longer than eight characters.

INVALID HEXLOC - operand (DMK004E)
The hexadecimal location you entered either contained nonhexadecimal characters or was longer than six characters.

Invalid mode 'mode' (DMS048E)
The mode which has been specified on the command is invalid. This may be due to misspelling or improper command format. Reissue the command with a correct filemode.

Invalid mode change (DMS051E)
The filemode letter specified for the old fileid is not the same as that specified for the new fileid. Reissue the command with the correct filemodes.

INVALID OPERAND - operand (DMK002E)
One of the command operands you entered was not correct.

INVALID OPTION - option (DMK003E)
One of the options you entered was not correct or should not be used with the command variation you used. You might have given too short an abbreviation for the option.

Invalid option 'option' (DMS003E)
The specified option is invalid. It may have been misspelled or an improper abbreviation for the option may have been specified.

Invalid parameter 'parameter' (DMS070E)
An invalid operand was specified in the command line. Correct and reissue the command.

Invalid parameter 'parameter' for 'function' function (DMS026E)
The data specified for the given function is misspelled, missing or invalid. Reissue the command with valid data for the function.

Invalid parameter 'parameter' in the option 'option' field (DMS029E)
The data entered following the specified option was invalid. Reissue the command with valid data after the option.

Invalid PC fileid 'd:filename.ext' specified (DMS804E)
The PC fileid specified is invalid. Retry the command with a valid fileid.

INVALID PSW - psw (DMK012E)
The PSW (program status word) you entered in the STORE command has the extended control mode bit on.

INVALID RANGE - range (DMK009E)
The range of addresses or registers you entered is incorrect. For instance, the end address may be smaller than the starting address.

INVALID REGISTER - register (DMK010E)
The register you entered was not in the range 0 through 15 or hexadecimal 0 through F, or was not 0, 2, 4, or 6 for a floating-point register.

INVALID SPOOLID - spoolid (DMK008E)
The spoolid you entered was not a decimal number or was longer than four characters.

Invalid subset command (DMS)
The command you entered cannot be executed from CMS subset mode due to restrictions in where the command must execute. Commands which execute in the user area are not permitted to be executed from CMS subset mode.

Invalid subsystem was specified on ALLOCATE (SERV)
You should retry VMPCSERV and contact your technical co-ordinator if the program continues to fail.

Invalid subsystem was specified on DEALLOCATE (SERV)
You should retry VMPCSERV and contact your technical co-ordinator if the program continues to fail.

IRRECOVERABLE MACHINE CHECK (DMK008T)
The 370 processor has encountered an error for which there is no recovery.

L

Line I/O error nnnnn during read operation. Sense data is xx. (SERV)
You should retry VMPCSERV and contact your technical co-ordinator if the program continues to fail.

Line I/O error nnnnn during write operation. Sense data is xx. (SERV)
You should retry VMPCSERV and contact your technical co-ordinator if the program continues to fail.

LINECT MISSING OR INVALID (DMK034E)
You did not enter the number of lines per page or you entered it incorrectly. It is incorrect if it contains nondecimal digits or is greater than 255.

M

MODIFY MISSING OR INVALID (DMK253E)
You did not enter the MODIFY operand or it is longer than four characters.

N

No 'option' specified (DMS005E)
The specified option was entered in an incomplete form.

NO CFRB AVAILABLE (DMK011T)
An error has occurred in the management of console function requests.

No device specified (DMS028E)
The command requires that you specify a device address. Reissue the command with a valid device address.

No fileid<(s)> specified (DMS042E)
At least one fileid must be specified for the command issued. Reissue the command, specifying at least one fileid.

No filemode specified (DMS223E)
A filemode was not specified in the command line. Reissue the command specifying a filemode.

No filename specified (DMS001E)
The command you entered requires that you specify at least one filename. Reissue the command and specify the filename(s).

No files loaded (DMS040E)
You have not previously issued a LOADMOD or LOAD command to load an executable module. You should load the required files, and then reissue the original command.

No filetype specified (DMS023E)
The command you entered requires that you specify both filename and filetype. Reissue the command and specify the filename and filetype.

NO IOB AVAILABLE (DMK006T)
A Control Program I/O scheduling error has occurred.

NO PAGE FRAMES AVAILABLE (DMK018T)
A virtual storage management error has occurred.

NO SAVE AREA AVAILABLE (DMK001T)
A Control Program subroutine call has been issued and no internal save area is available.

No user defined FILEDEFs in effect (DMS324I)
No current FILEDEFs exist. Reissue the command specifying a filemode.

NO USER DEFINITIONS IN CONFIGURATION FILE; RUN CONFIGURATOR (DMK017T)
The control program found no user information in the configuration file. Run the configurator to define the user(s) of the system.

O

OPERAND MISSING OR INVALID (DMK026E)
You did not enter a required operand or entered it incorrectly.

OS simulation is not available (DMS098S)
An OS SVC was issued, and the CMS OS simulation support was not available at that time. Issue the command SET OSSIM ON, and re-issue the command.

P

PAGING I/O ERROR ON CP NUCLEUS (DMK019T)
The control program had an irrecoverable data error while reading the control program nucleus file. Restore the nucleus file from VM/PC installation diskette 2.

R

Ready; (DMS)
This is the CMS prompt, and tells you that CMS is waiting for a command. It may also appear as Ready(rc); where rc is the return code of the previous command you entered.

Remote disk mode (cuu) released (DMS802E)
The communication with the remote server has been stopped or restarted. The remote minidisks which you had accessed are released automatically. To re-establish the connection to remote minidisks you should restart the remote server if it is not running, and issue the appropriate LINK and ACCESS commands to establish the connection to the remote minidisks again.

Server ended due to severe I/O errors (SERV)
You should retry VMPCSERV and contact your technical co-ordinator if the program continues to fail.

SPOOL ERROR; FILE CLOSED WITH HOLD OPTION (DMK428E)
An error occurred while writing spool information onto disk. A spool file is created, if possible, with any existing spool data, and that file is placed in HOLD status.

SPOOL ERROR; VOLUME NOT MOUNTED (DMK430E)
You do not have a diskette in the drive to which spool files have been directed.

SPOOL SPACE FULL; FILE CLOSED WITH HOLD OPTION (DMK427E)
The disk or disk directory is full, or the maximum number of spool files have already been used. A spool file is created, if possible, with any existing spool data, and that file is placed in HOLD status.

SPOOLID nnnn DOES NOT EXIST (DMK042E)
You have entered a spoolid that does not exist or is currently being processed.

SPOOLID MISSING OR INVALID (DMK027E)
You did not enter a spoolid or entered it incorrectly. It is incorrect if it is longer than four characters or contains nondecimal digits.

STORAGE EXCEEDS ALLOWED MAXIMUM (DMK094E)
The storage size you entered exceeds the system allowed maximum of 4 megabytes.

STORAGE MISSING OR INVALID (DMK025E)
You did not enter the amount of storage or entered it incorrectly. It is incorrect if it contains nondecimal digits or does not end with the letter K or M.

STORE REGISTER EXCEEDS MAXIMUM VALUE (DMK163E)
You entered more data to be consecutively stored in either general or floating-point registers than there are registers for that data.

T

TERMINAL ASYNCHRONOUS INTERRUPT WAS NOT ATTENTION (DMK002T)
An asynchronous interrupt from the simulated 3277 for the Local Session was presented and the status associated with that interrupt was not attention.

The following names are undefined: (DMS201W)
A list of names of unresolved external references is displayed. You may obtain access to the specified files by issuing the appropriate INCLUDE command, or ignore the warning message if the modules are not needed for correct execution of your program.

The remote server has been restarted. Connection to remote disks has been lost. (DMS805E)
VM/PC detected that the remote server has been stopped and restarted. You must re-establish the connection to remote minidisks by issuing the appropriate LINK and ACCESS commands for those minidisks.

The remote server is not available. Connection to remote disks has been lost. (DMS801E)
The communication with the remote server program has been stopped. You must re-start the remote VMPCSERV program to re-establish communications with the remote server, and issue the appropriate LINK and ACCESS commands to establish your remote minidisks.

The xx disk does not exist. Disk yyy accessed as mode z. (SERV)
As the VMPCSERV program terminates and tries to restore the CMS environment which existed prior to your using the remote server program, an error occurred as

your CMS minidisks are being restored. The disk to be accessed was originally accessed as an extension of another disk. The extension mode disk was not accessed when the VMPCSERV program was invoked, and therefore, the original environment cannot be recreated for you.

TRACE NOT STARTED (DMK180W)
You entered the TRACE OFF command and trace was not in effect.

U

Unknown CP/CMS command (DMS)
The command you entered is not a valid CP or CMS command. You might have typed the command incorrectly or have given too short an abbreviation for the command.

USERID MISSING OR INVALID (DMK020E)
You have either failed to enter a userid in a command or entered an invalid userid, such as one longer that eight characters.

V

Virtual storage capacity exceeded (DMS109S)
There is not enough storage in your virtual machine to successfully complete execution of the command. You may wish to define a larger virtual storage, restart CMS and reexecute the command.

VIRTUAL STORAGE TRANSLATION EXCEPTION (DMK010T)
A virtual storage management error has occurred.

VM/PC CONFIGURATION FILE NOT FOUND (DMK012T)
The control program could not find the configuration file. The configuration file filespec is CONFIG.1VM and may reside on either fixed disk C or D.

You must have at least 26 device addresses available between 201 and 5FF to start the remote server. (SERV)
The VMPCSERV program may need up to 26 different virtual devices available in order to link to remote minidisks for you. As the program starts, it checks to see if it will have enough available, and terminates if this condition is not satisfied. You should make at least 26 virtual addresses between 201 and 5FF available before restarting VMPCSERV.

List of All VM/PC CMS Messages

The following is a list of possible VM/PC CMS Messages. Those preceded by a bullet (•) are included in the section "Message Explanations" on page 8-3. For further information on the other messages, see the IBM publication *VM/SP System Messages and Codes*, order number SQ19-6204.

Note: The first 5 messages are unnumbered; the others are identified by the last 4 characters of their 10-character identifier. For example, message DMSxxx001E is shown as 001E.

- Ready;
- Unknown CP/CMS command
- Invalid CMS command
- CMS subset
- Invalid subset command
- 001E NO FILENAME SPECIFIED
  002E (INPUT | OVERLAY) (FILE(S)) ('fn (ft (fm))') NOT FOUND
  002I FILE 'fn' TXTLIB NOT FOUND
- 002W FILE 'fn ft (fm)' NOT FOUND
- 003E INVALID OPTION 'option'
- 005E NO 'option' SPECIFIED
  006E NO READ/WRITE ('A') DISK ACCESSED (FOR 'fn ft')
  007E FILE 'fn ft fm' (IS) NOT FIXED, 80 CHAR. RECORDS
  008E DEVICE 'cuu' (INVALID OR NONEXISTENT | UNSUPPORTED DEVICE TYPE)
  009E COLUMN 'col' EXCEEDS RECORD LENGTH
  010E PREMATURE EOF ON FILE 'fn ft (fm)'
  010W PREMATURE EOF ON FILE 'fn ft fm' -- SEQ NUMBER '.....' NOT FOUND
  011E CONFLICTING FILE FORMATS
  013E MEMBER 'name' NOT FOUND IN LIBRARY ('fn ft fm' | 'libname')
  013W MEMBER 'name' NOT FOUND IN LIBRARY 'fn ft fm'
  015E (UNKNOWN (CP/CMS | CMS | CP) | INVALID (CMS | SUBSET)) COMMAND

017E INVALID DEVICE ADDRESS 'cuu'
018E NO LOAD MAP AVAILABLE
- 019E IDENTICAL FILEIDS
021E ENTRY POINT 'name' NOT FOUND
- 023E NO FILETYPE SPECIFIED
- 024E FILE 'fn ft fm' ALREADY EXISTS
(-- SPECIFY 'REPLACE' (option))
- 026E INVALID PARAMETER 'parameter'
FOR 'function' FUNCTION
- 028E NO DEVICE SPECIFIED
- 029E INVALID PARAMETER 'parameter'
IN THE OPTION 'option' FIELD
030E FILE 'fn ft fm' ALREADY ACTIVE
031E LOADER TABLES CANNOT BE MODIFIED
- 032E INVALID FILETYPE 'ft'
033E FILE 'fn ft fm' IS NOT A LIBRARY
- 034E FILE 'fn ft fm' IS NOT FIXED LENGTH
036E OPEN ERROR CODE 'nn' ON 'ddname'
- 037E DISK 'cuu' IS REAL/ONLY
038E FILEID CONFLICT FOR DDNAME
'(ASSEMBLE | SYSIN)'
039E NO ENTRIES IN LIBRARY 'fn ft fm'
- 040E NO FILES LOADED
041E INPUT AND OUTPUT FILES ARE THE SAME
- 042E NO FILEID(S) SPECIFIED
044E RECORD LENGTH EXCEEDS ALLOWABLE
MAXIMUM
046E NO LIBRARY NAME SPECIFIED
047E NO FUNCTION SPECIFIED
- 048E INVALID MODE 'mode'
049E INVALID LINE NUMBER 'line number'
050E PARAMETER MISSING AFTER
(DDNAME | SYSxxx | function)
- 051E INVALID MODE CHANGE
052E MORE THAN 100 CHARS OF OPTIONS
SPECIFIED
053E INVALID SORT FIELD PAIR DEFINED
054E INCOMPLETE FILEID SPECIFIED
055E NO ENTRY POINT DEFINED
056E FILE 'fn ft (fm)' CONTAINS INVALID
(NAME | ALIAS | ENTRY | ESD)
RECORD FORMATS
056W FILE 'fn ft fm' CONTAINS INVALID
(NAME | ALIAS | ENTRY | ESD)

RECORD FORMATS

- 059E 'cuu' ALREADY ACCESSED AS READ/WRITE 'mode' DISK
- 060E FILE(S) 'fn (ft (fm))' NOT FOUND. DISK 'mode (cuu)' WILL NOT BE ACCESSED
- 062E INVALID CHARACTER 'character' IN FILEID 'fn ft fm'

063E NO (SORT | TRANSLATION | SPECIFICATION) LIST (ENTERED | GIVEN)

064E INVALID (TRANSLATE) SPECIFICATION AT OR NEAR 'list'

- 065E 'option' OPTION SPECIFIED TWICE
- 066E 'option' AND 'option' ARE CONFLICTING OPTIONS

067E COMBINED INPUT FILES ILLEGAL WITH PACK OR UNPACK OPTIONS

068E INPUT FILE 'fn ft fm' NOT IN PACKED FORMAT

- 069E DISK <'mode' | 'cuu'> NOT ACCESSED

069I (OUTPUT) DISK 'mode' NOT ACCESSED

- 070E INVALID PARAMETER 'parameter'
- 071E ERASE * * (fn | *) NOT ALLOWED
- 072E ERROR IN EXEC FILE fn, LINE nnn - message

073E UNABLE TO OPEN FILE ddname

074E ERROR (RE)SETTING AUXILIARY DIRECTORY

075E (DEVICE) devicename (INVALID | ILLEGAL) FOR (INPUT | OUTPUT)

084E INVALID USE OF 'FROM' AND 'TO' OPTIONS

- 085E ERROR IN fn ft fm, LINE nnn - message
- 098S OS SIMULATION NOT AVAILABLE
- 098W OS SIMULATION NOT AVAILABLE

101S 'SPECS' TEMP STRING STORAGE EXHAUSTED AT '.....'

102S TOO MANY FILEIDS

103S NUMBER OF SPECS EXCEEDS MAX 20

- 104S ERROR 'nn' READING FILE 'fn ft fm' FROM DISK
- 105S ERROR 'nn' WRITING FILE 'fn ft (fm)' ON DISK
- 107S DISK 'mode ((cuu))' IS FULL

108S MORE THAN 8 LIBRARIES SPECIFIED

- 109S VIRTUAL STORAGE CAPACITY EXCEEDED
- 109T VIRTUAL STORAGE CAPACITY EXCEEDED
- 112S Disk 'mode (cuu)' has invalid directory format

113S (mode | DEVICE | DISK | PRINTER) (cuu) NOT ATTACHED

116S LOADER TABLE OVERFLOW

117S ERROR WRITING TO DISPLAY TERMINAL

119S UNSUPPORTED FORM OF ('macro' | 'parm') (MACRO | SVC)

120S (IN | OUT)PUT ERROR CODE 'nn' ON 'ddname'

121S UNSUPPORTED SVC svc (HEX xx) CALLED FROM addr

122S ERROR IN CALL TO routine FROM addr, ERROR CODE nnn (HEX xxxxxx)

123S ERROR 'nn' (PRINTING | PUNCHING) FILE 'fn ft fm'

125S PERMANENT UNIT CHECK ON DISK 'mode(cuu)'

126S ERROR (READING | WRITING) LABEL ON DISK 'mode(cuu)'

132S FILE 'fn ft fm' TOO LARGE

133S INVALID GETMAIN OR FREEMAIN SPECIFICATION

134S UNSUPPORTED SVC 203 CODE nn CALLED FROM addr.

135S MAXIMUM SVC DEPTH 'nnnn' HAS BEEN EXCEEDED WITH A CALL AT addr

136T SVC CALL FROM addr ILLEGALLY RE-ENTERS INTSVC. RE-IPL CMS

137S ERROR nn ON STATE FOR 'fn ft fm'

137T CALL TO routine FROM addr DESTROYED SYSTEM SAVE AREA. RE-IPL CMS

138T 'DMSKEY' CALL FROM addr OVERFLOWS KEY STACK, WITH MAX DEPTH 7

139T 'DMSKEY RESET' FROM addr UNDERFLOWS KEY STACK

140T routine ROUTINE CALLED FROM addr DID DMSKEY WITH NO RESET.

141T OPERATION EXCEPTION OCCURRED AT xxxxxx IN ROUTINE xxxxxxxx.

142T exception EXCEPTION OCCURRED AT addr in ROUTINE routine DURING 'SPIE' EXIT ROUTINE

143T exception EXCEPTION OCCURRED AT addr in

SYSTEM ROUTINE called. IPL CMS again
144T exception EXCEPTION OCCURRED AT addr IN ROUTINE called WHILE 'UFDBUSY' = xx. IPL CMS AGAIN.
145S INTERVENTION REQUIRED ON PRINTER
148T SYSTEM ABEND xxx CALLED FROM 'addr'
149T nnn (HEX xxx) DOUBLEWORDS OF SYSTEM STORAGE HAVE BEEN DESTROYED. RE-IPL CMS
150W nnn (HEX xxx) DOUBLEWORDS OF SYSTEM STORAGE WERE NOT RECOVERED
152T SYSTEM ABEND xxx CALLED FROM addr WHILE 'UFBUSY' = xx. RE-IPL CMS
153W 'HX' DURING ABEND PROCESSING WAS IGNORED
154T SAVE AREA FOR SVC CALL FROM addr CANNOT BE ALLOCATED
155T USER ABEND nnnn CALLED FROM addr
156E (FROM | RECORD) 'nnn' NOT FOUND - FILE 'fn ft fm' HAS ONLY 'nnn RECORDS
157E LABEL 'label' NOT FOUND IN FILE 'fn ft fm'
157S MACLIB LIMIT EXCEEDED (, LAST MEMBER ADDED WAS 'membername')
159T INSUFFICIENT STORAGE AVAILABLE TO SATISFY DMSFREE REQUEST FROM addr
160T INVALID DMSFREE CALL FROM addr
161T INVALID DMSFRET CALL FROM addr, ERROR NUMBER nn
162T VITAL FREE STORAGE POINTERS DESTROYED (INTERNAL ERROR CODE nn), RE-IPL CMS
163T (LOW | HIGH)-CORE USER CHAIN HAS BEEN DESTROYED (INTERNAL ERROR CODE nn)
164T (LOW | HIGH)-CORE USER NUCLEUS STORAGE POINTERS DESTROYED (INTERNAL CODE nn). RE-IPL CMS
165T CHAIN HEADER AT addr: xxxxxx xxxxxx xxxxxx xxxxxx
166T UNEXPECTED ERROR IN FREE STORAGE MANAGEMENT ROUTINE (INTERNAL ERROR CODE nn). RE-IPL CMS
167S PREVIOUS MACLIB FUNCTION NOT FINISHED

167T FREE STORAGE MANAGEMENT ERROR, INTERNAL ERROR CODE nn
168S PSEUDO REGISTER TABLE OVERFLOW
169S ESDID TABLE OVERFLOW
171T PERMANENT CONSOLE ERROR. RE-IPL CMS
172E TO LABEL 'label' (EQUALS | IS AN INITIAL SUBSTRING OF) FRLABEL 'label'
173E NO RECORDS WERE COPIED TO OUTPUT FILE 'fn ft fm'
174W SEQUENCE ERROR INTRODUCED IN OUTPUT FILE: '.....' TO '.....'
175E INVALID EXEC COMMAND
176W SEQUENCING OVERFLOW FOLLOWING SEQ NUMBER '.....'
177I WARNING MESSAGES ISSUED (SEVERITY = nn). ('REP' OPTION IGNORED.)
178I UPDATING 'fn ft fm' APPLYING 'fn ft fm' ((APPLYING 'fn ft fm'))
179E MISSING OR DUPLICATE 'MACS' CARD IN CONTROL FILE 'fn ft fm'
- 179I COMPARING 'fn ft fm' WITH 'fn ft fm'
180W MISSING PTF FILE 'fn ft fm'
181E NO UPDATE FILES WERE FOUND
182W SEQUENCE INCREMENT IS ZERO
183E INVALID (CONTROL | AUX) FILE CONTROL CARD
184W './S' NOR FIRST CARD IN UPDATE FILE -- IGNORED
185W (INVALID | NON NUMERIC) CHARACTER IN SEQUENCE FIELD '.....'
186W SEQUENCE NUMBER ('.....') NOT FOUND
187E OPTION 'STK' INVALID WITHOUT 'CTL'
- 201W THE FOLLOWING NAMES ARE UNDEFINED:
202W DUPLICATE IDENTIFIER 'identifier'
203W "SET LOCATION COUNTER" NAME 'name' UNDEFINED
206W PSEUDO REGISTER ALIGNMENT ERROR
207W INVALID UPDATE FILE CONTROL CARD
- 209W FILES DO NOT COMPARE
210E (LIBRARY | FILE) 'fn ft' IS ON A READ/ONLY DISK.

210W INPUT FILE SEQUENCE ERROR:
'.....' TO '.....'
211E COLUMN FIELDS OUT OF SEQUENCE
212E MAXIMUM NUMBER OF RECORDS EXCEEDED
213W LIBRARY 'fn ft fm' NOT CREATED
- 214W CANNOT RECOMPUTE WITHOUT LOSS OF DATA. NO CHANGE
216E INSUFFICIENT BLOCKS ON DISK TO SUPPORT CMS DISK STRUCTURE
220R ENTER DATA SET NAME:
221E INVALID DATA SET NAME
- 223E NO FILEMODE SPECIFIED
224E fileid ALREADY IN USE
253E FILE fn ft fm CANNOT BE HANDLED WITH SUPPLIED PARAMETER LIST
255T INSUFFICIENT STORAGE FOR EXEC INTERPRETER
299E INSUFFICIENT STORAGE TO COMPLETE UPDATE
300E INSUFFICIENT STORAGE TO BEGIN UPDATE
304I UPDATE PROCESSING WILL BE DONE USING DISK
- 324I NO USER DEFINED FILEDEFs IN EFFECT
495E THE 'module1' MODULE CAN ONLY BE EXECUTED BY THE 'module2' MODULE, IT CANNOT BE EXECUTED BY ITSELF
500E UNABLE TO UNPACK FILE 'fn ft fm'
501I nn LINES DELETED
502E nn LINES TO RECOVER
502W NO LINE TO RECOVER
503E TRUNCATED
504E nn LINE(S) TRUNCATED
505E NOT EXECUTED: THE TARGET LINE (nn) IS WITHIN THE LINES TO (MOVE | COPY)
506I nn LINES (MOVED | COPIED)
507E NO PRESERVED DATA TO RESTORE
508E 'load' MUST BE THE FIRST SUBCOMMAND IN THE PROFILE
509W USE GET AND/OR PUT(D) TO MOVE LINES IN UPDATE MODE
510I AUTOSAVED AS 'fn ft fm'.
511E STRING2 CONTAINS MORE ARBITRARY

CHARACTERS THAN STRING1
512E INVALID SUBSET COMMAND
513E UNKNOWN CP/CMS COMMAND
514E RETURN CODE 'nn' FROM 'command'
515E RECFM MUST BE (F | V | FP | VP)
516E LRECL TOO LARGE FOR V-FORMAT FILE
517I nn OCCURRENCE(S) CHANGE ON nn LINE(S)
518E nn OCCURRENCE(S) CHANGED ON nn LINE(S); nn LINE(S) TRUNCATED.
519E LCREL MUST BE LOWER THAN WIDTH (nn)
520E INVALID OPERAND : operand
521E INVALID LINE NUMBER
522I nn OCCURRENCES
523I TYPEWRITER MODE
524W NONDISP CHARACTER RESET TO BLANK
525E INVALID PFKEY NUMBER
526E OPTION 'option' VALID IN DISPLAY MODE ONLY
527E INVALID COLUMN NUMBER
528E INVALID RANGE : TARGET2 (LINE nn) PRECEDES TARGET1 (LINE nn)
529E SUBCOMMAND IS ONLY VALID IN (DISPLAY | EDITING) MODE
530I nn FILE(S) IN STORAGE.
531E DISK IS FULL. SET NEW FILEMODE OR CLEAR SOME DISK SPACE
532E DISK IS FULL; AUTOSAVE FAILED
533E LINE 'linenum' IS NOT RESERVED
534E TOO MANY LOGICAL SCREENS DEFINED
535E INVALID PARMS FOR RENUM
536E NUMBER OF LINES EXCEEDS PHYSICAL SCREEN SIZE
537E EACH LOGICAL SCREEN MUST CONTAIN AT LEAST 5 LINES
538E NO NAME DEFINED
539E NAMED LINE NOT FOUND
540E NAME ALREADY DEFINED ON LINE 'nn'
541E INVALID NAME
542E NO SUCH SUBCOMMAND : name
543E INVALID NUMBER : xxxxxxxx
544E INVALID HEX DATA : xxxxxxxx
545E MISSING OPERAND(S)
546E TARGET NOT FOUND

547E SYNONYM DEFINITION INCOMPLETE
548E INVALID SYNONYM OPERAND : operand
549E SYNONYM ABBREVIATION TOO LARGE
550E TOO MANY OPERANDS IN SYNONYM DEFINITION
551I STRING 'string' FOUND. (---PFnn SET FOR SELECTIVE CHANGE.)
552E NO SYNONYM CURRENTLY DEFINED
553I EDITING FILE: fn ft fm
554E NO STORAGE AVAILABLE
555E FILE 'fn ft fm' ALREADY IN STORAGE
555W FINE 'fn ft fm' ALREADY IN STORAGE
556S UNDEFINED MESSAGE (DMSXER CALLED AT hex loc.)
557S NO MORE STORAGE TO INSERT LINES
558E WRONG FILE FORMAT FOR SERIALIZATION
559W WARNING FILE IS EMPTY
560W NOT ENOUGH SPACE FOR SERIALIZATION BETWEEN TRUNC AND LRECL
561E CURSOR IS NOT ON A VALID DATA FIELD
562E NO LINE(S) SAVED BY PUT(D) SUBCOMMAND
563W RECORDS TRUNCATED
564W EOF REACHED
565W EOF REACHED: RECORDS TRUNCATED
569E NO 'CHANGE' OR 'CLOCATE' SUBCOMMAND SPECIFIED
571I CREATING NEW FILE:
572E TERMINAL ERROR. FORCED CASE UPPERCASE.
573I INPUT MODE:
574E CHANGE NOT VALID (WITH CLOCATE | AFTER CURSOR MOVEMENT)
575E INVALID (JOIN | SPLIT | TABS | VERIFY | ZONE) COLUMNS DEFINED
576E (TOTAL VERIFY WIDTH EXCEEDS SCREEN SIZE (nn) | TOTAL OFFSET EXCEEDS LRECL (nn)
577E FILE HAS BEEN CHANGED. USE QQUIT TO QUIT ANYWAY
579R RECORDS WILL BE TRUNCATED TO nn. CONTINUE (YES/NO)?:
580W PUT NOT EXECUTED

582S EDITOR ABEND
583I EOF:
584I TOP:
585E NO LINE(S) CHANGED
586E NOT FOUND (ON SCREEN)
587I XEDIT:
588E PREFIX SUBCOMMAND WAITING...
590E DATA SET TOO LARGE
597E UNABLE TO MERGE UPDATES
CONTAINING './S' CARDS
598S UNABLE TO BUILD UPDATE FILE :
INTERNAL LIST DESTROYED
599S UNABLE TO BUILD UPDATE FILE :
SERIALIZATION DESTROYED
601R ENTER SPECIFICATION LIST:
602R ENTER TRANSLATION LIST:
- 603R FORMAT WILL ERASE ALL FILES ON DISK 'mode(cuu)'. DO YOU WANT TO CONTINUE? (YES | NO):
604R ENTER SORT FIELDS:
- 605R ENTER DISK LABEL:
616W 'name' DOES NOT EXIST
617E ERROR CODE nn FROM DMSFRET WHILE UNLOADING 'module name' MODULE
618E NUCEXT FAILED
621E BAD PLIST:
622E INSUFFICIENT FREE STORAGE (message)
623S MODULE CANNOT BE LOADED AT LOCATION 'location' BECAUSE IT WOULD OVERLAY THE CMS NUCLEUS
624I NO NUCLEUS EXTENSIONS ARE LOADED
624W NO NUCLEUS EXTENSIONS ARE LOADED
628E INVALID GLOBALV FUNCTION 'function'
631E SET/SETLP/SETLS CAN ONLY BE EXECUTED FROM AN EXEC2 EXEC OR AS A CMS COMMAND
632E I/O ERROR IN EXECIO: RC=nn FROM 'command' COMMAND
637E MISSING (VALUE | 'value') FOR THE ('option' OPTION | 'operand' OPERAND
649E EXTRANEOUS PARAMETER 'parameter'
704I INVALID CLEAR REQUEST
705I DISK REMAINS UNCHANGED
711I NO SYSTEM SYNONYMS IN EFFECT

721I COPY 'fn ft fm' (TO | APPEND | OVLY)
'fn ft fm' (OLD | NEW) FILE

- 723I mode (cuu) READ-ONLY
- 724I 'cuu' REPLACES 'mode (cuu)'
- 725I cuu ALSO = 'mode'-DISK
- 726I 'cuu mode' RELEASED

728I DEBUG ENTERED (EXTERNAL
INT | BREAKPOINT nn AT xxxxxx)

- 732I 'nnn' FB-512 BLOCKS FORMATTED ON DISK 'mode(cuu)'
- 733I FORMATTING DISK 'mode'
- 740I EXECUTION BEGINS ...
- 800E INVALID EXPORT OF A CMS FILE TO A SYSTEM FILE
- 801E The remote server is not available. Connection to remote disks has been lost
- 802E REMOTE DISK mode (cuu) RELEASED
- 803E mode (cuu) IS NOT AVAILABLE FROM THE REMOTE SERVER
- 804E INVALID PC FILEID 'd:filename.ext' SPECIFIED
- 805E The remote server has been restarted. Connection to remote disks has been lost
- 806E Export terminated, insufficient disk space available
- 807S Drive d not ready. Make drive ready and press enter
- 808S File cannot be exported with the 'EOL' option

901T UNEXPECTED ERROR AT 'addr': PLIST 'plist fn ft fm' AT addr, BASE: addr, RC nn

903T IMPOSSIBLE PHASE CODE 'xx'

904T UNEXPECTED UNPACK ERROR AT 'addr', BASE 'addr'

907T I/O ERROR ON FILE 'fn ft fm'

908T FILE SYSTEM ERROR DETECTED. VIRTUAL ADDR 'cuu'. REASON CODE 'nn'

909T PERM I/O ERROR ON 'cuu'. CSW cccccccc cccccccc SENSE ssssssss ssssssss ssssssss ssssssss ssssssss ssssssss

951E INVALID SVC nn. GAM/SP NOT INSTALLED

Appendix A. CP/CMS Interface

Privileged Operation Simulation

The following privileged operations are supported:

- Set Storage Key - SSK
- Insert Storage Key - ISK
- Set System Mask - SSM
- Load PSW - LPSW
- Test I/O - TIO
- Store CPU ID - STIDP
- Set Clock - SCK
- Set PSW Key from Address - SPKA
- Insert PSW Key - IPK
- Set CPU Timer - SPT
- Store CPU Timer - STPT

Privileged operations not listed above are not supported and, if executed, will cause an operation program check to be reflected to the Virtual Machine. Operations listed above but not described below, function as described in the System/370 Principles of Operations.

For privileged operations which modify the PSW, only the BC format of the PSW is supported.

Set Storage Key - SSK

The hardware does not support storage protection, but storage keys can be set and may be retrieved by the ISK instruction. The Fetch Protect and Dynamic Address Translation bits (4-6) are ignored.

Insert Storage Key - ISK

The requested storage key is returned as set by the SSK instruction, but the Fetch Protect and Dynamic Address Translation bits (4-6) will always be zero.

The storage key bits in the PSW, 8-11, (for LPSW and interrupt PSW swaps) will be maintained although no storage protection is performed.

Test I/O - TIO

The TIO instruction may be used to clear pending I/O interrupts. However, since most of the I/O operations are performed synchronously (without I/O interrupts) via the Diagnose Interface, most of the time this instruction is executed it will return condition code 0 (I/O device available) or 3 (I/O device not available).

The only virtual machine I/O interrupts that are presented are for asynchronous terminal attentions and completion interrupts for Diagnose 58.

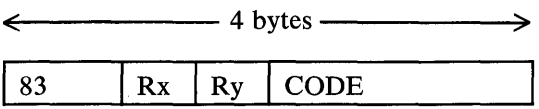
Set Clock - SCK

The TOD Clock is managed by the PC DOS and is always secure. As a result the SCK instruction always returns condition code 1.

Diagnose Instruction in VM/PC

The DIAGNOSE instruction cannot be used in a virtual machine for its normal function. If a virtual machine attempts to execute a DIAGNOSE instruction, a program interrupt returns control to CP. Since a DIAGNOSE instruction issued in a virtual machine results only in returning control to CP and not in performing normal DIAGNOSE functions, the instruction is used for communication between a virtual machine and CP.

The machine language format of DIAGNOSE is:



Where:

83
is X'83' and interpreted by the assembler as the DIAGNOSE instruction.

Note: There is no assembler mnemonic for DIAGNOSE.

Rx, Ry
are general purpose registers that contain operand storage addresses or return codes passed to the DIAGNOSE interface.

CODE
is a two-byte hexadecimal value that CP uses to determine what DIAGNOSE function to perform. The codes defined for the VM/PC user are described in this section.

Because DIAGNOSE operates differently in a virtual machine than it does in a real machine, a program should determine that it is operating in a virtual machine before issuing a DIAGNOSE instruction, and prevent execution of a DIAGNOSE when in a real machine. The Store Processor ID (STIDP) instruction provides a program with information about the processor in which it is executing, including the

processor version number. If STIDP is issued from a virtual machine, the version number will be X'FF' in the first byte of the CPUID field.

A virtual machine issuing a diagnose instruction should run with interrupts disabled. This prevents loss of status information pertaining to the diagnose operation such as condition codes and sense data.

Note: A DIAGNOSE instruction with invalid parameters may in some cases result in a specification exception.

DIAGNOSE Code X'00' -- Store Extended-Identification Code

Execution of DIAGNOSE code X'00' allows a virtual machine to examine the VM/PC extended-identification code.

The register specified as Rx contains the double-word aligned virtual storage address where the VM/PC extended-identification code is to be stored. The Ry register contains the number of bytes to be stored entered as an unsigned binary number (maximum of 32 bytes).

The following data is returned to the virtual machine (at the location specified by Rx):

Field	Description	Characteristics
System Name	"VM/PC"	8 bytes, EBCDIC
Version Number	The first byte is the version number, the second byte is the level, and the third byte is the Service Level number.	3 bytes, hexadecimal
Version Code	VM/PC CP executes the STIDPt1 byte, hexadecimal (Store Processor ID) instruction to determine the version code.	
MCEL	VM/PC CP executes the STIDPt2 bytes, hexadecimal instruction to determine the maximum length of the MCEL (Machine Check Extended Logout) area.	
Processor Address	Always zero	2 bytes, hexadecimal

Field	Description	Characteristics
Userid	The userid of the virtual machine issuing the DIAGNOSE.	8 bytes, EBCDIC
Program Product Bit Map	Identifies the program products that are installed. Valid values and the program products each identifies are:	8 bytes, hexadecimal

Value	Program Product
X'0100000000000000'	VM/PC

Upon return, Ry contains its original value less the number of bytes that were stored.

No completion code is returned, and the condition code remains unchanged.

DIAGNOSE Code X'08' -- Virtual Console Function

DIAGNOSE Code X'08' enables a virtual machine running in supervisor state to issue CP commands. The virtual machine must specify the command, the command parameters, and whether CP is to return the command response to the user's terminal or to a buffer. In addition to returning the command response, CP sets a completion code in the Ry register and may set a condition code.

When DIAGNOSE Code X'08' is issued, the Rx and Ry registers must be set up as follows:

Rx

Rx must point to the character string in virtual storage that contains the CP commands and parameters. If the character string contains multiple commands, each command and its associated parameters must be separated from adjacent commands by the value X'15'.

Ry

The high-order byte contains flag bits; the other three bytes specify, in bytes, the length of the CP commands and parameters. The maximum allowable length is 240 characters.

Set the flag bits as follows. If CP is to return the command response in a buffer, set the second flag bit to a value of one (X'40').

If the Ry register contains the value X'00000000', the DIAGNOSE Code acts as a no-operation (NOP) instruction. As a consequence, the issuing virtual machine is placed into a "CP-READ" state.

If the command response is to be returned in a buffer, Rx and Ry cannot be consecutive registers nor can either be register 15. In addition, the Rx+1 and Ry+1 registers must be setup as follows:

Rx+1
Rx+1 must point to the buffer in virtual storage where CP is to return the command response.

Ry+1
Ry+1 must specify, in bytes, the length of the buffer. This value must not exceed 8192.

If the command response is to be returned in a buffer, CP sets a condition code and returns information as follows:

condition code 0
The request was successful. The Rx+1 register points to the buffer that contains the command response. The Ry+1 register specifies the length of the response.

condition code 1
The request was unsuccessful. The response does not fit into the buffer. The Ry+1 register contains a value that specifies how many bytes of the response would not fit into the buffer.

When CP returns to a program executing a DIAGNOSE 8 instruction, the length value that was supplied in register Ry is replaced by the CP completion code value. This value is zero if the CP console function was successfully executed. If an error occurred, the completion code is the numeric value expressed in the message describing the error. For example, if error message DMKCFM045E is issued, CP sets a completion code of 45.

If the user has not specified a command response buffer, error messages and informational messages are generated according to the current values established by SET EMSG and SET IMSG commands.

If a command response buffer is used, error and informational messages are always put into the buffer instead of being written to the console. Each line of the response is followed by a new line character (X'15'). If the buffer is not long enough to contain all of the response lines, only as many complete lines as can fit into the buffer are supplied, so the last character written into the response buffer by CP is always a new line character. Any unused portion of the

response buffer is not changed. The setting of IMSG is ignored (it is considered always to be ON) and the setting of EMSG determines only whether the error message code is retained. (SET EMSG OFF is treated the same as SET EMSG ON; SET EMSG TEXT suppresses error message codes.)

The completion code values returned by CP are not affected by the values of EMSG and IMSG, or by the use of a command response buffer.

If CP is executing multiple commands and encounters an invalid command, processing stops and CP ignores the remaining commands.

Following are two examples showing how to specify DIAGNOSE Code X'08'. The first example shows how a program issues the QUERY SET command. In this example the response is returned to the user's terminal.

```
          LA      6,CMMD
          LA      10,CMMDL
          DC      X'83',X'6A',XL2'0008'
           .
           .
           .
CMMD      DC      C'QUERY SET'
CMMDL     EQU     *-CMMD
           .
           .
```

The second example shows how to specify a string of commands when multiple commands are to be issued.

```
          LA      6,CMMD
          LA      10,CMMDL
          DC      X'83',X'6A',XL2'0008'
           .
           .
           .
CMMD      DC      C'QUERY SET'
          DC      X'15'
          DC      C'SET RUN OFF'
CMMDL     EQU     *-CMMD
```

DIAGNOSE Code X'0C' -- Pseudo Timer

Execution of DIAGNOSE Code X'0C' causes CP to store four double-words of time information in the user's virtual storage. The register specified as Rx contains the address of the 32 byte area where the time information is to be stored. The address must be on a double-word boundary.

The first eight bytes contain the Month/Day-of-Month/Year. The next eight bytes contain the time of day in Hours/Minutes/Seconds. The last 16 bytes contain the virtual and real processor time used and each is simulated with the elapsed time (in micro-seconds) since the virtual machine was logged on. No completion code is returned, and the condition code remains unchanged.

DIAGNOSE Code X'10' -- Release Pages

Pages of virtual storage can be released by issuing a DIAGNOSE Code X'10'. When a page is released, it is considered all zero. The register specified by Rx contains the address of the first page to be released, and the Ry register contains the address of the last page to be released. Both addresses must be on page boundaries. A page boundary is a storage address whose low order three digits, expressed in hexadecimal, are zero. No completion code is returned, and the condition code remains unchanged.

Do not use DIAGNOSE Code X'10' to release noncontiguous virtual storage.

DIAGNOSE Code X'18' -- Standard DASD I/O

Input/Output operations to FBA type DASD minidisks, of the type used by VM/PC CMS, can be performed from a virtual machine using Diagnose 018. No interrupts are returned to the virtual machine; the instruction completes only when the read or write operation has completed. Rx contains the address of the minidisk; Ry points to the following chain of CCWs:

```
DEFINE EXTENT,extent,CC,16
LOCATE BLOCK,blockid,CC,8
READ or WRITE BLOCK,data,0,n*512
```

or

```
DEV CHAR,data,0,32
```

Where:

`extent`
defines the extent of the minidisk:

```
DS  X        MASK X'00'=allow write
                  X'01'=inhibit write
DC  X'00'    RESERVED
DC  X'0000'  BLKSIZE=512
DC  F'0'     DEVICE OFFSET TO FIRST BLOCK
DC  F'0'     FIRST BLOCK IN MINIDISK
DS  F        LAST BLOCK IN MINIDISK
```

`blockid`
defines the block(s) to read or write:

```
DS  X      SUB-COMMAND X'01'=write
                       X'05'=write/check
                       X'06'=read
DC  X'00'  RESERVED
DS  H      NUMBER OF BLOCKS
DS  F      BLOCK OFFSET WITHIN MINIDISK
```

data
the storage address of the data area to read into or write from. For improved performance, this data area should not cross a page boundary.

NOP (X'03') and TIC (X'08') CCW codes are also supported anywhere within the chain.

The condition and completion code are as follows:

CC=0
I/O complete with no errors

CC=1
Device availability error, reg 15 contains completion code:

R15=1 - Device not in virtual machine configuration
R15=2 - Device not available (e.g. disabled or powered off)
R15=3 - Attempt to write to read/only device.
R15=5 - Device busy

CC=2
CCW or control data error, reg 15 contains completion code:

R15=1 - extent error
R15=2 - operation error (operation byte in blockid doesn't match the READ or WRITE CCW)
R15=3 - invalid CCW chain format
R15=5 - CCW string not on double-word boundary

CC=3
Unrecoverable I/O error, reg 15 contains completion code:

R15=3 - CCW count not multiple of 512
R15=4 - End-of-File
R15=5 - Media I/O error

DIAGNOSE Code X'20' -- General I/O

Input/Output operations to the terminal and printer devices can be performed from a virtual machine using Diagnose 020. No interrupts are returned to the virtual machine; the instruction completes only when the read or write operation has completed. Rx contains the virtual device address; Ry points to the following CCWs:

Dasd: Same as Diagnose 18, except that the data transfer length need not be a multiple of 512 bytes.

Terminal control: X'03',data,0,control,1

Terminal write: X'09',data,0,len

Terminal read: X'0a',data,0,len

Printer write - Local printer:

```
X'03',data,CC,1       NOP
X'08',data,CC,1       TIC
X'01',data,CC,len     Print, no space
X'09',data,CC,len     Print, space 1
X'11',data,CC,len     Print, space 2
X'19',data,CC,len     Print, space 3
X'89',data,CC,len     Print, skip to top of page
X'0b',data,CC,1       Space 1
X'13',data,CC,1       Space 2
X'1b',data,CC,1       Space 3
X'8b',data,CC,1       Skip to top of page
```

Printer write - Remote Spool: Any valid CCW chain

Where:

len
: is the length of the data area. For the terminal input and local printer output, the maximum length is 132.

control
: control flags

  X'80'
  : quiet mode command control - used to replace RUNNING display status with WAITING.

data
is the storage address of the data area to read into or write from.

The condition and completion code are as follows:

CC=0
I/O complete with no errors

CC=1
Device availability error, reg 15 contains completion code:

R15=1 - Device not in virtual machine configuration
R15=2 - Device not available (e.g. disabled or powered off)
R15=5 - Device busy

CC=2
CCW error, reg 15 contains completion code:

R15=3 - invalid CCW
R15=5 - CCW string not on double-word boundary

CC=3
Unrecoverable I/O error, reg 15 contains completion code:

R15=1 - Print: Remote Services not available
R15=2 - Print: Spool space full
R15=3 - Print: Spool volume not mounted
R15=4 - Dasd: End-of-File
R15=5 - Dasd: Media I/O Error
R15=53- Print: Spool I/O Error

DIAGNOSE Code X'24' -- Device Type and Features

DIAGNOSE Code X'24' requests CP to provide the virtual machine with identifying information and status information about a specified virtual machine device. The virtual machine must specify the device address for which the information is requested. CP returns information about that device in the Rx, Ry, and Ry+1 registers. CP also provides a condition code identifying the specific device information returned to the virtual machine.

When a virtual machine issues DIAGNOSE Code X'24', the Rx register must contain the virtual device address for which information is requested or the values negative 1 or 2 (-1 or -2). Specify -1 when the device is the keyboard/display whose address is unknown to the virtual machine. Specify -2 when the device is the virtual machine printer whose address is unknown.

When CP returns control to the virtual machine, the Ry, Ry+1, and Rx registers contain device information. The Ry and Ry+1 register contain information about the device. If -1 was specified, the Rx register contains the address of the keyboard/display. If -2 was specified, the Rx register contains the address of the virtual machine printer.

The following diagrams identify specific information returned by CP and show how to locate this information in the Rx, Ry, and Ry+1 registers. The symbolic names used in these diagrams are the symbolic names used with VDEVBLOK, and RDEVBLOK in *VM/SP Data Areas and Control Block Logic.*

Rx Register:

Byte 0	Byte 1	Byte 2 Byte 3
		virtual device address

Ry Register:

Byte 0	Byte 1	Byte 2	Byte 3
VDEVTYPC	VDEVTYPE	VDEVSTAT	VDEVFLAG

Symbolic Name Meaning:

```
VDEVTYPC    Virtual device type class
VDEVTYPE    Virtual device type
VDEVSTAT    Virtual device status
VDEVFLAG    Virtual device flags
```

Ry+1 Register:

Byte 0	Byte 1	Byte 2	Byte 3
RDEVTYPC	RDEVTYPE	RDEVMDL	RDEVFTR - or - REDVLLEN

Symbolic Name	Meaning
RDEVTYPC	Real device type class
RDEVTYPE	Real device type
RDEVMDL	Real device model number.
RDEVFTR	Real device feature code for a device other than the keyboard/display.
RDEVLLEN	Current device line length for the keyboard/display.

The following chart lists the condition codes CP can return for DIAGNOSE Code X'24', the meaning of each condition code, and the registers where data is returned.

If the condition code equals	This register contains information			Comments
	Rx[1]	Ry	Ry+1[2]	
0	X	X	X	Normal completion
3				Invalid device address or the virtual device does not exist.

1 The Rx register contains information only when DIAGNOSE Code X'24' specifies a keyboard/display or virtual machine printer whose address is unknown.

2 If Ry is register 15, CP returns only virtual device information: no information is returned in register Ry+1.

DIAGNOSE Code X'54' -- Control The Function of the PA2 Key

DIAGNOSE Code X'54' controls the function of the PA2 key. The PA2 key can be used either to simulate an external interrupt to a virtual machine or to clear the output area of a display screen.

The function performed depends upon how Rx is specified when DIAGNOSE Code X'54' is issued. If Rx contains a nonzero value, the PA2 key simulates an external interrupt to the virtual machine. If Rx contains a value of zero, the PA2 key clears the output area of the display screen.

The external interrupt is simulated only when the display screen is in the VM READ, HOLD, or MORE status and the TERMINAL APL ON command has been issued.

DIAGNOSE Code X'58' -- 3270 Virtual Console Interface

DIAGNOSE Code X'58' enables a virtual machine to communicate with 3270 display stations. Using DIAGNOSE Code X'58', a virtual machine may:

- Display up to a full screen of data using only one write operation.
- Provide attribute characters along with data that is sent to the display station. An attribute character provides control information for the data, for example, a request to intensify the data when it is displayed.
- Place a 3270 display station under control of the virtual machine (full screen mode).

When a virtual machine issues DIAGNOSE Code X'58', the virtual machine must provide one or more channel command words (CCWs). These CCWs specify the 3270 operation to be performed, provide control information for the display station, and specify the address of data to be displayed during a write operation or the address of a buffer where data is to be stored during a read operation. If only one CCW is used, the Rx register must contain its address. If CCWs are chained, the Rx register must contain the address of the first CCW in the chain. The Ry register must contain the virtual address of the display station where the operation is to be performed. This value must be right-justified.

Displaying Data

To display up to a full screen of data, code a CCW using the following assembler language instructions:

```
DS OD
DC AL1(CCWCODE),AL3(DATADDR),AL1(FLAGS),AL1(CTL),AL2(COUNT)
```

Where:

CCWCODE
the command code X'19'.

DATADDR
is the virtual storage address of the first byte of data to be displayed.

FLAGS
are standard CCW flags. The suppress-incorrect-length indicator, bit 34, must be set to a value of one. Set other bits as needed.

CTL
is a control byte defined as follows:

- If the high-order bit (bit 0) is on, CP displays more status before erasing the display station screen and displaying the new data.

- Bits 2-7 identify the line on the display screen where the display is to start. A value of 0 (B'xx00 0000') corresponds to the first or top line, a value of 1 (B'xx00 0001') corresponds to the second line and so forth.

- If the control byte contains the value X'FF', CP clears the output area on the display screen. No new data is displayed.

- CCWs may be command chained to combine several operations in one DIAGNOSE. If the control byte contains a value of X'FE', CP will:

  - Erase the entire screen.
  - Rewrite the attribute bytes for the CP screen format.
  - Reset the cursor to the beginning of the input area.

COUNT
specifies the number of bytes of data to be displayed. The maximum that can be specified for this command code is 2032 bytes.

To provide attribute characters for the data, place the attribute character in the data stream immediately following a 3270 start-field order. The start-field order, a one-byte

value, notifies the 3270 display system that the next byte in the data stream is an attribute character. For a description of how the 3270 display system uses attribute characters, and to determine the values to specify for attribute characters and the start-field order, see the *IBM 3270 Information Display System Component Description.*

After processing DIAGNOSE code X'58', CP sets a condition code. If the operation was successful - that is, no I/O errors occurred - CP sets a condition code of zero. If an I/O error occurred, CP sets a condition code of one.

If an I/O error occurred, the application program can check the I/O status and the error type by:

- Issuing a TEST I/O (TIO) instruction
- Examining the returned condition code
- Examining the virtual CSW

The returned condition codes and CSW status are the standard condition codes and status defined in the *IBM System/370 Principles of Operation.*

You must also make sure that the interrupt for the virtual device is enabled by setting the appropriate bit and channel mask in the PSW. For example, if the virtual address of your console is 01F, bit 0 in the channel mask must be set to one (that is, bit 0 must be on). This may be the case if you are loading programs in the VM/PC CMS transient area.

Full Screen Mode

DIAGNOSE X'58' provides a means by which a virtual machine may share, with CP, control of a 3270 display station. Two CCW operations, X'29' and X'2A', in addition to performing the requested I/O, notify CP that the display station is operating under the control of the virtual machine.

CCW code X'29' performs a WRITE, or ERASE/WRITE, operation, depending on the value of the control field. For the WRITE, and ERASE/WRITE, the virtual machine must provide appropriate control information beginning with the

Write Control Character (WCC) and including 3270 orders following the WCC. Data may be written anywhere on the screen. The virtual machine must provide the address where the write is to begin; it uses a SET BUFFER ADDRESS (SBA) order to do this. Writing can also start at the current cursor address.

CCW code X'2A' performs a READ BUFFER or a READ MODIFIED operation, depending on the value of the control field.

To specify the full screen mode CCW, use the following assembler language instructions:

```
DS 0D
DC AL1(CCWCODE),AL3(DATADDR),AL1(FLAGS),AL1(CONTROL),AL2(COUNT)
```

Where:

CCWCODE
is a CCW code (X'29' or X'2A')

DATADDR
for a write operation, specifies the first byte of the data stream (WCC) to be written. For a read operation, specifies the address of the read buffer.

FLAGS
is the standard CCW flag field.

CONTROL
for a write operation, an ERASE/WRITE is performed by specifying a ccwcode of X'29', if the high-order bit (X'80') is on.

If the high-order bit (X'80') is on for a read operation, a READ MODIFIED is performed; otherwise, a READ BUFFER is performed. The addition of X'10' to the CONTROL field values for ERASE/WRITE making it X'90' causes the PA1 key interrupt to be reflected to the virtual machine. This replaces the normal PA1 key function of returning the virtual machine to CP mode.

This allows a virtual machine to have full control of the keyboard. Normal PA1 key function is restored when full screen mode is reset.

COUNT
for a write operation, specifies the number of bytes to be displayed plus the number of bytes of control information. For a read operation, specifies the number of display characters to be read plus the number of bytes of control information. The maximum number of bytes that can be specified is 3000. The maximum number of displayable positions is 1920.

Full Screen Interactions

The virtual machine console exists in either of two modes, CP mode and full screen mode. CP mode is the default screen mode and is indicated by the screen status field in the lower right-hand corner of the screen. When in CP mode, the screen format is controlled by CP, and the data that appears on the screen is provided by CP and the programs running in the virtual machine. Full screen mode is initiated by the application program running in the virtual machine. When in full screen mode, the screen format and data are under complete control of the program running in the virtual machine.

CP mode is terminated and full screen mode is initiated when the application program issues an ERASE/WRITE instruction. Full screen mode may be terminated by a CP mode type I/O to the screen any time the keyboard is in a locked state.

Interactions between CP and the application program in the virtual machine using full screen support are listed below. The application programmer must be familiar with the operation of the IBM 3270 display station. Detailed information on its operation can be found in the *IBM 3270 Information Display System Component Description.* Also listed below are general programming considerations that must be followed to effectively use the DIAGNOSE X'58' instruction for full screen I/O.

1. A full screen ERASE/WRITE operation establishes full screen mode.

2. The application program is responsible for all I/O status and error checking, just as if START I/O (SIO) were being used instead of DIAGNOSE. This is done by using the TEST I/O (TIO) instruction and examining the returned condition code, and by examining the virtual CSW. The returned condition codes and CSW status are the standard condition codes and status as defined in the *IBM System/370 Principles of Operation*, with one exception noted below in number 3.

3. When a mode switch has occurred and the screen is in CP mode, the application program is notified by an X'8E' in the CSW unit status byte following a full screen I/O operation. An ERASE/WRITE instruction should be issued to reestablish full screen mode and reformat the screen. If control of the PA1 key interrupt had been transferred to the virtual machine via the CONTROL option, it must be respecified to return PA1 key control back to the virtual machine. Otherwise, depression of the PA1 key places the display in CP mode.

   An X'8E' in the CSW unit status byte following an ERASE/WRITE ALTERNATE instruction indicates that non-full screen data (CP mode) is waiting to be read. The application program should issue a non-full screen READ and then reissue the ERASE/WRITE instruction.

4. Non-full screen virtual machine messages are displayed immediately when in full screen mode.

5. The application program must establish an environment to handle attention interruptions. This could be done using the CMS macros HNDINT and WAITD. An attention interruption to the application program when CP receives an attention interruption indicating that the virtual machine console operator has caused an interruption. (For example, depressed the ENTER or a PF key on the display keyboard). The application program should respond by issuing a READ.

6. The application program must also establish an environment to handle I/O interruptions and must ensure that channel end and device end have been received before processing continues.

7. If the test request key is depressed when in full screen mode, X'604040' is returned to the application program in the read buffer.

8. If you press the PA1 key in full screen mode, CP posts an attention interrupt to your virtual machine. If the virtual machine does not respond with a READ and you press the PA1 key a second time, your virtual machine is put in CP mode and "CP READ" is displayed in the screen's status area. However, if you set bit X'10' of the CONTROL option on before the initial ERASE/WRITE and press the PA1 key, the interrupt is reflected to your virtual machine for handling. If you have not set bit X'10' of the CONTROL option on and you press the PA1 key, your virtual machine is put in CP mode and "CP READ" is displayed in the screen's status area.

DIAGNOSE Code X'5C' -- Error Message Editing

Execution of DIAGNOSE Code X'5C' causes the editing of an error message according to the user's setting of the EMSG function:

Rx
contains the address of the message to be edited.

Ry
contains the length of the message to be edited.

The EMSG options CODE and TEXT are tested, and DMKHVC tests the VMMLEVEL field of the VMBLOK. Rx and Ry are modified as follows:

LEVEL		Registers on Return	
CODE	TEXT	Rx	Ry
ON	ON	no change	no change
ON	OFF	no change	10 (length of code)
OFF	ON	pointer to text part of message	length of text alone
OFF	OFF	N/A	0

Note: DIAGNOSE Code X'5C' does not write the message; it merely rearranges the starting pointer and length. For CMS error messages, a console write is performed following the DIAGNOSE unless Ry is returned with a value of 0.

DIAGNOSE Code X'60' - Determining the Virtual Machine Storage Size

Execution of DIAGNOSE Code X'60' allows a virtual machine to determine its size. On return, the register specified as Rx contains the virtual machine storage size.

DIAGNOSE Code X'64' - Finding, Loading, and Purging a Named Segment

Execution of DIAGNOSE Code X'64' controls the linkage of discontiguous saved segments. Since VM/PC CP does not support saved segments, the function return code to all subcodes will be:

Return Code	Meaning
44	Named segment does not exist

Appendix B. The 370 Processor Control Session

The 370 Processor Control Session is a general purpose debug facility similar to the debug facilities found on the operator console of other IBM 370 processors. It runs on the IBM Personal Computer XT/370 as one of several concurrent sessions. It can be entered from any other session and conversely, can exit to any other session.

The 370 Processor Control Session allows you to debug code by providing the following functions to the 370 processor:

- Stopping and starting the processor
- Instruction stepping the processor
- Stopping the processor by instruction address compare
- Reset and Clear Reset of the 370 processor
- Generation of an External Interrupt to the 370 processor
- Full screen editing of the following:
  - the 370 General Purpose Registers
  - the 370 Floating Point Registers
  - the 370 Control Registers
  - the 370 Program Status Word Register
  - the 370 Storage, both Real and Virtual
  - the 370 Page Address Table

370 Processor Control - Screen Format

The 370 Processor Control Session is a full screen utility designed to increase the productivity of debugging sessions through the use of such features as full screen register and storage editing. There are no commands to remember. All actions are communicated to the 370 Processor Control utility through the use of function keys. The description of these keys is on the screen in case you should forget them

(there are very few). The 370 Processor Control Session was also designed to be keystroke efficient. That is, actions you request require a minimum amount of keystrokes.

The 370 Processor Control screen is divided into four areas as indicated in Figure B-1 . The documentation that follows references these areas. Figure B-2 shows the actual layout of the screen.

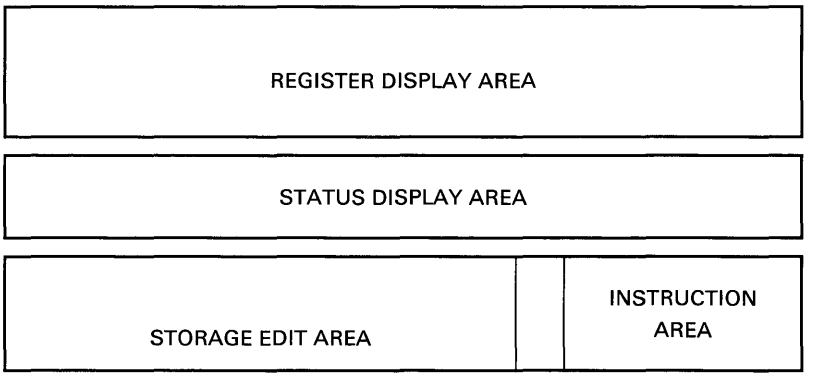


Figure B-1. Four Areas of 370 Processor Control Screen

```
                              370 PROCESSOR CONTROL
GPR0 .... ....        GPR1 .... ....        GPR2 .... ....        GPR3 .... ....
GPR4 .... ....        GPR5 .... ....        GPR6 .... ....        GPR7 .... ....
GPR8 .... ....        GPR9 .... ....        GPRA .... ....        GPRB .... ....
GPRC .... ....        GPRD .... ....        GPRE .... ....        GPRC .... ....
PSW .... .... .... ....                                 PROCESSOR STATUS - RUNNING
I-COUNTER  ......  CC ..  CMWP                          ADDRESS COMPARE (REAL) ......
              MAIN STORE EDIT ( REAL )                      FUNCTION KEY DEFINITION
000000  .... .... .... ....  .... .... .... ....          --------------------
000010  .... .... .... ....  .... .... .... ....          F1 - INSTRUCTION STEP
000020  .... .... .... ....  .... .... .... ....          F2 - STOP / START
000030  .... .... .... ....  .... .... .... ....          F3 - EDIT (GPR/FPR/CR)
000040  .... .... .... ....  .... .... .... ....          F4 - EDIT (MAIN/PAT)
000050  .... .... .... ....  .... .... .... ....          F5 - EDIT (REAL/VIRT)
000060  .... .... .... ....  .... .... .... ....          F6 - EBCDIC ( ON/OFF )
000070  .... .... .... ....  .... .... .... ....          F7 - PAGE FORWARD
000080  .... .... .... ....  .... .... .... ....          F8 - PAGE BACKWARD
000090  .... .... .... ....  .... .... .... ....
0000A0  .... .... .... ....  .... .... .... ....          ALT-F6  - PROGRAM RESET
0000B0  .... .... .... ....  .... .... .... ....          ALT-F7  - CLEAR RESET
0000C0  .... .... .... ....  .... .... .... ....          ALT-F8  - EXTERNAL INTR
```

Figure B-2. Actual Layout of 370 Processor Control Screen

F2 Key - Start/Stop

You use the F2 key to start and stop the 370 processor. Normally, when you enter the 370 Processor Control session, the processor will be in the "RUNNING" state. (See Processor State below.) To stop the processor, press the F2 key. The processor status field on the screen indicates that the processor is "STOPPED." To start the processor, press the F2 key again. Note that you use the F2 key to both stop and start the processor. If the processor is stopped, pressing F2 starts it. If the processor is started, pressing F2 stops it. Many of the other function keys have this same reverse action.

When you stop the processor, the register and storage areas on the screen are unlocked. You can then move the cursor around the screen and change register and storage contents. Conversely, when you start the 370 processor, these areas are locked.

Processor Status Field

The 370 processor can be in one of 5 states. The processor status indicator field, located in the status area of the screen, identifies the processor state. This field is the only one on the screen that blinks.

The five possible processor states are:

- "STOPPED" State
- "RUNNING" State
- "ADDR-COMPARE" State
- "INSTR-STEP" State
- "ISTEP-WAIT" State

The "STOPPED" and "RUNNING" states indicate whether or not the processor is executing instructions. "STOPPED" indicates that the processor is not executing instructions, and the user is free to alter storage and register contents.

"RUNNING" indicates the processor is executing instructions or could possibly be in the 370 wait state. The storage and register locations can not be altered in this state.

The "ADDR-COMPARE" (Address Compare) state indicates that an address compare match has occurred. This will appear on the screen without user prompting. The processor is stopped in this state. You would normally use the F1 (Instruction Step) or F2 (Start) keys to exit this state.

When you press the F1 key, the processor enters the "INSTR-STEP" (Instruction Step) state. You would enter this state to force the 370 processor to execute one 370 instruction with each press of the F1 key. The processor is stopped in this state. You would normally use the F2 (Start) key to exit this state.

When you press the F1 key and the processor is in the wait state, the processor enters the "ISTEP-WAIT" (Instruction Step - Wait) state. The 370 instruction counter does not advance in this state because the processor will not execute instructions in the wait state. As in the "INSTR-STEP" state, the processor is stopped. You would normally use the F2 (Start) key to exit this state.

F1 Key - Instruction Step

You press the F1 key if you want to have the processor execute one instruction at a time. If the processor was in the "RUNNING" state, pressing F1 has the same effect as stopping the 370 processor, and the status area indicates the "STOPPED" state. Each time you press the F1 key, the 370 processor will execute one 370 instruction. The status area will indicate "INSTR-STEP." You would normally exit from this state by pressing the F2 (Start) key.

Editing 370 Registers

The registers you would most commonly edit are the 370 General Purpose Registers (GPRs) and the 370 Program Status Word Register (PSW). Both are displayed on the screen.

The 370 Program Status Register is always displayed on the screen and you may edit it when the processor is in the "STOPPED" state. If the processor is not stopped, the field is locked.

Note that the 370 Instruction Counter, the 370 Condition Code, and the CMWP fields are taken from the PSW and displayed immediately below the 370 Program Status Word Register. You may not change these fields. If you want to change them, you change the PSW.

F3 Key - Edit (GPR/FPR/CR)

You can edit the 370 GENERAL PURPOSE REGISTERS if they are displayed on the screen and the fields are unlocked. Again, the fields are unlocked if the processor is stopped.

If you want to edit the 370 FLOATING POINT REGISTERS or the CONTROL REGISTERS, they must be displayed on the screen. Press the F3 key to display them.

When you press the F3 key the register display area of the screen is overlayed with a different set of registers. Initially, the 370 General Purpose Registers are displayed in this area. If you press F3, the Floating Point registers are displayed. Press F3 again and the Control Registers are displayed. Press F3 again and the General Purpose registers are again displayed. You can move through displays of these 3 sets of registers by pressing the F3 key.

Storage Edit

The storage display area of the screen initially displays the 370 real main storage. Sixteen bytes of storage are displayed on thirteen consecutive lines. The six digit addresses are displayed on the left side of the screen, followed by the storage contents. The fields are unlocked so you may edit them if the processor is stopped.

You can specify the starting address of the storage display by typing an address in the storage edit address field. Note that this field is located where the first storage address is displayed on the screen. (See Figure B-1 on page B-2.) It is the only address that isn't highlighted, and the cursor is generally placed at the beginning of this field. You need not

supply leading zeros of the address, but can type any non-valid hexadecimal character after the last entered digit. If you enter an invalid address, such as too large an address, the 370 Processor Control utility forces the address to zero.

The storage address field is unlocked, even if the processor is running. You may display storage at all times but can only alter it if the processor is stopped. You must press the enter key to refresh the storage data on the screen. In other words, the storage data on the screen is not constantly updated as a running program is changing it. You must request periodic updates,if required, by pressing the enter key. You cannot update displayed registers this way. The contents of displayed registers are only updated when the processor is stopped.

If you enter an invalid character in any of the input fields (except the storage edit address and address compare stop fields), the alarm will sound, the input field will be displayed in reverse video format, and the cursor will be placed under the invalid character prompting you for a correction.

F7 and F8 Keys - Page Forward and Backward

Press the F8 key to have the next (higher address) block of storage displayed on the screen. Press the F7 key to have the previous (lower address) block of storage displayed.

F6 Key - EBCDIC (On/Off)

Press the F6 key to display the EBCDIC representation of the data. The EBCDIC representation overlays the instruction area of the screen. Press the F6 key again to have the instructions displayed again. Since the instruction display is generally more important then the EBCDIC display, if you press almost any of the attention generating keys (F1,F2,F3, etc.) the instructions reappear on the screen. Only the F7, F8, and ENTER keys don't cause this to happen.

F5 Key - Edit (Real/Virtual)

Press the F5 key to display VIRTUAL STORAGE. The virtual storage display overlays the real 370 storage display. Press F5 again to display the real 370 storage again. By pressing F5 you can switch between displays of real and virtual storage in the storage display area of the screen. If the data for a virtual address is not currently in storage, the storage contents will be displayed as "X"s and the fields will be locked.

F4 Key - Edit PAT/Edit Main

Press F4 to edit the PAT (Page Address Table). This is the hardware table that maps virtual addresses to real addresses. The PAT edit screen data overlays the 370 storage edit data. Pressing the F4 key again causes the storage edit data to be displayed again.

The PAT edit format has a column of virtual addresses on the far left of the screen, followed by columns of the corresponding real addresses, the valid bits, the reference bits, and the change bits respectively. For further information on the PAT, see the "IBM Personal Computer XT/370 Technical Manual," and the section on Dynamic Address Translation in the "370 Principles of Operations."

You should only alter the PAT data if you absolutely know what you're doing. Virtual storage management is a function of the 370 operating system. Altering this data could cause unpredictable results.

Address Compare Stop Function

The address compare feature automatically stops the 370 processor when it detects an instruction execution at an address you've entered. To use this function, enter the address in the Address Compare Field located in the status area on the screen. You need not supply leading zeros of the address, but may type any non-valid hexadecimal character after the last entered digit. If you enter an invalid address, such as too large, the 370 Processor Control utility makes

address compare inactive. You can cancel an address compare simply by typing a non-hexadecimal character in the first digit location.

This function has some limitations:

- It is only an instruction compare stop and not a data compare stop.
- The address entered must be the address of the start of the instruction. Entering an address in the middle of an instruction could cause the program to have unpredictable results.
- It only works with addresses in real 370 storage (as opposed to virtual addresses).

ALT F6 Keys - Program Reset

If you simultaneously press the Alternate key and the F6 key, you cause a Program Reset to occur to the 370 processor (see Program Reset in "370 Principles of Operations"). The processor is in the "STOPPED" state at the end of this operation.

ALT F7 Keys - Clear Reset

If you simultaneously press the Alternate key and the F7 key, you cause a Clear Reset to occur to the 370 processor (see Clear Reset in "370 Principles of Operations"). This clears main storage. The processor is in the "STOPPED" state at the end of this operation.

ALT F8 Keys - External Interrupt

If you simultaneously press the Alternate key and the F8 key, you cause an External Interrupt to occur to the 370 processor (see External Interrupt in "370 Principles of Operations").

Appendix C. Comparison of VM/PC and VM/SP

Relationship of VM/PC and VM/SP

VM/PC is based on VM/SP Release 2. VM/PC has changed some aspects of VM/SP Release 2 to fit its environment.

Externally, some of the commands and command options available on VM/PC are not the same as those available on a VM/SP Release 2 system. In addition, some of the command response formats may be different from a VM/SP Release 2 system. Some VM/SP Release 2 commands and options are not present in VM/PC, since VM/PC does not have support for tape I/O, virtual readers, or virtual punches. In addition, the CMS/DOS environment is not supported on VM/PC.

Internally, the control blocks of CMS are identical in format with those of VM/SP Release 2 CMS.

If we examine a typical CMS session, some of the differences between VM/PC and VM/SP Release 2 can be pointed out and explained.

- Logon screen

  You don't have to clear the system logo in order to log on to VM/PC. In addition, you may specify an environment to start or you may have a default environment, specified in your system configuration file.

- CP Commands

  CP commands that are allowed to have multiple operands in VM/SP (for example, DISPLAY) will be allowed only one operand by VM/PC CP.

- CMS Initialization

  When you start (IPL) CMS, you may pass a parameter on the command line. If you specify the environment on the logon screen or in the autoipl configuration option, the system will pass the parameter 'AUTOIPL'. VM/PC CMS does not prompt you for a first command, as VM/SP Release 2 CMS does. The initialization of CMS continues, uninterrupted by a prompt.

  The initialization of CMS involves accessing the '100' minidisk as your S disk. Only the mode 2 files will be accessed. The initialization of CMS then checks to see if an exec called 'SYSPROF EXEC' exists on this S-disk. If it does, the command 'EXEC SYSPROF' is executed and the parameter passed to CMS on the IPL command is passed to this exec.

  The SYSPROF exec checks for the parameter 'NOSPROF', which you specify if you don't want the system profile to be executed. If you specify 'NOSPROF', the exec ends without doing anything else. Otherwise, the exec issues the command 'SET OSSIM ON' to start OS Simulation, and sets the system synonyms from the file 'SYSTEM SYNONYM' on the S-disk.

  If you have a minidisk defined at address '101', that disk is accessed as your A-disk. If you have a minidisk defined as '10E', that disk is accessed as your Y-disk. Unlike VM/SP Release 2, the Y disk will not be selectively accessed.

  If you specified the keyword 'NOPROF' on the IPL of CMS, your user profile is not executed. If you didn't specify 'NOPROF', the file 'PROFILE EXEC' is executed. The parameters you used to start CMS are passed to your profile exec, where you may test for special keywords.

  This initialization is similar to the way VM/SP Release 2 initializes, although it is not identical. The initialization

of CMS in VM/PC is less complex, and allows you to tailor your CMS environment for special requirements you may have.

- Screen format

  The screen format of VM/PC is similar to that of VM/SP Release 2, although it is not identical. The scroll field at the bottom right of the screen allows you to scroll back through the contents of the terminal history buffer. In addition, VM/PC has two additional options on the CP TERMINAL command, which allow you to control the action of the PA2 key and the display of information lines longer than the width of the screen. Also, the VM status area on the lower right of the screen differs from that of VM/SP Release 2. It will indicate 'Waiting' when the system is not actually doing work for you and is waiting for you to enter a command. The CMS prompt 'Ready;' is also slightly different than that of VM/SP.

- CMS messages

  Most messages VM/PC CMS will display to you are in mixed case, unlike the messages displayed by VM/SP. In addition, the text of these messages comes from a system message file which is on the S-disk.

- Commands do not support tape drives, virtual readers, or punches.

  The support for these devices is not present in VM/PC, and commands which support these devices are not present.

- IDENTIFY command output

  Since there is no RSCS virtual machine on your VM/PC, the fields in the IDENTIFY command output which indicate node and userid of the RSCS virtual machine are filled with token placeholders of '*'. Also, since you set the timer of your VM/PC, the time zone is designated as 'LOCAL' in the command output.

- CMS Maclibs

  The macro libraries which are on the CMS 100 minidisk contain a subset of the macros available on a VM/SP Release 2 CMS system. The CMS and OS maclibs are named 'CMSPCLIB MACLIB' and 'OSPCLIB MACLIB', respectively.

- I/O interface changes.

  VM/SP CMS uses the SIO instruction to support the console. Since VM/PC does not support this instruction, VM/PC CMS has changed the console support to use the DIAGNOSE X'20' support in VM/PC CP.

- Storage Keys

  Although storage protection is not performed, the storage keys have been retained to provide compatibility for programs that use the storage keys to determine system and user area partitioning.

- Interval Timer

  Since VM/PC does not support the interval timer, the timer routines in OS Simulation have been modified to use the CPU timer.

- File System

  You cannot create or access local minidisks formatted in 800-byte blocks. You may access remote minidisks formatted at this size, however. VM/PC supports CMS minidisks in 512, 1024, 2048, and 4096-byte sizes. VM/PC chose these as the best sizes for minidisks, since they must be mapped into Personal Computer DOS files, which are maintained in 512 byte blocks.

  In addition, the support for remote minidisks and remote files has changed the CMS file system somewhat. The FSTs for remote minidisks which you access are not copied to your local session storage. Rather, all requests for file system functions are sent to the remote server program. The results of these requests are passed back

to you as if the files and disks were locally attached. Some of the fields in the internal file system tables ADT and AFT are used differently when the minidisk or file is located remotely.

- NUCXLOAD Command

  The NUCXLOAD command in VM/PC is different from that of VM/SP Release 2. The VM/PC command is located in the CMS nucleus, and supports the loading of modules which are adcon free and those which have their RLD information saved with the module, through the use of the 'RLDSAVE' option of LOAD and INCLUDE. Also, the NUCXLOAD command doesn't support loading of modules from OS LOADLIBs, as this support does not exist in VM/PC.

- XEDIT and EXECIO Initialization

  Both XEDIT and EXECIO are initialized differently in VM/PC than in VM/SP Release 2. Both programs make use of the capabilities of the NUCXLOAD command to load the programs as nucleus extensions. After the first invocation of these commands, the nucleus extension will normally be used when you invoke the command. In addition, OS Simulation uses of this capability to load the OSSIM module into your virtual storage to provide the capabilities of OS Simulation in your CMS environment.

- XEDIT

  In addition to the difference in the initialization of XEDIT, the messages and screen format of XEDIT use mixed case to facilitate the reading of information. Also, the default PF keys for XEDIT set the PF1 key to 'TOP' rather than 'HELP', since the HELP facility does not exist on VM/PC.

- FILELIST

  The defined PF keys for FILELIST set the PF1 key to 'SORT on name' instead of HELP.

- Simplification

  - Command Interpreter Simplification

    VM/PC has simplified the CMS command interpreter to eliminate infrequently used options in favor of greater performance. For instance, the BLIP capability has been removed.

  - VM/PC has removed the SVCTRACE capability, since the VM/PC CP TRACE command provides an advanced trace capability which allows tracing SVC execution in CMS.

Moving Application Programs from VM/SP to VM/PC

It is expected that application programs to be run on VM/PC will be either downloaded from a remote system or developed to run on both a VM/PC and a remote system. Once the desired application has been installed on VM/PC (by whatever means), VM/PC can function alone as long as the application has no need of the remote system.

VM/SP Commands Not Supported in VM/PC

The following VM/SP 2 CMS commands are not supported:

AMSERV	FETCH	NOTE	RSERV
ASSEMBLE	HELP	OPTION	RUN
ASSGN	LABELDEF	OSRUN	SENDFILE
CMSBATCH	LISTDS	PEEK	SETPRT
DDR	LISTIO	PSERV	SSERV
DISK	LKED	PUNCH	SVCTRACE
DLBL	LOADLIB	RDR	TAPE
DOSLIB	MOVEFILE	RDRLIST	TAPEMAC
DOSLKED	NAMEFIND	READCARD	TAPPDS
DSERV	NAMES	RECEIVE	TELL
EDIT			
ESERV			

Finding Further Information in VM/SP Library

You can find further information about using VM/PC in the VM/SP Release 2 library. We list the names and order numbers of those books below.

Note: When seeking further information for using VM/PC in the VM/SP library, remember that those books were written for VM/SP users and contain information about features and functions not supported in VM/PC. Keep in mind the differences listed earlier in this appendix when consulting the VM/SP library.

Here's a table to help you find specific information in the VM/SP library:

Information Sought	VM/SP Book to Look In
System Product Editor (XEDIT) Use	*SP Editor User's Guide*
System Product Editor (XEDIT) Commands	*SP Editor Command Reference*
EXEC 2 Use	*EXEC 2 Reference*
CMS Commands	*CMS Commands and Macros*
EXEC Use	*CMS User's Guide*
CMS or CP Messages	*System Messages and Codes*

Order Numbers for VM/SP Release 2 Books

The order numbers for the VM/SP Release 2 books are:

- *Introduction,* GQ19-6200
- *System Programmer's Guide,* SQ19-6203
- *System Messages and Codes,* SQ19-6204
- *Terminal User's Guide,* GQ19-6206
- *CMS Command and Macro Reference,* SQ19-6209
- *CMS User's Guide,* SQ19-6210
- *EXEC 2 Reference,* SQ24-5219
- *System Product Editor User's Guide ,* ST24-5220
- *System Product Editor Command and Macro Reference,* SQ24-5221
- *Data Areas and Control Block Logic,* LQ60-0891
- *System Logic and Problem Determination Guide, Volume 2 (CMS),* LQ60-0893
- *Quick Guide for Users,* ST00-0913
- *Command Summary (General Users),* ST00-0914
- *System Product Editor Command Reference Summary,* ST00-0918
- *EXEC 2 Language Reference Summary,* ST00-0917

Appendix D. Operating Hints

Backing up Minidisks

VM/PC minidisks are managed as files with a "filename.extension" of "userid.device address" and may be exist on either a fixed disk drive, or a diskette drive. You may wish to periodically back-up your minidisks to prevent data loss or for security reasons. Since a minidisk is a DOS 2.00 file, it may be backed up with any of the backup facilities available with DOS 2.00.

Minidisks on Diskettes

If you define a minidisk on a diskette drive, you should not remove the diskette while a command is executing. You should wait for all commands you have issued to complete before removing the diskette from the drive. In addition, any time you insert such a minidisk in the drive, issue the ACCESS command to update the virtual storage copy of the new minidisk's directory.

Also, if the CMS minidisk to be used does not yet exist on the diskette, you should issue the CP LINK command, prior to using the CMS minidisk, to ensure that the proper DOS file allocation takes place on the diskette prior to your using the minidisk. Failure to do this may result in unreadable data on the diskette.

Transporting CMS Files

There are two methods of transporting CMS files between VM/PC systems. The first uses the EXPORT and IMPORT commands; the second uses a minidisk defined on a diskette.

EXPORT/IMPORT Method

The file to be transported is EXPORTed to a diskette (the Personal Computer DOS filespec is given as an operand of the EXPORT command). The diskette is then carried to

another system and the IMPORT command is used to load the file. This method should be used only to transport character data files.

Minidisk Method

Your configuration file must include a definition for a minidisk on your diskette drive. Since minidisks are files whose names are in the form "userid.device address," it is helpful to have a common userid on both systems with identical minidisk definitions. This userid need not be the one that you use normally, but you must be able to link to that minidisk in R/W mode (it might be preferable to use a separate userid for the diskette minidisk because your logon will take considerably longer if the diskette drive is empty or open).

The file to be transported is copied onto the diskette minidisk with the COPYFILE command and when the command is completed, the diskette may be removed and transported to another system. (If the userids and device addresses do not match, as suggested above, the minidisk file must be renamed accordingly.) The minidisk can then be accessed and the file can be copied to any other minidisk.

File Access Performance

As with any Personal Computer DOS file, the files used by VM/PC are subject to performance degradation if they are fragmented (its records are not allocated contiguously). The Personal Computer DOS CHKDSK command can be used to determine if a file is fragmented.

Contiguous record allocation is extremely important for the paging file. Since it is normally deleted when you logoff, you will have to terminate VM/PC (using CTRL-BREAK) with the Local 3277 Session still active. You can then use the CHKDSK command to determine whether the paging file is fragmented.

If you suspect that a fragmented file is causing a performance problem, use the Personal Computer DOS COPY or BACKUP commands to eliminate the fragmentation. (For a performance problem, see also "Using DOS Features," which follows.) In the case of the paging

file, which is allocated by VM/PC during its execution, all files on the drive where the paging file resides should be copied; this will remove small areas of unused space that can result in a fragmented paging file.

Using DOS Features

DOS tree-structured directories can be used for the files needed to run VM/PC. However, when installing VM/PC, all the files are copied to the root directory. If you wish to set up another DOS directory for your VM/PC files, you will need to copy (using the DOS COPY command) the VM/PC files to the desired directory. After doing so, you can erase the VM/PC files from the root directory if you wish.

Both VM/PC and DOS provide facilities for printing files while executing other tasks. When VM/PC is running, it attempts to control the printer for its own printer spooling functions. For this reason, the DOS PRINT command queue should be empty prior to invoking VM/PC.

Certain DOS commands can be used to configure your system. One of these, the BUFFERS command, is used to set the number of buffers DOS is to reserve at startup time. The number chosen can ***significantly*** affect the performance of programs running under VM/PC. To find the value best for you, we suggest you measure the performance of your most frequently used VM/PC applications with the number of buffers set to several different choices. While there is no one number that is best for all programs, a choice of between 10 and 15 buffers is recommended as a starting point.

FORMAT Command Performance

The performance of the CMS FORMAT command can be improved by the use of the NOERASE option. Normally, the FORMAT command will clear all of the blocks of the minidisk to zeros. When formatting a CMS minidisk with the NOERASE option, the blocks are not cleared, and the performance of the command is improved. The NOERASE option should not be used if previous data on the minidisk should be cleared to zeros.

Remote Minidisks

Remote minidisks are used in command resolution the same way that local minidisks are. The performance of command resolution may be improved if you access remote minidisks lower in your search order than the S-disk. This is of course dependent on the actual commands you are using, and where VM/PC must look to resolve the commands.

For example, if something on your S disk (which is local) is also on a remote disk accessed earlier than S, the system will use the remote disk's copy, thereby degrading performance.

Spool Files

Whenever VM/PC is asked to print a file on the local printer, a spool file is created on disk and given the DOS filespec of SPOOL.nnn, where nnn is a number in the range from 1 to 999. Also, in order to manage the spool files, VM/PC creates two control files, SPOOLIN.DEX and SPOOLSF.BLK.

If you were to erase either of the control files before VM/PC has completed all printing activity, any unprinted spool files would be lost. New control files will, however, be created by VM/PC if local printing is requested and existing control files cannot be found.

Should you ever wish to move spool files from one VM/PC system to another VM/PC system, you will need to move the two control files also, as VM/PC cannot process the spool files without the control files.

Appendix E. ASCII to EBCDIC Conversion

The VM/PC programs use the following table to translate VM/PC CMS EBCDIC format data to ASCII format data whenever necessary.

EBCDIC VALUE HEX	EBCDIC VALUE DEC	=	ASCII VALUE HEX	ASCII VALUE DEC
00	0		00	0
01	1		01	1
02	2		02	2
03	3		03	3
04	4		20	32
05	5		09	9
06	6		20	32
07	7		7F	127
08	8		20	32
09	9		20	32
0A	10		20	32
0B	11		0B	11
0C	12		0C	12
0D	13		0D	13
0E	14		0E	14
0F	15		0F	15
10	16		10	16
11	17		11	17
12	18		12	18
13	19		13	19
14	20		20	32
15	21		20	32
16	22		08	8
17	23		20	32
18	24		18	24
19	25		19	25
1A	26		20	32
1B	27		20	32
1C	28		20	32
1D	29		1D	29
1E	30		1E	30
1F	31		1F	31
20	32		20	32
21	33		20	32
22	34		1C	28
23	35		20	32
24	36		20	32
25	37		0A	10
26	38		17	23
27	39		1B	27
28	40		20	32
29	41		20	32
2A	42		20	32
2B	43		20	32
2C	44		20	32
2D	45		05	5
2E	46		06	6
2F	47		07	7
30	48		20	32
31	49		20	32
32	50		16	22
33	51		20	32
34	52		20	32
35	53		20	32
36	54		20	32
37	55		04	4
38	56		20	32
39	57		20	32
3A	58		20	32
3B	59		20	32
3C	60		14	20
3D	61		15	21
3E	62		00	32
3F	63		1A	26
40	64		20	32
41	65		20	32
42	66		20	32
43	67		20	32
44	68		20	32
45	69		20	32
46	70		20	32
47	71		20	32
48	72		20	32
49	73		20	32
4A	74		60	96
4B	75		2E	46
4C	76		3C	60
4D	77		28	40
4E	78		2B	43
4F	79		7C	124
50	80		26	38
51	81		20	32

EBCDIC VALUE HEX	DEC	=	ASCII VALUE HEX	DEC
52	82		20	32
53	83		20	32
54	84		20	32
55	85		20	32
56	86		20	32
57	87		20	32
58	88		20	32
59	89		20	32
5A	90		21	33
5B	91		24	36
5C	92		2A	42
5D	93		29	41
5E	94		3B	59
5F	95		5E	94
60	96		2D	45
61	97		2F	46
62	98		20	32
63	99		20	32
64	100		20	32
65	101		20	32
66	102		20	32
67	103		20	32
68	104		20	32
69	105		20	32
6A	106		20	32
6B	107		2C	44
6C	108		25	37
6D	109		5F	95
6E	110		3E	62
6F	111		3F	63
70	112		20	32
71	113		20	32
72	114		20	32
73	115		20	32
74	116		20	32
75	117		20	32
76	118		20	32
77	119		20	32
78	120		20	32
79	121		60	96
7A	122		3A	58
7B	123		23	35
7C	124		40	64
7D	125		27	39
7E	126		3D	61
7F	127		22	34
80	128		20	32
81	129		61	97
82	130		62	98
83	131		63	90
84	132		64	100
85	133		65	101
86	134		66	102
87	135		67	103
88	136		68	104
89	137		69	105

EBCDIC VALUE HEX	DEC	=	ASCII VALUE HEX	DEC
8A	138		00	32
8B	139		7B	123
8C	140		F3	243
8D	141		20	32
8E	142		20	32
8F	143		C5	197
90	144		20	32
91	145		6A	106
92	146		6B	107
93	147		6C	108
94	148		6D	109
95	149		6E	110
96	150		6F	110
97	151		70	112
98	152		71	113
99	153		72	114
9A	154		20	32
9B	155		7D	175
9C	156		DB	219
9D	157		20	32
9E	158		F1	241
9F	159		FE	254
A0	160		20	32
A1	161		7E	126
A2	162		73	115
A3	163		74	116
A4	164		75	117
A5	165		76	118
A6	166		77	119
A7	167		78	120
A8	168		79	121
A9	169		7A	122
AA	170		20	32
AB	171		1C	28
AC	172		A9	169
AD	173		5B	91
AE	174		F2	242
AF	175		FA	250
B0	176		20	32
B1	177		20	32
B2	178		20	32
B3	179		20	32
B4	180		20	32
B5	181		20	32
B6	182		20	32
B7	183		20	32
B8	184		20	32
B9	185		20	32
BA	186		20	32
BB	187		D9	217
BC	188		BF	191
BD	189		5D	93
BE	190		BE	190
BF	191		20	32
C0	192		7B	123
C1	193		41	65

EBCDIC VALUE HEX	DEC	=	ASCII VALUE HEX	DEC
C2	194		42	66
C3	195		43	67
C4	196		44	68
C5	197		45	69
C6	198		46	70
C7	199		47	71
C8	200		48	72
C9	201		49	73
CA	202		20	32
CB	203		20	32
CC	204		20	32
CD	205		20	32
CE	206		20	32
CF	207		20	32
D0	208		7D	125
D1	209		4A	74
D2	210		4B	75
D3	211		4C	76
D4	212		4D	77
D5	213		4E	78
D6	214		4F	79
D7	215		50	80
D8	216		51	81
D9	217		52	82
DA	218		20	32
DB	219		20	32
DC	220		20	32
DD	221		20	32
DE	222		20	32
DF	223		20	32
E0	224		5C	92
E1	225		20	32

EBCDIC VALUE HEX	DEC	=	ASCII VALUE HEX	DEC
E2	226		53	83
E3	227		54	84
E4	228		55	85
E5	229		56	86
E6	230		57	87
E7	231		58	88
E8	232		59	89
E9	233		5A	90
EA	234		20	32
EB	235		20	32
EC	236		20	32
ED	237		20	32
EE	238		20	32
EF	239		20	32
F0	240		30	48
F1	241		31	49
F2	242		32	50
F3	243		33	51
F4	244		34	52
F5	245		35	53
F6	246		36	54
F7	247		37	55
F8	248		38	56
F9	249		39	57
FA	250		20	32
FB	251		20	32
FC	252		20	32
FF	253		20	32
FE	254		20	32
FF	255		20	32

The VM/PC programs use the following table to translate ASCII format data to VM/PC CMS EBCDIC format data whenever necessary.

ASCII VALUE HEX	ASCII VALUE DEC	=	EBCDIC VALUE HEX	EBCDIC VALUE DEC
00	0		00	0
01	1		01	1
02	2		02	2
03	3		03	3
04	4		37	55
05	5		2D	45
06	6		2E	46
07	7		2F	47
08	8		16	22
09	9		05	5
0A	10		25	37
0B	11		0B	11
0C	12		0C	12
0D	13		0D	13
0E	14		0E	14
0F	15		0F	15
10	16		10	16
11	17		11	17
12	18		12	18
13	19		13	19
14	20		3C	60
15	21		3D	61
16	22		32	50
17	23		26	38
18	24		18	24
19	25		19	25
1A	26		3F	63
1B	27		27	39
1C	28		22	34
1D	29		1D	29
1E	30		1E	30
1F	31		1F	31
20	32		40	64
21	33		5A	90
22	34		7F	127
23	35		7B	123
24	36		5B	91
25	37		6C	108
26	38		50	80
27	39		7D	125
28	40		4D	77
29	41		5D	93
2A	42		5C	92
2B	43		4E	78
2C	44		6B	107
2D	45		60	96
2E	46		4B	75
2F	47		61	97
30	48		F0	240
31	49		F1	241
32	50		F2	242
33	51		F3	243
34	52		F4	244
35	53		F5	245
36	54		F6	246
37	55		F7	247
38	56		F8	248
39	57		F9	249
3A	58		7A	122
3B	59		5E	94
3C	60		4C	76
3D	61		7E	126
3E	62		6E	110
3F	63		6F	111
40	64		7C	124
41	65		C1	193
42	66		C2	194
43	67		C3	195
44	68		C4	196
45	69		C5	197
46	70		C6	198
47	71		C7	199
48	72		C8	200
49	73		C9	201
4A	74		D1	209
4B	75		D2	210
4C	76		D3	211
4D	77		D4	212
4E	78		D5	213
4F	79		D6	214
50	80		D7	215
51	81		D8	216
52	82		D9	217
53	83		E2	226
54	84		E3	227
55	85		E4	228
56	86		E5	229
57	87		E6	230

ASCII VALUE HEX	DEC	=	EBCDIC VALUE HEX	DEC
58	88		E7	231
59	89		E8	232
5A	90		E9	233
5B	91		AD	173
5C	92		E0	224
5D	93		BD	189
5E	94		5F	95
5F	95		6D	109
60	96		4A	74
61	97		81	129
62	98		82	130
63	99		83	131
64	100		84	132
65	101		85	133
66	102		86	134
67	103		87	135
68	104		88	136
69	105		89	137
6A	106		91	145
6B	107		92	146
6C	108		93	147
6D	109		94	148
6E	110		95	149
6F	111		96	150
70	112		97	151
71	113		98	152
72	114		99	153
73	115		A2	162
74	116		A3	163
75	117		A4	164
76	118		A5	165
77	119		A6	166
78	120		A7	167
79	121		A8	168
7A	122		A9	169
7B	123		C0	192
7C	124		4F	79
7D	125		D0	208
7E	126		A1	161
7F	127		07	7
80	128		00	0
81	129		00	0
82	130		00	0
83	131		00	0
84	132		00	0
85	133		00	0
86	134		00	0
87	135		00	0
88	136		00	0
89	137		00	0
8A	138		00	0
8B	139		00	0
8C	140		00	0
8D	141		00	0
8E	142		00	0
8F	143		00	0

ASCII VALUE HEX	DEC	=	EBCDIC VALUE HEX	DEC
90	144		00	0
91	145		00	0
92	146		00	0
93	147		00	0
94	148		00	0
95	149		00	0
96	150		00	0
97	151		00	0
98	152		00	0
99	153		00	0
9A	154		00	0
9B	155		4A	74
9C	156		00	0
9D	157		00	0
9E	158		00	0
9F	159		00	0
A0	160		00	0
A1	161		00	0
A2	162		00	0
A3	163		00	0
A4	164		00	0
A5	165		00	0
A6	166		00	0
A7	167		00	0
A8	168		00	0
A9	169		AC	172
AA	170		5F	95
AB	171		00	0
AC	172		00	0
AD	173		00	0
AE	174		00	0
AF	175		00	0
B0	176		00	0
B1	177		00	0
B2	178		00	0
B3	179		00	0
B4	180		00	0
B5	181		00	0
B6	182		00	0
B7	183		00	0
B8	184		00	0
B9	185		00	0
BA	186		00	0
BB	187		00	0
BC	188		00	0
BD	189		00	0
BE	190		BE	190
BF	191		BC	188
C0	192		00	0
C1	193		00	0
C2	194		00	0
C3	195		00	0
C4	196		6D	109
C5	197		8F	143
C6	198		00	0
C7	199		00	0

ASCII VALUE HEX	DEC	=	EBCDIC VALUE HEX	DEC
C8	200		00	0
C9	201		00	0
CA	202		00	0
CB	203		00	0
CC	204		00	0
CD	205		00	0
CE	206		00	0
CF	207		00	0
D0	208		00	0
D1	209		00	0
D2	210		00	0
D3	211		00	0
D4	212		00	0
D5	213		00	0
D6	214		00	0
D7	215		00	0
D8	216		00	0
D9	217		BB	187
DA	218		00	0
DB	219		9C	156
DC	220		00	0
DD	221		00	0
DE	222		00	0
DF	223		00	0
E0	224		00	0
E1	225		00	0
E2	226		00	0
E3	227		00	0
E4	228		00	0

ASCII VALUE HEX	DEC	=	EBCDIC VALUE HEX	DEC
E5	229		00	0
E6	230		00	0
E7	231		00	0
E8	232		00	0
E9	233		00	0
EA	234		00	0
EB	235		00	0
EC	236		00	0
ED	237		00	0
EE	238		00	0
EF	239		00	0
F0	240		00	0
F1	241		9E	158
F2	242		AE	174
F3	243		8C	140
F4	244		00	0
F5	245		00	0
F6	246		00	0
F7	247		00	0
F8	248		A1	161
F9	249		00	0
FA	250		AF	175
FB	251		00	0
FC	252		00	0
FF	253		00	0
FE	254		9F	159
FF	255		00	0

Appendix F. Distribution Diskette Contents

All six of the VM/PC Distribution Diskettes are double-sided, double-density and write-protected.

The Personal Computer DOS files shipped on each diskette are shown in the following table:

Diskette		File Name	File Description
1	*	INSTALL.BAT	Batch file that begins installation
1	*	VMPCLOGO.BAS	Installation program that presents logo screen
1	*	VMPCNB.BAT	Batch file that issues VMPC00 message
1	*	VMPC00.MSG	Installation message - No DOS facilities
1	*	VMPC02.BAS	Installation program that issues VMPC02 message
1	*	VMPC02.MSG	Installation message - Found temporary files
1	*	VMPC04.BAS	Installation program that issues VMPC04 message
1	*	VMPC04.MSG	Installation message - Preliminary Check ok
1	*	TVMPCHD.MSG	Installation message header
1	*	TVMPC01.BAS	Installation program that issues TVMPC01 message
1	*	TVMPC01.MSG	Installation message - Completion
1	*	TVMPC03.BAS	Installation program that issues TVMPC03 message
1	*	TVMPC03.MSG	Installation message - Did not complete
1	*	VMPCLC.BAT	Batch file that continues installation
1	*	VMPCLX.EXE	Installation program that lists existing files
1	*	TVMPCDD2.BAS	Program that installs diskette 2 files
1	*	TVMPCDD3.BAS	Program that installs diskette 3 file
1	*	TVMPCDD4.BAS	Program that installs file from diskettes 4, 5 and 6
1	*	VMPCDEL.BAT	Batch file that ends installation
1		CMS.101	VM/PC Remote Server source minidisk
1		CPIO.SYM	Symbol Table used as Service Aid
2		CONFIG.1VM	VM/PC Configuration file
2		VMPCCON.EXE	VM/PC Configurator Program
2		VMPC.COM	VM/PC Program
2		CPIO.1VM	VM/PC Control Program I/O Services
2		CP.1VM	VM/PC Control Program
2		CPMSG.1VM	VM/PC Control Program Message file
2	*	VMPCDEL.BAT	Batch file that ends installation

Figure F-1 (Part 1 of 2). Distribution Diskette Contents

Diskette		File Name	File Description
3		CMS.1VM	VM/PC CMS Program
3		CMS.101	VM/PC Remote Server Installation minidisk
3	*	VMPCDEL.BAT	Batch file that ends installation
4		CMS.100	VM/PC CMS Library minidisk (part 1 of 3)
5		CMS.100	VM/PC CMS Library minidisk (part 2 of 3)
6		CMS.100	VM/PC CMS Library minidisk (part 3 of 3)
6	*	TVMPC.100	Temporary file
6	*	VMPCDEL.BAT	Batch file that ends installation

Figure F-1 (Part 2 of 2). Distribution Diskette Contents

The '*' indicates files used only during installation.

A CMS.101 file appears on both Distribution Diskette 1 and on Distribution Diskette 3. Although they have the same name, they do NOT contain the same data.

The CMS.101 file on Distribution Diskette 1 is a VM/PC CMS minidisk containing several source programs (in packed format) for the Remote Server.

The CMS.101 file on Distribution Diskette 3 is a VM/PC CMS minidisk containing the EXECs and other programs needed to install the Remote Server on the remote system.

Glossary

The following is a list of terms and acronyms used in this book and an explanation of their meaning. Glossaries with further definitions can be found in the Personal Computer Hardware Reference Library publication *BASIC*, and in the Virtual Machine/System Product Library publication *Library Guide and Master Index*, order number GT19-6207.

ASCII. American National Standard Code for Information Interchange. A standard code used for exchanging information among data processing systems and associated equipment. An ASCII file is a text file where the characters are represented in ASCII codes.

backup. Pertaining to a system, device, file, or facility that can be used in case of a malfunction or loss of data.

block. The unit of data physically read/written by VM/PC CMS when performing I/O to a minidisk.

buffer. An area of storage that is used to compensate for a difference in rate of flow of data, or time of occurrence of events, when transferring data from one device to another. Usually refers to an area reserved for I/O operations, into which data is read or from which data is written.

Configurator. A VM/PC program that helps the user specify and edit the parameters and data that are stored on the VM/PC Configuration file.

CMS. Conversational Monitor System. The component of VM/PC that provides the user with an application interface within the Local 3277 Session. CMS functionally resembles the similarly named component of the IBM Virtual Machine/System Product (VM/SP).

CP. Control Program. The component of VM/PC that intercedes between CMS and the hardware system. CP functionally resembles the similarly named component of the IBM Virtual Machine/System Product (VM/SP).

cursor. A moveable marker that indicates a position on the display.

default. A value or option that is assumed when none is specified.

delimiter. A character that groups or separates words or values in a line of input.

DOS. The IBM Personal Computer Disk Operating System (Version 2.00 or its equivalent).

download. The process of transferring a file from a remote computer to a local computer. See also 'Upload'.

EBCDIC. Extended Binary-Coded Decimal Interchange Code. A coded character set consisting of 8-bit coded characters.

edit. To enter, modify, or delete data.

environment. A program, loaded by VM/PC CP, which operates in the 370 virtual machine. CMS is an environment.

export. The process of creating a Personal Computer DOS file from data stored on a CMS file. See also 'import'.

EXEC. A CMS file with a filetype of EXEC. It contains a series of commands that are all executed when you enter the filename of the EXEC file.

file. A collection of related records, treated as a unit. To VM/PC CMS, a file has a unique three-part label. The first part of the label is the 'filename'. The second part is the 'filetype'. And the third part is the 'filemode'.

filemode. The third part of the label of a VM/PC CMS file. It indicates the minidisk on which the file resides.

filename. The first part of the label of a VM/PC CMS file.

filespec. The name of a Personal Computer DOS file in a form acceptable to Personal Computer DOS. Usually of the form d:filename.ext.

filetype. The second part of the label of a VM/PC CMS file. Some filetypes (for example, ASSEMBLE and EXEC) imply that the file has certain characteristics.

function key. One of the ten keys labeled F1 through F10 on the left side of the keyboard. In VM/PC, these are equivalent to PF 1 through PF 10, and PF 11 is the shift key and F1, and PF 12 is the shift key and F2.

import. The process of creating a CMS file from data stored on a Personal Computer DOS file. See also 'export'.

Kilobyte (K). 1024 bytes.

local session. A session running in the 370 virtual machine of your Personal Computer XT/370.

Megabyte (M). 1,048,576 bytes (1024 K).

menu. A list of available operations. You select the operation you want from the list.

minidisk. A Personal Computer DOS file that is managed by VM/PC as a logical disk drive. A minidisk, itself a single Personal Computer DOS file, can contain numerous CMS files.

null entry. A response, to a prompt, consisting only of pressing the ENTER key.

PC DOS. See DOS.

personality file. The Personal Computer DOS file containing the information needed to fully activate the Personal Computer XT/370 hardware.

PF key. See function key.

prompt. A question the computer asks when it needs you to supply information.

queue. A line or list of items waiting for service; the first item that went in the queue is the first item to be serviced.

record. A collection of related information, treated as a unit. For example, in stock control, each invoice might be one record.

remote session. A connection, over either a 3277-compatible coaxial cable or an asynchronous communications line, between a VM/PC user and a separate computer.

scroll. To move all or part of the display image vertically or horizontally so that new data appears at one edge as old data disappears at the opposite edge.

session. A connection between a user and a computer.

spooling. A method of printing files while concurrently continuing with other activities.

upload. The process of transferring a file from a local computer to a remote computer. See also 'download'.

USERID. User identification. The name by which VM/PC knows someone allowed to use the Local 3277 Session. Sometimes referred to as a "logonid."

VMPCSERV. The VM/PC Remote Server Program.

VM/PC. The IBM Virtual Machine/Personal Computer licensed program.

VM/SP. The IBM Virtual Machine / System Product

XEDIT. The VM/PC CMS file editor program. XEDIT functionally resembles the VM/SP System Product Editor.

Index

Special Characters

* command 7-10
#CP command 7-10

A

ACCESS command 7-62
ACCESS command, example 5-49
accessing data on remote system 1-9
accounting routine, sample 6-18
acronyms and terms, glossary of X-1
active session 4-5
active session screen indicator 4-20
activity indicator for session 4-18
AMSERV C-7
application programs on VM/PC C-6
ASCII - definition of X-1
ASCII format files 4-8
ASCII option of IMPORT and EXPORT 5-55
ASCII to binary conversion 4-9
ASCII to EBCDIC value conversion E-4
ASSEMBLE C-7
ASSGN C-7
asterisk in commands 7-4
asynchronous communications adapter 4-8
ATTN command 7-11
autolog id 4-12
availability screen indicator 4-20

B

backing up minidisks D-1
backup - definition of X-1
backup copies of diskettes 2-4
BEGIN command 7-12
binary from ASCII conversion 4-9
block - definition of X-1
book description 1-3
braces in commands 7-3
brackets in commands 7-3
buffer - definition of X-1
buffer, history 5-52
BUFFERS command D-3

C

cents sign character 4-4
CHANGE command 7-12
character set usage 7-56
CHKDSK command 2-19, 3-9
CLEAR key 4-4
CLOSE command 7-14
CMDCALL command 7-66
CMS - definition of X-1
CMS command search order 7-56

CMS
commands 7-58-7-130
CMS diskette 2-9
CMS environment 4-13
CMS file, copying as PC File 7-76
CMS file, copying PC file as 7-94
CMS files, transporting D-1
CMS messages 8-26
CMS.1VM F-2
CMS.1VM file 2-10, 2-17
CMS.100 F-2
CMS.100 file 2-17
CMS.101 F-2
CMSBATCH C-7
coaxial connection, 3277-2 6-1
command name 7-1
command operands 7-1
command options 7-2
command search order, CMS 7-56
commands 7-7
* 7-10
#CP 7-10
abbreviating 7-2
ACCESS 6-1, 7-62
ATTN 7-11
B 5-52
BEGIN 7-12
BOT 5-52
BUFFERS D-3
CHANGE 7-12
CHKDSK 2-19, 3-9
CLOSE 7-14
CMDCALL 7-66
comments in 7-2
COMPARE 7-66
CONWAIT 7-66
COPYFILE 7-67
CP 7-16, 7-67
DEBUG 7-68
DEFAULTS 7-69
DEFINE 7-16
DESBUF 7-71
DETACH 7-18
DISPLAY 7-19
DROPBUF 7-71
DUMP 7-23
Entering 7-1
ERASE 7-72
ESTATE/ESTATEW 7-7
EXEC 7-74
EXECIO 7-74
EXPORT 5-55, 7-76, D-1
EXTERNAL 7-26
F 5-52
FILEDEF 7-78
FILELIST 7-78
FINIS 7-79
FORMAT 3-16, 7-80
GENDIRT 7-84
GENMOD 7-84
GLOBAL 7-89
GLOBALV 7-90
HT 7-129
HX 7-129
IDENTIFY 7-93
IMPORT 5-55, 7-94, D-1
INCLUDE 7-97
INSTALL 6-7
IPL 7-26
L 5-52
LINK 6-1, 6-10, 7-27
LISTFILE 7-98
LOAD 7-99
LOADMOD 7-99
LOGOFF 7-30
MACLIB 7-100
MAKEBUF 7-101

MODMAP 7-102
not in VM/PC C-7
NUCXDROP 7-102
NUCXLOAD 7-102
NUCXMAP 7-104
ORDER 7-30
PRINT 5-38, 6-2, 7-104
PURGE 5-38, 7-31
QUERY 6-10, 7-32, 7-105
R 5-52
RELEASE 7-114
RENAME 7-114
REQUEST 7-37
RESTORE 2-11, 2-19
RETRIEVE 5-52
RT 7-130
scroll 4-15
scrolling 5-51
SENTRIES 7-114
SET 5-51, 7-38, 7-115, 8-2
SORT 5-24, 7-118
SPOOL 6-2, 7-41
START 7-119
STATE/STATEW 7-119
STORE 7-44
SYNONYM 7-119
TAG 7-46
TERMINAL 5-54, 7-48
TOP 5-52
TRACE 7-51
TXTLIB 7-126
TYPE 7-126
unsupported VM/SP C-7
UPDATE 7-126
VMPC 4-1
VMPCCON 3-3
VMPCSERV 6-8
working with 5-1
XEDIT 7-127
commands, PC DOS 2-19
comments in CMS Commands 7-2
COMPARE command 7-66
comparison of VM/PC and VM/SP C-1
concurrent session 5-47
CONFIG.UT1 3-3, 3-19
CONFIG.1VM F-2
CONFIG.1VM file 2-8, 2-17, 3-3
configuration 3-1
configuration disk allocation and checking 3-18
configuration limitations 3-19
configurator 3-2
Configurator - definition of X-1
configurator password menu 3-10
configurator screens 3-4
control program diskette 2-7
controlling cursor direction 4-19
CONWAIT command 7-66
COPYFILE command 7-67
copying files to or from remote system 1-9
CP - definition of X-1
CP command 7-16, 7-67
CP command summary 7-7
CP commands 7-7-7-55
CP Read indicator 4-17

CP.1VM F-2
CP.1VM file 2-8, 2-17
CPIO.1VM F-2
CPIO.1VM file 2-8, 2-17
CPI0.SYM F-1
CPMSG.1VM F-2
CPMSG.1VM file 2-8, 2-17
cursor - definition of X-1
cursor direction screen indicator 4-19
cursor direction, controlling 4-19

D

DDR C-7
DEBUG command 7-68
default - definition of X-1
DEFAULTS command 7-69
DEFINE command 7-16
delimiter - definition of X-1
DESBUF command 7-71
DETACH command 7-18
DIAGNOSE instruction A-3
differences between VM/PC and VM/SP C-1
  BLIP C-6
  CMS Maclibs C-4
  CMS/DOS C-1
  commands C-7
  cpcomm C-1
  EXECIO C-5
  file system C-4
  FILELIST C-6
  I/O interfaces C-4
  IDENTIFY command C-3
  initialization C-2
  interval timer C-4
  logon screen C-1
  messages C-3
  NUCXLOAD command C-5
  remote files C-4
  RSCS C-3
  screen format C-3
  storage keys C-4
  SVCTRACE C-6
  SYSPROF EXEC C-2
  tape I/O C-1
  TERMINAL command C-3
  virtual punches C-1
  virtual readers C-1
  XEDIT C-5
  Y disk C-2
directories, tree-structured D-3
DISK C-7
diskette contents, distribution F-1
diskettes
  CMS 2-9
  control program 2-7
  installation 2-1
  library 2-10
diskettes for installation 2-1
DISPAX TEXT file 6-7
DISPIO TEXT file 6-7
DISPLAY command 7-19
display, controlling 5-54
distribution diskette contents F-1
DLBL C-7
DMS messages 8-26

DOS - definition of X-2
DOS Features D-3
DOS features, using D-3
DOS PRINT command queue D-3
DOSLIB C-7
DOSLKED C-7
download 5-47
download - definition of X-2
downloaded files, logging 6-18
downloading - definition 6-13
DROPBUF command 7-71
DSERV C-7
DUMP command 7-23

E

EBCDIC - definition of X-2
EBCDIC to ASCII value conversion E-1
EDIT C-7
edit - definition of X-2
ellipsis in commands 7-3
ellipsis to indicate more to be seen 5-54
emsg on command 8-2
ENTER key 4-4, 5-4
Entering Commands 7-1
environment 4-13
environment - definition of X-2
EOL option of IMPORT and EXPORT 5-55
ERASE command 7-72
ERASE EOF key 4-4
ERASE INPUT key 4-4
ESERV C-7
ESTATE/ESTATEW command 7-72
EXEC - definition of X-2
EXEC command 7-74
EXEC, profile 5-50
EXECIO command 7-74
EXECs 5-40
existing files list 2-17
export - definition of X-2
EXPORT command 7-76, D-1
EXPORT command, example 5-55
EXPORT.CMS file 7-77
EXTERNAL command 7-26

F

FB-512 6-10
features of VM/PC 1-1
FETCH C-7
file - definition of X-2
file access performance D-2
file conversion program 4-9
FILECONV.EXE file 4-9
FILEDEF command 7-78
FILELIST command 7-78
filemode - definition of X-2
filename - definition of X-2
files, temporary 2-19
files, VM/PC 1-1
    CMS.1VM 2-10
    CMS.1VM file 2-17
    CMS.100 2-17

CONFIG.UT1 3-3, 3-19
CONFIG.1VM 2-8, 2-17, 3-3
configuration 3-1
CP.1VM 2-8, 2-17
CPIO.1VM 2-8, 2-17
CPMSG.1VM 2-8, 2-17
CPS0.SYM F-1
DISPAX TEXT 6-7
DISPIO TEXT 6-7
distribution
diskette F-1
existing list 2-17
FILECONV.EXE 4-9
GENPCSER
EXEC 6-7
HEXIN 6-4
HEXIN
ASSEMBLE 6-7
HSTCMS TEXT 6-7
HSTCOM TEXT 6-7
HSTFSR TEXT 6-7
HSTGMEAC
TEXT 6-7
HSTGMELN
TEXT 6-7
HSTHYPE TEXT 6-7
HSTLOGO
TEXT 6-7
HSTSPL TEXT 6-7
HSTVSI TEXT 6-7
IMLPU.1VM 2-2
personality 2-2
remote 6-1
SETUP.EXE 4-9
SPOOL.nnn D-4
SPOOLIN.DEX D-4
SPOOLSF.BLK D-4
temporary 2-19
TERMINAL.EXE 4-9
TVMPC.DSK 2-19
TVMPC.100 2-19
TVMPCDD2.BAS 2-19
TVMPCDD3.BAS 2-19
TVMPCDD4.BAS 2-19
TVMPCHD.MSG 2-19
TVMPCLC.BAT 2-19
TVMPC01.BAS 2-19
TVMPC01.MSG 2-19
TVMPC03.BAS 2-19
TVMPC03.MSG 2-19
VMPC.COM 2-8, 2-17
VMPCCON.EXE 2-8, 2-17, 3-3
VMPCSERV
EXEC 6-7
VMPCSERV
MODULE 6-7
filespec - definition of X-2
filetype - definition of X-2
final allocation and checking 3-18
finding information in VM/SP library C-8
FINIS command 7-79
FORMAT
command 3-16, 7-80
FORMAT command performance D-3
format of VM/PC screen 4-15
formatting a minidisk 3-16
function key - definition of X-2
function selection menu 3-6

G

GENDIRT
command 7-84
GENMOD
command 7-84
GENPCSER EXEC
file 6-7
GLOBAL command 7-89
GLOBALV
command 7-90
glossary of terms and
acronyms X-1

H

HELP C-7
HEXIN 6-4
HEXIN ASSEMBLE
file 6-7
HEXIN module 6-7
highlighting input 5-54
history buffer 4-15, 5-52
Holding indicator 4-17
how you can use
VM/PC 1-7
HSTACC TEXT
routine 6-18
HSTCMS TEXT file 6-7
HSTCOM TEXT file 6-7
HSTFSR module 6-18
HSTFSR TEXT file 6-7
HSTGMEAC TEXT
file 6-7
HSTGMELN TEXT
file 6-7
HSTHYPE TEXT file 6-7
HSTLOGO TEXT
file 6-7
HSTSPL TEXT file 6-7
HSTVSI TEXT file 6-7
HT command 7-129
HX command 7-129

I

IDENTIFY
command 7-93
if something goes
wrong 4-21
IMLPU.1VM file 2-2
immediate
commands 7-128
import - definition of X-2
IMPORT command 7-94,
D-1
IMPORT command,
example 5-55
improving performance,
hints for D-1
INCLUDE
command 7-97
indicators on 25th line of
screen 4-18
information in VM/SP
library C-8
inhibit screen
indicator 4-20
insert cursor of SET
command, example 5-51
Insert key 5-11
insert screen
indicator 4-20
INSTALL EXEC 6-7
installing VM/PC 2-1
backup diskette 2-4
C disk 2-2
D disk 2-3
diskettes 2-1
erasing temporary
files 2-23
existing files 2-17

failure 2-21
files 2-8
insufficient space 2-20
process 2-5
repeating 2-23
requirements 2-1
saving files 2-7
screens 2-6
skipping diskette 2-21
temporary files 2-19, 2-23
termination 2-8
installing VMPCSERV 6-3
insufficient space to install 2-20
Introduction to VM/PC 1-1
invoking VM/PC 4-1
IPL command 7-26

K

key values E-1
keyboard usage 4-2
keys, description 4-4
Kilobyte (K) - definition of X-2

L

LABELDEF C-7
library diskettes 2-10
limitations on configuration 3-19
lines, displaying long 5-54
LINK command 7-27
LINK command, example 5-49
link list menu 3-16
linking to virtual machines 5-48
LISTDS C-7
LISTFILE command 7-98
LISTIO C-7
LKED C-7
LOAD command 7-99
LOADLIB C-7
LOADMOD command 7-99
local disks, identifying 6-10
local session 1-8
local session - definition of X-2
LOCAL/REMOTE keyword of LINK 6-10
locked uppercase screen indicator 4-18
logging downloaded files 6-18
logical not character 4-4
Logo Screen, VMPCSERV 6-9
LOGOFF command 7-30
logon process 4-12
logon screen 4-13
Logon Screen, VM/PC 4-12
LRECL option of IMPORT 5-56

M

M indicator on 25th line 4-18
MACLIB command 7-100
MAKEBUF command 7-101

Megabyte (M) - definition of X-3
menus
configurator password 3-10
configurator screens 3-4
definition of X-3
function menu 3-6
link list 3-16
logon 4-13
minidisk list 3-13
session selection 4-5
system configuration 3-8
system options 3-10
update link 3-17
user environment 3-12
user function selection 3-12
user selection 3-11
3101 Emulation Program Selection 4-9
message numbers, setting on 8-2
minidisk - definition of X-3
minidisk list menu 3-13
minidisks, backing up D-1
minidisks, remote D-4
MODMAP command 7-102
More indicator 4-17
MOVEFILE C-7
moving application programs from VM/SP to VM/PC C-6
moving files to or from remote system 1-9

N

NAMEFIND C-7
NAMES C-7
new line key 4-4
NOASCII option of IMPORT and EXPORT 5-57
NOEOL option of IMPORT and EXPORT 5-57
Not Accepted indicator 4-17
notational conventions 7-2
NOTE C-7
NUCXDROP command 7-102
NUCXLOAD command 7-102
NUCXMAP command 7-104
null entry - definition of X-3
nulls, setting on 4-20
numbers, getting message 8-2
numeric screen indicator (N) 4-19

O

OPTION C-7
ORDER command 7-30
order numbers for VM/SP Release 2 Books C-9
OSRUN C-7

P

Page Dataset 3-9
password 4-12
PA2 key for scrolling 5-54
PC DOS - definition of X-3
PC file, copying as CMS File 7-94
PC file, copying CMS file as 7-76
PEEK C-7
performance tuning D-3
performance, file access D-2
performance, FORMAT command D-3
performance, hints for improving D-1
personality file 2-2
personality file - definition of X-3
PF key - definition of X-3
Primer, VM/PC 1-3
PRINT command 5-38, 7-104
printers, remote 6-1
printing a screen 5-38
printing files 5-38
printing on remote system 1-9
processing failures 4-23
PROFILE EXEC 5-48, 5-50
programs, application, on VM/PC C-6
prompt - definition of X-3
prt sc key to print a screen 5-38
PSERV C-7
PUNCH C-7
PURGE command 5-38, 7-31

Q

QUERY command 7-32, 7-105
queue - definition of X-3
quitting the configurator 3-7

R

RDR C-7
RDRLIST C-7
read-only mode 5-49
read-write mode 5-49
READCARD C-7
reading plan 1-3
RECEIVE C-7
record - definition of X-3
RELEASE command 7-114
RELEASE command, example 5-50
remote disks, identifying 6-10
remote files 6-1
remote minidisks 6-1, D-4
remote printers 6-1
remote server program 6-3
remote server with 3101 session 4-10
remote session - definition of X-3
remote system, accessing data on 1-9
Remote 3101 Session 4-7

REMOTE/LOCAL
keyword of LINK 6-10
RENAME
command 7-114
REP option of IMPORT
and EXPORT 5-57
repeating installation 2-23
REQUEST
command 7-37
RESET key 4-4
RESTORE
command 2-11, 2-19
RETRIEVE
command 5-52
RSERV C-7
RS232 interface 4-7
RT command 7-130
RUN C-7
Running indicator 4-17

S

sample profile
EXEC 5-50
SAMPLE virtual
machine 5-49
screen format 4-15
screen indicators, 25th
line 4-18
screen scroll facility 4-15
scroll - definition of X-3
scroll facility 4-15
Scroll screen area 4-15
scrolling commands 5-51
scrolling with PA2
key 5-54
search order, CMS
command 7-56
SELECT key 4-4
SENDFILE C-7
SENTRIES
command 7-114
session selection
menu 4-5
sessions
active 4-5
activity indicator 4-18
asynchronous 4-6
changing 4-6
concurrent 1-9, 4-5,
5-47, 6-1
definition of X-3
list of 4-6
local 1-8
Remote 3101
Session 4-7
selection 4-5
370 Processor
Control 4-6
SET command 7-38,
7-115
SETPRT C-7
SETUP.EXE file 4-9
SORT command, example
of use 5-24
SORT command,
format 7-118
SPOOL command 7-41
Spool Files D-4
SPOOL.nnn D-4
SPOOLIN.DEX D-4
spooling - definition
of X-3
SPOOLSF.BLK D-4
SSERV C-7
START command 7-119
starting VM/PC 4-1
startup failure 4-22
STATE/STATEW
command 7-119
STATUS screen
area 4-15

STORE command 7-44
SVCTRACE C-7
SYNONYM
command 7-119
system configuration
menu 3-8
system failure 4-21
system options
screen 3-10

T

TAG command 7-46
TAPE C-7
TAPEMAC C-7
TAPPDS C-7
TELL C-7
temporary files 2-19
TERMINAL
command 7-48
terminal emulation
program 4-9
terminal value
specification
program 4-9
TERMINAL.EXE
file 4-9
terminating
installation 2-8
terms and acronyms,
glossary of X-1
TEST REQ key 4-4
TRACE command 7-51
tracking downloaded
files 6-18
transferring PC DOS and
VM/PC Files 5-55
translating CMS file to PC
file 7-76
translating PC file to CMS
file 7-94
transporting CMS
files D-1
transporting files between
VM/PC systems D-1
tree-structured
directories D-3
TVMPC.DSK file 2-19
TVMPC.100 2-19, F-2
TVMPCDD2.BAS F-2
TVMPCDD2.BAS
file 2-19
TVMPCDD3.BAS F-2
TVMPCDD3.BAS
file 2-19
TVMPCDD4.BAS F-2
TVMPCDD4.BAS
file 2-19
TVMPCHD.MSG F-2
TVMPCHD.MSG
file 2-19
TVMPCLC.BAT
file 2-19
TVMPC01.BAS F-2
TVMPC01.BAS file 2-19
TVMPC01.MSG F-2
TVMPC01.MSG file 2-19
TVMPC03.BAS F-2
TVMPC03.BAS file 2-19
TVMPC03.MSG F-2
TVMPC03.MSG file 2-19
TXTLIB command 7-126
TYPE command 7-126

U

underscores in
commands 7-3
unsupported VM/SP
commands C-7
UPDATE
command 7-126

update link menu 3-17
upload 5-47
upload - definition of X-3
uploading - definition 6-13
uppercase screen indicators (S and C) 4-18
user environment menu 3-12
user function selection menu 3-12
user selection menu 3-11
userid 4-12
USERID - definition of X-3
using the keyboard 4-2

V

vertical bar character 4-4
VM Read indicator 4-17
VM/PC 1-1
  as local session 1-7
  as 3101 1-7
  as 3277 1-7
  components needed 1-2
  concurrent session 1-7
  configuration 3-1
  configurator 3-2
  definition of X-3
  error recovery 4-21
  features 1-1
  hardware 1-2
  installing 2-1
  invoking 4-1
  keyboard 4-2
  logging on to 4-12
  password 4-12
  remote session 1-7
  screen format 4-15
  sessions 1-7
  starting 4-1
  types of connection 1-7
  userid 4-12
  what you need 1-2
VM/PC and VM/SP comparison C-1
VM/PC Logon Screen 4-12
VM/PC Primer 1-3
VM/SP - definition of X-4
VM/SP commands not supported in VM/PC C-7
VM/SP library, finding information in C-8
VM/SP Release 2 Books, order numbers C-9
VMPC command 4-1
VMPC.COM F-2
VMPC.COM file 2-8, 2-17
VMPCCON command 3-3
VMPCCON.EXE F-2
VMPCCON.EXE file 2-8, 2-17, 3-3
VMPCDEL.BAT F-2
VMPCLC.BAT F-2
VMPCLOGO.BAS F-2
VMPCLX.EXE F-2
VMPCNB.BAT F-2
VMPCSERV 1-9, 6-3
VMPCSERV - definition of X-3
VMPCSERV EXEC file 6-7
VMPCSERV Logo Screen 6-9

VMPCSERV MODULE
file 6-7
VMPCSERV,
installing 6-3
VMPC00.MSG F-2
VMPC02.BAS F-2
VMPC02.MSG F-2
VMPC04.BAS F-2
VMPC04.MSG F-2

W

Waiting indicator 4-17
what to do if something
goes wrong 4-21
what you need for
VM/PC 1-2
WRAP operand of
TERMINAL
command 5-54
writing EXECs 5-40

X

X screen indicator 4-20
XEDIT - definition
of X-4
XEDIT command 7-127

2

24th line 4-15
25th line in 3101
session 4-10
25th line screen
indicators 4-18

3

3101 1-7
3101 Emulation Program
Selection Menu 4-9
3101 Session 4-7
3101 session, line 25
in 4-10
3101 with remote
server 4-10
3277 1-7
3277-2 coaxial
connection 6-1
370 PROCESSOR
CONTROL
SESSION B-1

The Personal Computer
Software Library

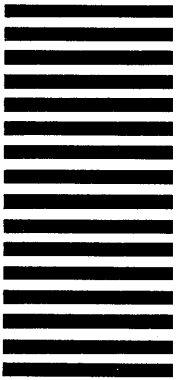
Product Comment Form

VM/PC User's Guide **6361530**

Your comments assist us in improving our products. IBM may use and distribute any of the information you supply in anyway it believes appropriate without incurring any obligation whatever. You may, of course, continue to use the information you supply.

Comments:

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATE



BUSINESS REPLY MAIL

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

POSTAGE WILL BE PAID BY ADDRESSEE:

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

Fold here

Please do not staple

designed. If you believe that there is a defect in a warranted program such that it does not meet its specification, you must notify IBM within the warranty period and in the manner set forth in the program specifications.

ALL OTHER PROGRAMS ARE PROVIDED 'AS IS' WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU (AND NOT IBM OR AN IBM AUTHORIZED REPRESENTATIVE) ASSUME THE ENTIRE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

IBM does not warrant that the functions contained in any program will meet your requirements or that the operation of the program will be uninterrupted or error free or that program defects will be corrected.

THE FOREGOING WARRANTIES ARE IN LIEU OF ALL OTHER WARRANTIES, EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

SOME STATES DO NOT ALLOW THE EXCLUSION OF IMPLIED WARRANTIES, SO THE ABOVE EXCLUSION MAY NOT APPLY TO YOU. THIS WARRANTY GIVES YOU SPECIFIC LEGAL RIGHTS AND YOU MAY ALSO HAVE OTHER RIGHTS WHICH VARY FROM STATE TO STATE.

LIMITATION OF REMEDIES

IBM's entire liability and your exclusive remedy shall be as follows:

1. With respect to defective media during the warranty period:
   a. IBM will replace media not meeting IBM's 'Limited Warranty' if returned to IBM or an IBM authorized representative with a copy of your receipt.
   b. In the alternative, if IBM or such IBM authorized representative is unable to deliver replacement media free of defects in materials and workmanship, you may terminate this Agreement by returning the program and your money will be refunded.
2. With respect to warranted programs, in all situations involving performance or nonperformance during the warranty period, your remedy is (a) the correction or bypass by IBM of program defects, or (b) if, after repeated efforts, IBM is unable to make the program operate as warranted, you shall be entitled to a refund of the money paid or to recover actual damages to the limits set forth below.

IN NO EVENT WILL IBM BE LIABLE TO YOU FOR ANY LOST PROFITS, LOST SAVINGS OR OTHER INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OF OR INABILITY TO USE ANY PROGRAM EVEN IF IBM OR AN IBM AUTHORIZED REPRESENTATIVE HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES, OR FOR ANY CLAIM BY ANY OTHER PARTY.

SOME STATES DO NOT ALLOW THE LIMITATION OR EXCLUSION OF LIABILITY FOR INCIDENTAL OR CONSEQUENTIAL DAMAGES SO THE ABOVE LIMITATION OR EXCLUSION MAY NOT APPLY TO YOU.

IBM's liability to you for actual damages for any cause whatsoever, and regardless of the form of action, shall be limited to the greater of $5,000 or the money paid for the program that caused the damages or that is the subject matter of, or is directly related to, the cause of action.

SERVICE

Service from IBM, if any, will be described in program specifications or in the statement of service, supplied with the program, if there are no program specifications.

IBM may also offer separate services under separate agreement for a fee.

GENERAL

Any attempt to sublicense, rent or lease, or, except as expressly provided for in this Agreement, to transfer any of the rights, duties or obligations hereunder is void.

This Agreement will be construed under the Uniform Commercial Code of the State of New York.

International Business Machines Corporation

P.O. Box 6
Endicott, New York 13760

6361530
Printed in U.S.A.