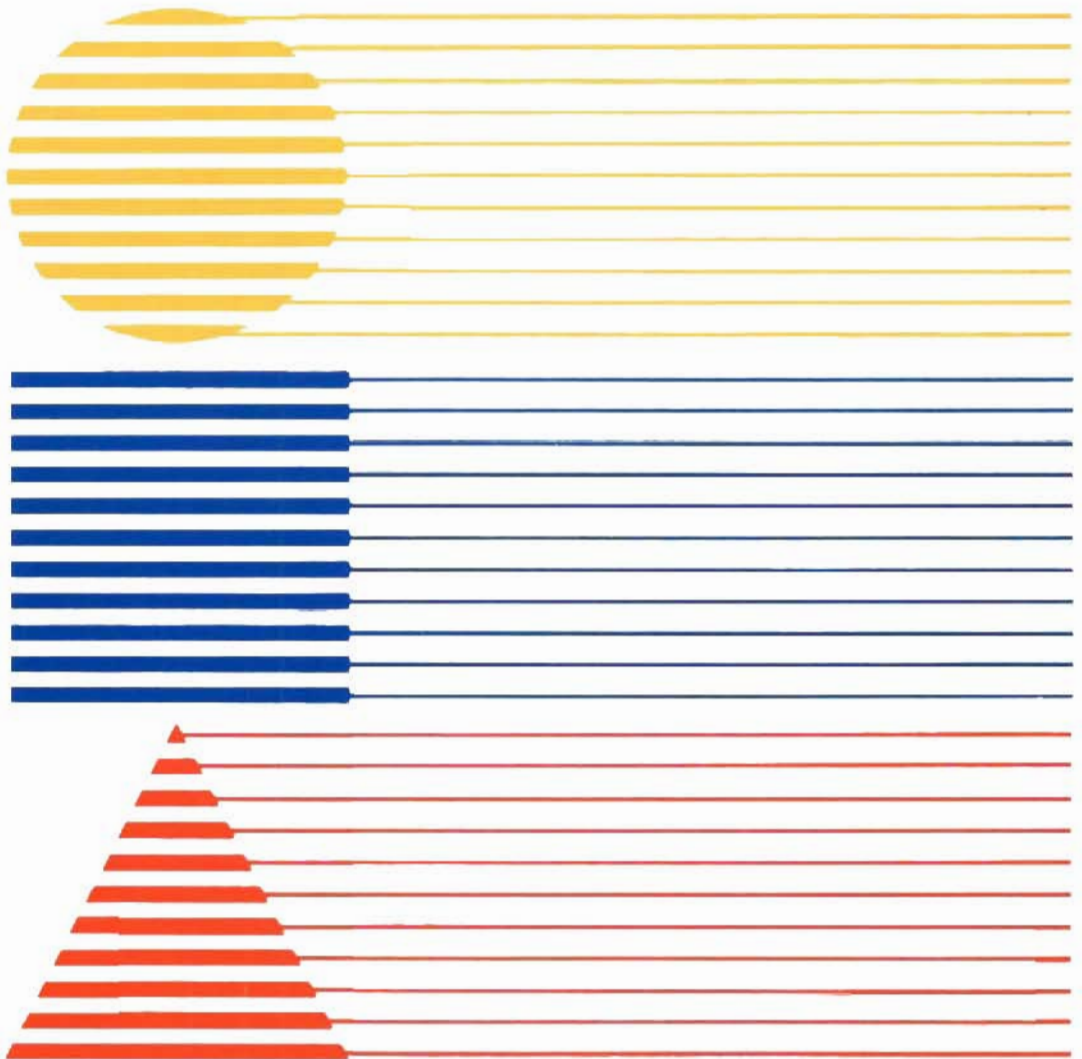




VM/System Product **CMS Primer**

Release 5

SC24-5236-3



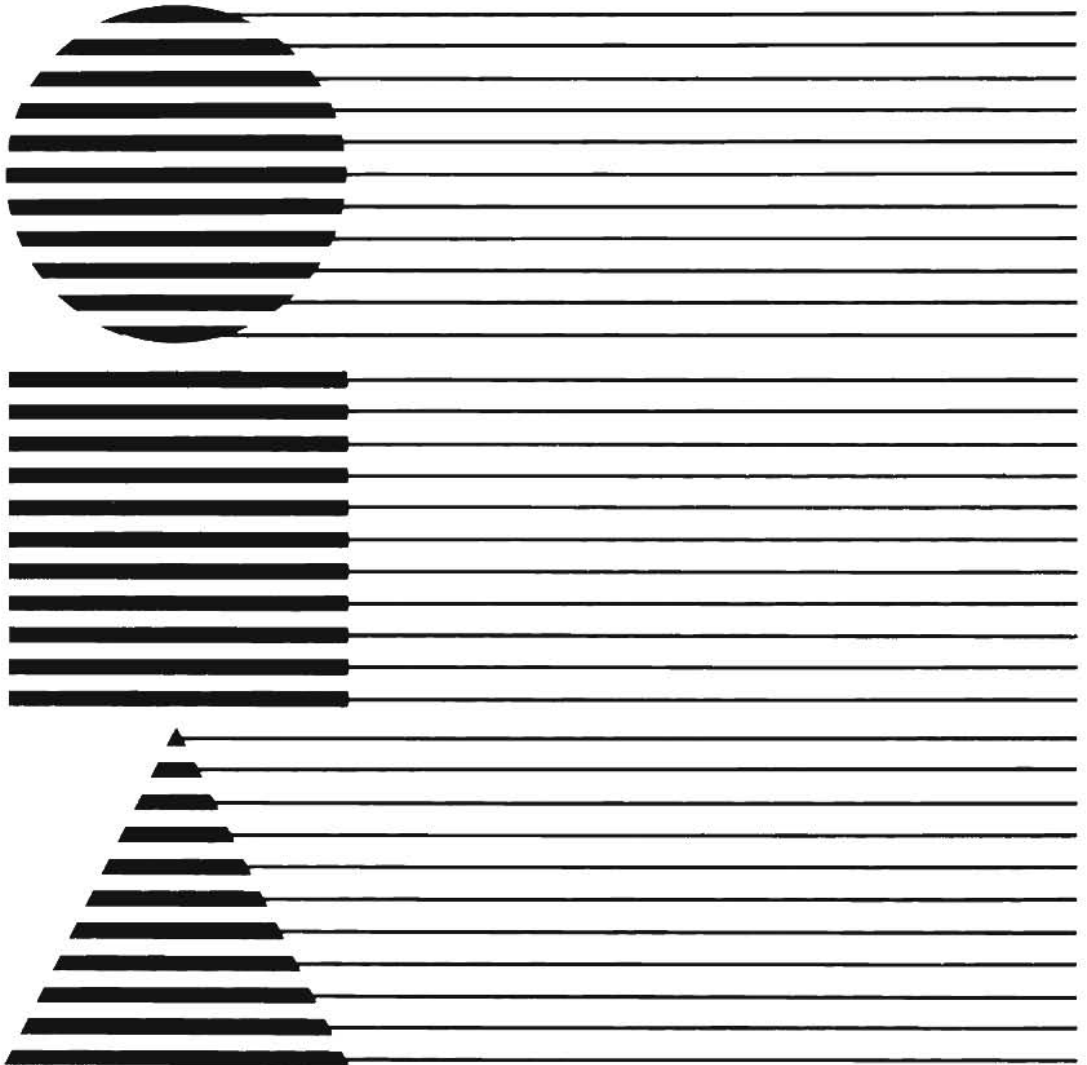




VM/System Product CMS Primer

Release 5

SC24-5236-3



Fourth Edition (December 1986)

This edition, SC24-5236-3, is a major revision of SC24-5236-2, and applies to Release 5 of the Virtual Machine/System Product (VM/SP), Program Number 5664-167, and to all subsequent releases of this product until otherwise indicated in new editions or Technical Newsletters. Changes are made periodically to the information herein; before using this publication in connection with the operation of IBM systems, consult the latest *IBM System/370, 30xx, and 4300 Processors Bibliography*, GC20-0001, for the editions that are applicable and current.

Summary of Changes

For a list of changes, see "Summary of Changes" on page 263.

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

In this manual are illustrations in which names are used. These names are fanciful and fictitious, created by the author; they are used solely for illustrative purposes and not for identification of any person or company.

Ordering Publications

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

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

The form for reader's comments provided at the back of this publication may also be used to comment on the VM/SP online HELP facility.

Preface

The objective of this primer is to teach you, a new user, how to do your work using the Virtual Machine/System Product (VM/SP) system and a video display terminal. We have attempted to present the material in a way that will interest you, enable you to be successful, and sometimes even make you smile. But the best way to learn the system is to use it. To this end, we suggest that you do the exercises, which are woven throughout the text.

Note: The examples in this book assume you are using an 80-column, IBM 3270-compatible terminal. If you have a "line mode" terminal, you should read the *VM/SP CMS Primer for Line-Oriented Terminals*, SC24-5242, instead of this book. If you do not know which type of terminal you have, ask your system administrator.

This book has ten chapters:



Chapter 1. Getting Started gets you acquainted with your terminal and teaches you how to sign on to the system.



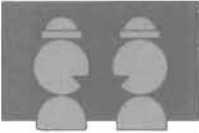
Chapter 2. Editing Files teaches you how to create and change files.



Chapter 3. Managing your Files shows you how to discard, rename, compare, and copy your own files. It also shows you how to use another person's files.



Chapter 4. Printing Files tells you how to get printed copies of your files.



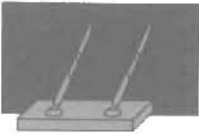
Chapter 5. Communicating With Other People (Who Use Computers) teaches you how to send messages, notes, and files to other computer users, and how to receive notes and files that other people send to you.



Chapter 6. Asking Questions shows you how you can find out useful information from your system, like the time of day and who else is using the computer system.



Chapter 7. Using Script to Format Your Files shows you how to use formatting instructions that make printed copies of your files look attractive. It is based on an IBM licensed program product that runs under the control of VM/SP.



Chapter 8. Creating and Using EXECs teaches you how to simplify routine tasks by creating sequences of general-purpose commands.



Chapter 9. Creating and Running Programs shows you how to create, compile, and run programs written in various programming languages.



Chapter 10. Looking at VM/SP Through Windows gets you acquainted with full-screen CMS and some of its features for managing several pieces of information on your screen at the same time.

At the end of the book, you will find a card that summarizes all the commands presented in the text. Additional copies of the card can be ordered separately as *VM/System Product CMS Primer Summary of Commands*, SX24-5151.

Contents

Chapter 1. Getting Started	1
Getting Acquainted with Your Terminal	2
The Keyboard	3
Correcting Typing Mistakes	6
Logging On and Logging Off	9
If You Make a Mistake in Your Userid or Password	10
You've Successfully Entered Your Userid and Password	12
Status Notices	14
Chapter 2. Editing Files	17
Creating Files	18
What Is a File?	18
Creating Your First File	20
Creating a Second File	27
Splitting and Joining Lines	30
Setting Tabs	32
Displaying PF Key Settings	35
Inserting Words in a Line	36
Summary	39
Changing Files (Part One)	40
Displaying a File on Your Screen	40
I Don't Remember the File Label	40
Adding Lines	42
Moving through a File	44
More Practice in Adding Lines	49
Deleting Lines	51
Duplicating Lines	54
Copying Lines	55
Moving Lines	57
RESET Command	58
Summary	58
Changing Files (Part Two)	60
LOCATE and CHANGE Commands	60
The = (equal) Command	72

Summary	74
Changing Files (Part Three)	75
Inserting Data from Another File	75
I Don't Want to Keep the Changes!	78
SET NUMBER ON Command	80
Sorting a File	81
Getting Help	87
Summary	89
Protecting Your Files	90
SAVE Command	90
SET AUTOSAVE Command	91
Summary	94
More About Editing	95
Power Typing	95
Splitting the Screen	98
Creating an XEDIT Profile	101
Summary	103
Chapter 3. Managing Your Files	105
What Files Do I Have?	106
Finding Files in the List	109
Scrolling through the List	109
Rearranging the List	109
Using the LOCATE Command	110
Variations on the FILELIST Command	110
Editing a File from the List	114
Discarding a File	115
Renaming a File	117
Copying a File	119
Comparing Two Files	120
Using Other People's Files	122
Summary	128
Chapter 4. Printing Files	129
Where Do I Enter These Commands?	130
PRINT Command	130
Using PRINT on the FILELIST Screen	131
SCRIPT Command	133
Using SCRIPT on the Filelist Screen	134
Summary	136

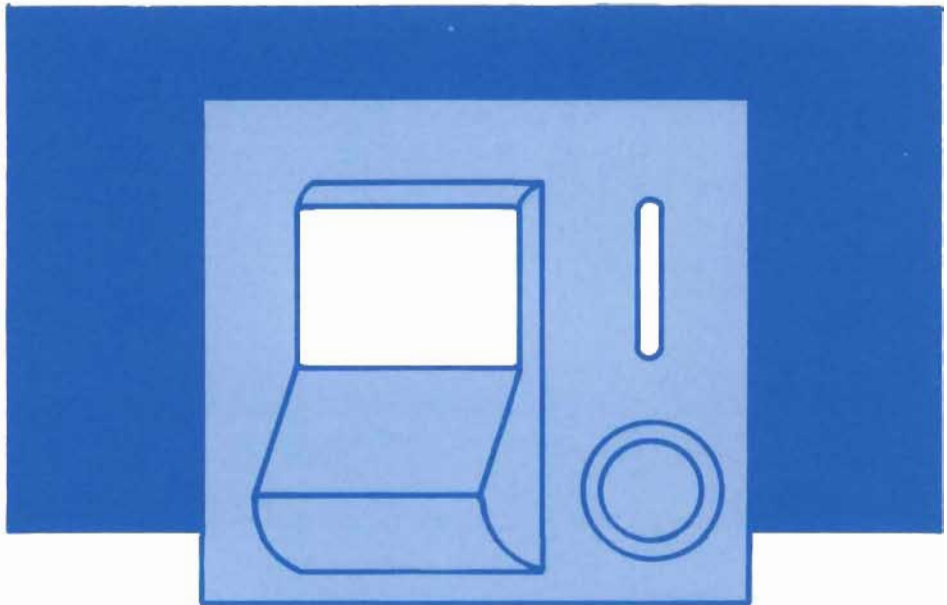
Chapter 5. Communicating With Other People (Who Use Computers) . . .	137
Keeping Track of Names	138
What Is a Names File?	139
How to Create a Names File	139
Keeping Your Names File up to Date	146
Summary of PF Keys on the Names Screen	148
Creating Your Own Names File	149
Sending and Receiving Messages	150
What Is a Message?	150
TELL Command	150
I Don't Want to Receive Messages	151
For Practice	152
Sending Notes	152
What Is a Note?	152
NOTE Command	153
Sending Notes to People on a List	156
Sending Complimentary Copies	158
Keeping Copies of Notes	159
Continuing Notes	159
Summary of PF Keys on the NOTE Screen	160
Summary of How to Send a Note	161
Sending Files	162
SENDFILE Command	162
Sending One File	163
Choosing the Options	164
Sending More than One File	165
For Practice	171
Receiving Notes and Files	171
RDRLIST Command	172
What Should I Do with the Files in My Reader?	173
Using PF Keys to Scroll through or Rearrange the List	177
Is There a File in Your Reader?	178
Summary	178
Chapter 6. Asking Questions	179
Where Do I Enter These Commands?	180
Who Am I, Where Am I, What's the Date and Time?	181
Who's Using the Computer?	181

How Much Disk Space Have I Used?	182
Which Command Should I Use?	183
Summary	186
Chapter 7. Using Script to Format Your Files	187
Learning Markup	188
What Kind of Files Can I Use?	188
Paragraphs	190
Headings	190
EXAMPLE OF A HEAD 2	192
Example of a Head 3	192
Exercise: Paragraphs and Headings	193
Lists	196
Exercise: Lists	200
Definition Lists	201
Exercise: Definition List	202
Highlighting	203
Exercise: Highlighting	204
Checking Your Spelling	205
The Ezgadget Newsletter	205
Summary of GML Tags	207
Chapter 8. Creating and Using EXECs	209
An EXEC to Print a File	210
An EXEC to Print Script Files	214
When Should I Use Quotation Marks?	216
An EXEC to Link to Someone Else's Disk	216
PROFILE EXEC	218
Chapter 9. Creating and Running Programs	219
Creating a Program	220
Compiling a Program	222
Running a Program	225
Using IBM BASIC	227
Creating a BASIC Program	227
Compiling a BASIC Program	229
Running a BASIC Program	230
Summary	231
Chapter 10. Looking at VM/SP Through Windows	233
Starting Full-Screen CMS	235

Working with Windows	239
Making the MESSAGE Window Appear	241
Using the PA2 Key to Scroll	245
Popping a Window	246
Using Border Commands	247
WARNING and NETWORK Windows	254
Using the WM Window	255
Logging Your Messages and Warnings	257
Using the CMS PF Keys	258
Setting a CMS PF Key to TELL	260
Leaving Full-Screen CMS	261
Summary	261
Summary of Changes	263
Bibliography	267
Index	269
 Summary of Commands (Card)	 Inside Back Cover

Additional copies of this card can be ordered separately as *VM/System Product CMS Primer Summary of Commands*, SX24-5151.

Chapter 1. Getting Started



Welcome to VM/SP (Virtual Machine/System Product). VM/SP is a computing system developed by IBM to make your work easier. With the VM/SP system, you can create a "file" (some typed information), store it, change it, make a copy of it, or send it to another VM user without using a single sheet of paper. But if you want printed copies of your file, you can get as many as you want simply by issuing a command to print the file.

It's not necessary to know anything about computers to begin using VM. (We'll call the system "VM," for short.) The keyboard and

screen you'll be using are called a **video display terminal**. The keyboard looks a lot like a typewriter keyboard. Many of the keys serve the same purpose as the corresponding keys on a typewriter keyboard. But when you type on your VM keyboard, the information you type appears on the screen rather than on a piece of paper.

You'll find it very easy to make changes or corrections using your video display terminal. All you have to do to change a letter or a number is to type over it.

Before you get started, just remember that you can't hurt the computer by using your terminal.

This section gives you practice using your terminal and teaches you the logon procedure.

Getting Acquainted with Your Terminal

Before you begin, ask your supervisor or VM administrator to explain:

- How to turn on your terminal and control the brightness and contrast of the display.
- How you can tell if the system is "up" (available for your use) or "down" (not available).
- The functions of the various keys. Many of the keys serve the same purpose as those on a typewriter. Others have special functions that your supervisor should explain.

Now turn on your terminal and adjust the brightness and contrast of the display. The computer displays a **logo** on the screen, similar to this:



VIRTUAL MACHINE/SYSTEM PRODUCT

```
VV      VV  MM      MM
VV      VV  MMM     MMM
VV      VV  MMMM    MMMM
VV      VV  MM MM  MM MM
VV  VV  MM  MMM  MM
VVV     MM  M   MM
V       MM  MM  MM

SSSSSSSS  PPPPPPPP
SS      SS  PP      PP
SS      SS  PP      PP
SSSSSSSS  PPPPPPPP
SS      SS  PP      PP
SS      SS  PP      PP
SSSSSSSS  PP
```

Fill in your USERID and PASSWORD and press ENTER
(Your PASSWORD will not appear when you type it)

```
USERID  ===> -
PASSWORD ===> -
COMMAND ===>
```

Running System

The Keyboard

Let's try some of the special keys on your keyboard. Ignore the logo for now. Follow the directions for using these keys as you read the chapter.

Keys That Move the Cursor

When your system is on and available for use, a short line (or little square) appears on the screen. It is called a **cursor**. The cursor shows you where the next character you type will be displayed on the screen.

As you type, the cursor automatically moves to the next space available. There are times, however, when you'll want to move it to a different place on the screen. Different keys move the cursor one

space forward or backward, or one line up or down. These keys are typamatic, that is, they keep repeating as long as they are held down.



This key moves the cursor down one line. Find it on your keyboard and hold it down. Notice that the cursor keeps moving. When it passes the bottom of the screen, it automatically reappears at the top of the screen. Release the key.



This key moves the cursor up one line. Find the key and hold it down. Notice that the cursor keeps moving. When it passes the top of the screen, it goes off and reappears at the bottom of the screen. Release the key.



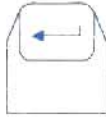
This key moves the cursor one position to the right. Hold this key down. Notice that the cursor keeps moving. When it reaches the right end of the line, it goes off the screen and reappears on the left side, one line below the line it was on. If the cursor is on the bottom line of the screen and is run all the way to the right, it goes off the screen and reappears in the upper left corner. Release the key.



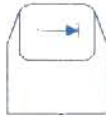
This key moves the cursor one position to the left. Hold this key down. Notice that the cursor keeps moving. When it reaches the left end of the line, it goes off the screen and reappears on the right, one



line above the line it was on. If the cursor is on the top line of the screen and is run all the way to the left, it goes off the screen and reappears in the bottom right corner. Release the key.



This is the new-line key. It moves the cursor to the first position in the next line. It's similar to a carriage return key on a typewriter.



This is the tab key. It moves the cursor to the next position in which you can type.





This is the backward tab (or *back tab*) key. It moves the cursor to the previous position in which you can type.


Capital and Small Letters

You can control the selection of capital (uppercase) and small (lowercase) letters as you would on a typewriter. There are two "shift" keys and a "lock" key. Depending on your terminal model, these keys are marked either SHIFT and LOCK or with special symbols.


To have uppercase letters displayed on the screen, hold down either the right or left shift key as you type. When you press the lock key, the keyboard is set in uppercase letters until you press one of the shift keys.

Correcting Typing Mistakes

Now you're going to practice typing. Press the key marked . The cursor moves next to one of the large arrows near the bottom of the screen. If you try to type anywhere else on the screen other than next to the large arrows, the keyboard may lock. You won't be able to type. (On some terminal models, an "input inhibited" light will turn on.) If your keyboard locks for any reason, press the key marked RESET, and then press the key marked .


If the cursor is not next to the command arrow (COMMAND ==>), keep pressing the key marked  until it is. There will be a space between COMMAND ==> and the cursor. You can't type in that space. Now type these letters:

S O S S O S S O S

If you make a typing mistake, ignore it. Now press the key marked . It moves the cursor back to the beginning of the line, under the first "s" you typed.

A long **space bar** is at the bottom of the keyboard. It is similar to the space bar on a typewriter, with one important difference. This space bar moves the cursor forward a space at a time, but it also erases from the screen anything you have typed in that space. Press the space bar once. The first "s" is erased. Hold the space bar down longer. The cursor moves forward, erasing as it goes. Finish erasing the letters, if any remain.


When you're typing information on a blank screen, use the space bar as you would in normal typing—to put spaces between words.

Again press the key marked  to place the cursor back to the beginning of the line, next to COMMAND ==>. Now type the following sentence, with the errors:



Thus is my fisst experiense typing on tge VM systim.

It's easier to correct typing mistakes with VM than it is on a typewriter. To correct the error in the first word, move the cursor under the letter **u** and type an **i**. Now move the cursor under the first **s** in **fisst** and type an **r**. Next move the cursor under the **s** in **experiense** and type a **c**. Now move the cursor under the **g** in **tge** and type an **h**. Last, move the cursor under the **i** in **systim** and type an **e**.

Let's try another sentence. You could type right over this sentence, but let's erase it instead. You could use the space bar to erase it, but there's another way. Move the cursor back to the beginning of the line with the key marked . Then press the key marked ERASE EOF. You have erased everything from the cursor to the end of the line. EOF means "end of field," which is the end of the line in this case. Now type the following sentence (with errors):

Typing mistales are easy to fiz whem using VM.

To correct the error in the second word, move the cursor under the **l** and type a **k**. Now move the cursor under the **z** in **fiz** and type an **x**. Then move the cursor under the **m** in **whem** and type an **n**.

Let's change this sentence to read: Typing mistakes are easy to fix! Move the cursor to the space after **fix**. Type the exclamation mark, and then erase the rest of the sentence by pressing the ERASE EOF key.

Delete Key

First let's erase the sentence on the screen. Move the cursor to the beginning of the sentence and then press the ERASE EOF key.

Then type:

I ordered zinnia, mariqold, pansy, and daisy seeds.

Let's take out the marigold seeds. First move the cursor under the **m** in **marigold**. Now find the delete key on your keyboard. Depending on the model of your terminal, it is marked either with DEL or a special symbol. Press the delete key once. Notice that the **m** has disappeared, and the rest of the sentence has moved over one space to the left. Now press the delete key until the rest of the word **marigold**, the comma, and the extra space have been deleted. If you had used the space bar instead of the delete key to erase the word, you would have a large blank space where the word **marigold** was.

Now let's type something with numbers. First get rid of the last sentence by positioning the cursor at the beginning of the sentence and pressing the ERASE EOF key.

Now type:

```
The pet store sold 6 dogs, 10 cats, 4 rabbits, and 8 goldfish.
```

Be careful not to type a small "L" instead of the number "1." On most typewriters, it doesn't make much difference which one you use. With VM it does. Notice that the number "0" looks different on the screen. It has a dot in the middle. This is to distinguish it from the capital letter "O."

Now let's change the sentence to reflect the next day's sales at the pet shop. Dog sales were up by one. Position the cursor under the **6** and type **7**. Cat sales were down. Position the cursor under the **1** and type **9**. Then use the delete key to erase the **0**. No rabbits were sold. Use the delete key to erase the **4**, the space, the word **rabbits**, the comma, and the extra space.

Now you should feel comfortable correcting typing mistakes. Move the cursor to the beginning of the line and use the ERASE EOF key to erase the line. You're ready to begin a terminal session.



Logging On and Logging Off

Before you can use the VM system, ask your supervisor or VM administrator to tell you:

- Your user identification (userid)
- Your password
- Whether or not your system has “automatic IPL.” (We’ll explain this later.)

You must begin every terminal session with the simple **logon** (sign on) procedure outlined here. This lets the computer know who you are, where you are, and makes the system available to you.

If you skipped the preceding section, turn on your terminal and adjust its contrast and brightness. The computer displays a **logo** on the screen, similar to this:

```
VIRTUAL MACHINE/SYSTEM PRODUCT
```

```

VV      VV  MM      MM
VV      VV  MMM     MMM
VV      VV  MMMM    MMMM
VV      VV  MM MM  MM MM
VV  VV    MM  MMM  MM
VVV      MM  M    MM
V        MM      MM
          //      //
          SSSSSSS  PPPPPPPP
          SS      SS  PP      PP
          SS      SS  PP      PP
          SSSSSSS  PPPPPPPP
          SS      SS  PP
          SS      SS  PP
          SSSSSSS  PP
```

```
Fill in your USERID and PASSWORD and press ENTER
(Your PASSWORD will not appear when you type it)
```

```
USERID   ==> -
PASSWORD ==> -
COMMAND ==>
```

```
Running System
```

If you did the typing practice in the last section, move the cursor so it is next to `USERID ==>`. Use the key marked `→|`. (The system leaves one space between `USERID ==>` and the cursor.)

Now let's get started:

1. Type your userid. You can use either capital or lowercase letters, or a combination of the two. If your userid is less than eight characters long, press the key marked `→|` or the new-line key to get to the next line.
2. Type your password. You can use either capital or lowercase letters, or a combination of the two. In some systems, you will not see your password on the screen as you type it. This is a security measure. It prevents other people from learning your password and using your system without authorization.

Now press the `ENTER` key. When you press `ENTER`, both your userid and password are sent to the system.

If You Make a Mistake in Your Userid or Password

If you made a typing mistake in your userid, you'll receive this error message:

```
userid not in CP directory.
```

If you made a mistake in your password, you'll receive this error message:

```
LOGON unsuccessful--incorrect password.
```

(If your system has security devices, the keyboard may lock when you give the incorrect password. Then you may have to ask your system operator how to unlock it.)



When the userid or password is incorrect, you will also receive these messages:

Enter one of the following commands:

LOGON userid	{Example: LOGON VMUSER1}
DIAL userid	{Example: DIAL VMUSER2}
MSG userid message	{Example: MSG VMUSER2 GOOD MORNING}
LOGOFF	

You can still log on by typing the following command (the cursor is already in the correct position):

```
L  userid
  ↑  ↑
  |  | type your userid here
  |  | short for "LOGON"
```

For example, if your userid is JONES, you would type L JONES. (You can type commands in either capital or lowercase letters, or a combination of the two.)

After typing the LOGON command, **press the ENTER key**. When you press ENTER, the command you type is sent to the system.

VM responds with this message:

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

Type your password and **press the ENTER key**.

If you make another typing mistake while entering either your userid or password, you will receive one of the two error messages listed above and you must start over again. Type L with your userid and press the ENTER key. VM will ask for the password again.

You've Successfully Entered Your Userid and Password

After you successfully enter your userid and password, VM gives you a logon message. The logon message means that you successfully entered the system. It shows you the time and date you logged on. It can also contain information for VM users, like the hours the system is available on that day. (If the logon message fills up the screen and the word MORE... appears in the lower right corner, press the CLEAR key to continue.)

Does your system have "automatic IPL?" If it does, skip the information in the following box. ("Automatic IPL" means that the system does some work that you would otherwise need to do yourself to IPL the system.)

If your system does not have "automatic IPL," you need to enter another command. Type:

```
IPL CMS
```

You need to enter the IPL CMS command only once during a terminal session, but it's an important command. IPL stands for Initial Program Load. CMS stands for the Conversational Monitor System. CMS is a part of the VM system. It provides a variety of tools to do your job using your terminal. You'll need to use the IPL CMS command after you receive the logon message whenever you want to create files. Now press the ENTER key. VM responds by repeating the command in capital letters.

When you see the message VM READ in the bottom right corner of your screen, press the ENTER key again. VM responds with a message. The last line of the message looks something like this:

```
Ready; T=0.1/0.1 09:02:50
```

This is called the **ready message**. It tells you that VM is ready to continue. The T stands for time. The first series of numbers tells



how much time it took VM to do the last task you gave it. The second set of numbers gives you the time of day.

Now, if you want to begin learning how to use VM, continue with the next chapter. If you've had enough for now, end this terminal session with the LOGOFF command. Type the following command in the command line, where your cursor is:

```
LOG      (short for "LOGOFF")
```

Press the **ENTER** key. VM displays the time you logged off plus other accounting information. It also displays this message:

```
Press enter or clear key to continue
```

Now you can turn off your terminal. Or, if you want to start another terminal session, press the **ENTER** key. The VM logo will appear again, and you can log on.

Here is a summary of the logon procedure:

```
Turn on your display to get the logo.  
Type your userid next to USERID ===>  
Type your password next to PASSWORD ===>  
Press ENTER key.
```

```
With automatic IPL: When VM READ appears,  
press ENTER again and stop.
```

```
Without automatic IPL:  
You type:  IPL CMS  
Press ENTER. When VM READ appears, press ENTER again.
```

To logoff:

You type: LOG
Press the ENTER key.

Status Notices

As you do work on your terminal, the lower right corner of the screen will display various status notices. They tell you what is happening in the system at the present time. If you forget what these notices mean, you can come back to this section for reference.

CP READ

This notice appears after you log on and before you enter the IPL CMS command. It means that a part of the system called the Control Program (CP) is waiting for you to enter a command, such as the IPL CMS command.

The CP READ notice may also appear if you accidentally press the PA1 key. If that happens, type:

b (short for "begin")

and press the ENTER key.

VM READ

This notice means that CMS is waiting for you to enter a command.

RUNNING

This notice means that the computer is working on something or that CMS is waiting for you to enter a command.



MORE . . .

This means that there is more information than can fit on the screen. After one minute, the next screen of information is displayed. To see the next screen right away, press the CLEAR key. To hold this information on the screen, press the ENTER key. This changes the MORE... notice to HOLDING.

HOLDING

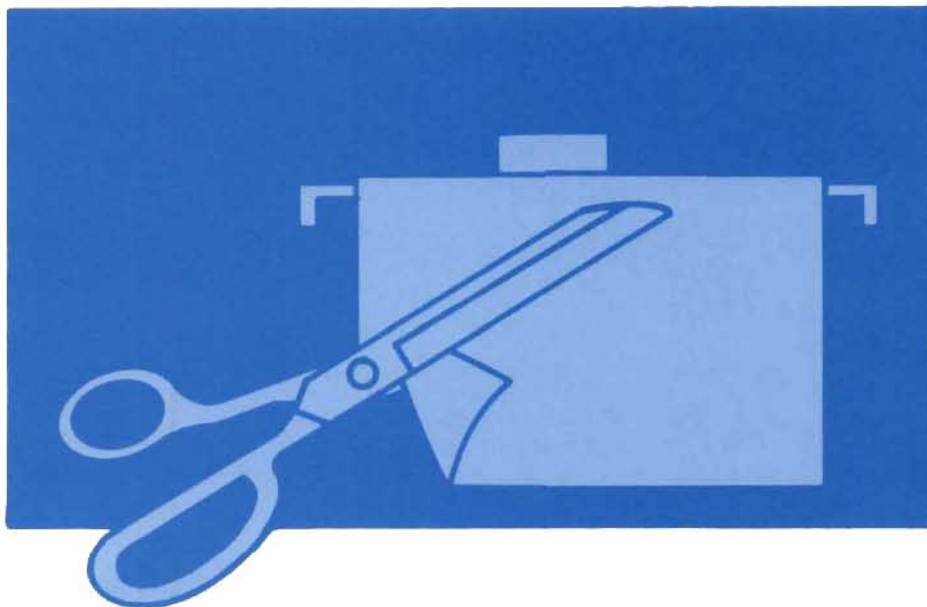
This notice means that the system is waiting for you to clear the screen before showing you more information. It appears when the screen displays a MORE... notice and you press the ENTER key. The notice can also appear when another user sends you a message. To end the hold, press the CLEAR key.

NOT ACCEPTED

This means that the system is working on something and is too busy to accept another command from you. Wait several seconds and enter your command again.

Most of the work you do during a terminal session will be with files. The next chapter teaches you how to create and make changes in files.

Chapter 2. Editing Files



The part of the VM/SP system that helps you work with files is called the **editor**, or XEDIT. To make the editor work for you, you'll need to learn certain commands. This chapter teaches you the commands that create new files and perform basic editing tasks. You should read this chapter at your terminal, doing the exercises as you read the text.

More information about XEDIT 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?

A file is a collection of data. It can be a customer list, inventory, IRS data, motor vehicle data, a document, or anything else. You'll create your own uses.

One thing you need to know about creating files with VM is that each file needs its own 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**.

The filename and the filetype can be any combination of letters and numbers that is no longer than eight characters. (Certain special characters can also be used, but we will use only letters and numbers in our examples.)

You'll probably want to give your files labels that describe their contents and are easy to remember. For example, if you wanted to create a file containing information about animals that were in danger of becoming extinct, you might use the filename **rare**. For the filetype, you might use the word **animals**. The filemode is always a letter. In this chapter, the filemode for all the examples is **a**. So your file about endangered animals would have the label **rare animals a**.

If you wanted to create another file about endangered animals, you would need to use a different file label. You could keep either the same filename or the same filetype, but you couldn't use exactly the same label. For example, your new file could be called **indanger animals** or **rare beasts**.



Here is another important thing you need to know about creating files and using the VM system. 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. If the command has a number 1 in it, be sure to use the number 1 key instead of a small letter L. 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:

```
GET filename filetype filemode
```

The word GET 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 label in their places. You might type your command like this:

```
get rare animals a
```

Some 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.

The last thing you need to know before you create your first file is that you must have followed the logon procedure outlined in the previous chapter. After logging on, you must enter the IPL CMS command (if you don't have automatic IPL). If you forget to enter IPL CMS and try to enter other commands, you'll receive an error message that begins with

```
Unknown CP command
```

Creating Your First File

The name of the command that creates a file is XEDIT. Its format is:

```
Xedit filename filetype filemode
```

The XEDIT command begins this **editing session**. An editing session is the time between entering an XEDIT command and entering a command (discussed later) to let the editor know you're finished working on the file.

Let's create a simple file. It will contain information about things that are found at picnics. The filename is **picnic**. The filetype is **ingredts**. (We can't use **ingredients** because it has more than eight letters, so we'll shorten it to **ingredts**.) The filemode is **a**, but you don't have to type it. If you leave out the filemode, it is assumed to be **a**. Your cursor should be at the bottom left of the screen.

Type

```
x picnic ingredts
```



Press the ENTER key. Your screen now looks like this:

```
1 PICNIC  INGREDTS A1  F 80  Trunc=80 Size=0 Line=0 Col=1 Alt=0
2 Creating new file:
7
5 |----- * * * Top of File * * *
   |...+...1...+...2...+...3...+...4...+...5...+...6...+...7...
   |----- * * * End of File * * *
3 |-----> _
                                     X E D I T  1 File
```

Diagram annotations: A vertical line labeled '6' points to the first column of the file content area. A vertical line labeled '8' points to the eighth column. A vertical line labeled '4' points to the 'X E D I T' text. A bracket labeled '5' spans the two lines of the file content area. A line labeled '7' points to the top of the file content area.

Note: If XEDIT does not work as described in this manual, it is probably because someone in your computer center has defined an *XEDIT profile* for you. (We'll be discussing XEDIT profiles on page 101.) You should ask your supervisor or VM administrator how to avoid using that profile.

Before we see how to enter data in the file, let's look at the way the editor organizes the screen.

1 File Identification Line

The first line on the screen gives you 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. Because we haven't entered any data yet, the file contains zero lines. Line = 0 is the file line number of the current line,

which is explained in number 7, below. Alt=0 is the alteration count, which is explained in "Protecting Your Files" on page 90.

2 Message Line

The editor communicates with you by displaying messages on the second line (and third line, if necessary). These messages might tell you that you made an error in a command, or they might provide information. Here, the editor is telling you that you are creating a new file.

3 Large Arrow Command Line

The large arrow (====>) at the bottom 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. For example, "INPUT," "Input," and "i" are all valid ways to type the INPUT command.

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.

Entering Data

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

```
Input
```

Your cursor should be on the large arrow command line. Now type this command and press the ENTER key.

```
====> input
```

Notice how the screen changes in input mode.



Now type the following input:

```
a loaf of bread
a jug of wine
thou
```

Press the ENTER key. Your screen should look like this:

```
PICNIC  INGREDTS A1  F 80  Trunc=80 Size=12 Line=3 Col=1 Alt=3

* * * Top of File * * *
A LOAF OF BREAD
A JUG OF WINE
THOU
|...+...1...+...2...+...3...+...4...+...5...+...6...+...7...+...

====> * * * Input Zone * * *                                     Input-mode 1 File
```

Cursor

The lines that you typed moved to the top half of the screen. The last line you typed is the new current line (THOU). After you press the ENTER key, the editor changes 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).

You have no more data to type, so press the ENTER key again. This takes you out of input mode and back into edit mode.

```
PICNIC  INGREDTS A1  F 80  Trunc=80 Size=3 Line=3 Col=1 Alt=3
XEDIT:
```

```
==== * * * Top of File * * *
==== A LOAF OF BREAD
==== A JUG OF WINE
==== THOU
==== |...+...1...+...2...+...3...+...4...+...5...+...6...+...7...
==== * * * End of File * * *
```

```
====> _
```

```
X E D I T 1 File
```

The screen is restored to the edit mode layout, and the file contains the data. Notice the changes in the file identification line. Now, Size = 3, because your file contains three lines. Line = 3 means that the current line is line 3. Alt = 3 means that you made three changes (in this case, you added three lines) since you started editing this file.

While you're working on a file, you can enter input mode at any time to insert new lines of data in the file. Just enter the **INPUT** command, and the editor will make room for you to type new lines of data after the current line. To return to edit mode, remember to press the ENTER key **twice**. In this exercise, the Top of File line was the current line when you entered the INPUT command. Later, you will see how to make any line current, so that you can insert lines in input mode between any two existing lines in a file.

We're finished working with this file, so we must tell the editor what to do with it. The editor created your file in **virtual storage**. Virtual storage is temporary storage. If you want to keep the file, you must



enter a FILE command. It tells the editor to put the file on a disk, which is permanent storage. The format of the FILE command is:

```
FILE
```

Now type the command to write your file on disk, and press the ENTER key:

```
====> file
```

The file disappears from your screen, but the editor saved it on your disk. You'll see a ready message (Ready;). It tells you how much computer time you used in creating this file. It also tells you that this editing session is over and VM is ready to do more work for you. Notice that the cursor has returned to a blank line at the bottom of the screen. The blank line is called the *CMS command line*. That's where you'd type the command to create another file.

Note: By now, you should be used to pressing the ENTER key after you type a command on the command line. We won't remind you to press the ENTER key when we ask you to "enter" a command.

Creating a Second File

Let's create another simple file. It will be about crime. The filename will be **crime**. The filetype will be **theft**. Because we're not going to specify a filemode, the filemode will be **a**.

Enter

```
x crime theft
```

Your screen should look like this:

```
CRIME THEFT A1 F 80 Trunc=80 Size=0 Line=0 Col=1 Alt=0  
Creating new file:
```

```
==== * * * Top of File * * *  
|...+...1...+...2...+...3...+...4...+...5...+...6...+...7...  
==== * * * End of File * * *
```

```
====> _
```

```
X E D I T 1 File
```

Capital and Lowercase Letters

In your first file, all the letters you typed were changed to capitals when they were entered in the file. In this file, we want the letters to be entered in the file exactly the way we type them. You can use the SET CASE command to tell 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
```

Enter

```
====> set case m
```



Now let's use input mode to type the data for this file. Enter:

```
====> i      (short for "input")
```

Type the following lines in the input zone:

```
Tom, Tom, the piper's son, Stole a pig and away he run.  
The pig got loose  
and scared a goose,  
And Tom got put in the calaboose.
```

Press the ENTER key **twice** to return to edit mode. Your screen should look like this.

```
CRIME THEFT A1 F 80 Trunc=80 Size=4 Line=4 Col=1 Alt=4  
XEDIT:  
  
==== * * * Top of File * * *  
==== Tom, Tom, the piper's son, Stole a pig and away he run.  
==== The pig got loose  
==== and scared a goose,  
==== And Tom got put in the calaboose.  
==== |...+...1...+...2...+...3...+...4...+...5...+...6...+...7...  
==== * * * End of File * * *  
  
====> _
```

X E D I T 1 File

The letters appear in both capital and lowercase, exactly as you typed them.

All the files we use to practice editing are short. Your "real" files will probably be longer. 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



Press PF11. Now you have two lines:

```
==== Tom, Tom, the piper's son, _ ← cursor
==== Stole a pig and away he run
```

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. For example:

```
==== Tom, Tom, the piper's son, hopped a rig, did a jig,
==== Stole a pig and away he run.
```

To join two short lines:

- Place the **cursor** at the end of the first line.
- Then **press PF11**.

For example, to join the following two lines, place the cursor as shown, and press PF11. (Make sure you leave one space between "loose" and the cursor.)

```
==== The pig got loose _ ← cursor
==== and scared a goose,
```

The result is:

```
==== The pig got loose and scared a goose,
```

Now enter the command to write this file on disk:

```
====> file
```

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 by using a SET TABS command.

Let's create a file that needs tabs. The file will be a list of shortwave radio frequencies. Its filename is **radio**. Its filetype is **waves**. The filemode is **a**.

Enter:

```
x radio waves
```

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

```
Query TABS
```

Now enter:

```
====> q tabs
```

The tab settings are displayed in the message lines near the top of 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.

We're going to use tabs set at columns 1 and 20.



Enter:

```
====> set tabs 1 20
```

Now if you enter QUERY TABS, the new tabs settings are displayed (in the message line). These settings remain in effect for this editing session only. The next time you edit this file, you must enter the SET TABS command again. Otherwise, the initial tab settings will be in effect.

Enter:

```
====> i      (short for "input")
```

In the first line of the input zone, type:

```
voice of america
```

You want to type the radio frequency for Voice of America at tab setting 20. Press PF4. The cursor moves to column 20.

Type:

```
6125
```

Now you're ready to type the next entry. Press PF4, and the cursor moves to column one on the next line.

Type:

```
australia
```

Press PF4 to move the cursor to the next tab setting.

Then type:

```
6035
```

Now you're ready to type the last entry. Press PF4 to move the cursor to the first tab setting in the next line and type:

```
greece
```

Press PF4 to move the cursor to the next tab setting. Type:

```
9615
```

Press the ENTER key twice to move the data up on the screen and to re-enter edit mode. Your screen should look like this:

```
RADIO WAVES A1 F 80 Trunc=80 Size=3 Line=3 Col=1 Alt=3
XEDIT:

===== * * * Top of File * * *
===== VOICE OF AMERICA 6125
===== AUSTRALIA 6035
===== GREECE 9615
===== |...+...1...+...2...+...3...+...4...+...5...+...6...+...7...
===== * * * End of File * * *

=====> _ X E D I T 1 File
```

Enter the command that tells the editor to write this file on disk:

```
=====> file
```

Let's create another file that uses tab settings. If you were a stockroom manager, you might want an inventory file that lists part numbers, part descriptions, locations, and quantities. Use the filename **stock**, the filetype **inven**, and the filemode **a**.



Enter:

```
x stock inven
```

Now you're ready to set the tabs. Enter:

```
====> set tabs 1 8 24 32
```

Enter:

```
====> i      (short for "input")
```

Using PF4 for tabbing, type:

03469	gasket	aa0306	9432
32554	ratchet	aa0205	3762
32868	doohickey	bb0206	5248
64400	gizmo	cc0101	9942
09263	widgel	aa0403	8645
60637	thingamabob	bb0202	5432
26023	whatchamacallit	bb0306	9321

Press the ENTER key twice to return to edit mode.

Displaying PF Key Settings

What can you do if you forget a PF key setting? You can use this command:

```
====> 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.)

To see the setting of a particular PF key, use the QUERY PF command with the PF key number, like this:

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

The PF key setting is displayed in the message line.

Now enter the command to save this file on disk:

```
====> file
```

Inserting Words in a Line

Let's create one last file. Its file label will be *vacation camping a*.

Enter:

```
x vacation camping
```

Enter the command that puts you in input mode:

```
====> i
```

Type, with the error:

```
a clean, stream to fish in
```

There is obviously something wrong with this line. It should read "a clean, clear stream to fish in."

One way to insert letters, words, or spaces in a line is to follow the steps below. For now, just read the instructions.

- Press the **PA2 key**. (Do not confuse this with the PF2 key.) The PA2 key may have a different label, depending on your terminal. If you don't have a key labeled PA2, ask your supervisor to show you its equivalent.
- Then press the **insert mode key**. The insert mode key is labeled either with INS MODE or a special character.



- Then you would be in insert mode, and you could type the characters to be inserted.
- 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. The PA2 key makes it possible for the characters to move over.

Now you can try insert mode. Press the PA2 and insert mode keys. Move the cursor under the **s** in **stream**. Type the word **clear** and press the space bar. As you type, the words "stream to fish in" move over to make room for the new word. When you finish typing, you must get out of insert mode in order to type normally. Press the RESET key to get out of insert mode.

Use the new-line key to continue typing the file. Type:

```
wood nearby forcooking
```

You need to insert a space between **for** and **cooking**. Press PA2 and the insert mode keys. Move the cursor under the **c** in **cooking**. Press the space bar once. Now press the RESET key to get out of insert mode.

Use the new-line key to move the cursor to the beginning of the next line. Type, with the error:

```
no wil bears
```

You want to change **wil** to **wild**, using insert mode. Press the PA2, then the insert mode key. Move the cursor after the letter **l** in **wil**. Type a **d**. Then press the RESET key.

Move the cursor to the beginning of the next line and type:

```
nothing to fear
```

Now you want to insert the word **else** between the words **nothing** and **to**. Press the PA2 and insert mode keys. Move the cursor under the **t** in **to**. Type the word **else** and press the space bar. Last, press the RESET key.

Move the cursor to the beginning of the next line and type, with the error:

```
soothing surroundings forthe eye
```

Using the insert mode, insert a space between **for** and **the**. Then move the cursor to the beginning of the next line and type, with the error:

```
nothing to ofend the ear
```

Using the insert mode, change **ofend** to **offend**. Then move the cursor to the beginning of the next line and type:

```
no early rises camped nearby
```

Using the insert mode, change **rises** to **risers**. Move the cursor to the beginning of the next line and type, with the error:

```
easy on my hard-earn money
```

Using the insert mode, change **earn** to **earned**.

Now press the ENTER key twice to get out of input mode. Then enter:

```
====> file
```

Remember: Each time you need to insert characters in a different line, you must press PA2 in order to use insert mode. After you're finished inserting characters in that line, remember to press the RESET key. You can use insert mode in both edit and input modes.



To continue learning about editing, go on to the next section. If you've had enough for now, enter the command to end this terminal session:

```
log
```

Summary

You learned the following commands/PF keys in this section:

Command/PF Key	Function
Xedit	Creates a file.
Input	Enters input mode for typing your data.
FILE	Asks the editor to write your file on disk.
Query TABS	Displays tab settings.
SET TABS	Changes tab settings.
SET CASE Mixed	Causes data to be entered in capital and lowercase letters.
PF4 (Tab key)	Tabs the cursor.
PF11 (Split/Join)	Splits a line or joins two lines, at the cursor.
PA2 and Insert Mode keys	Enters insert mode for one line.
Query PF	Displays all PF key settings.
Query PFn	Displays one PF key setting, where n is the PF key number.
RESET key	Gets out of insert mode.

Changing Files (Part One)

Once you create a file, sooner or later you may have to change it. You may want to add or take out some information or move the information around. This section teaches you the commands that display a list of your files, add new lines, take out lines, duplicate, move, and copy lines.

Displaying a File on Your Screen

To work on a copy of a file that is on your disk, use the same command that you used to create the file:

```
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 command.

I Don't Remember the File Label

You may not remember the exact name of a file you want to work with. You can use the FILELIST command to see a list of the files that are on your disk. One format of this command is:

```
FILEList
```

Now enter this command on the CMS command line.

```
filel
```




If you did the exercises in the previous section, you have at least five files on your disk. Your list looks something like this:

```
yourid  FILELIST A0  V 108  Trunc=108  Size=19  Line=1  Col=1  Alt=0
Cmd    Filename Filetype Fm Format Lrecl  Records  Blocks  Date      Time
      VACATION  CAMPING  A1 F      80       8        1  9/14/86  10:02:43
      STOCK   INVEN    A1 F      80       7        1  9/14/86  9:57:11
      RADIO   WAVES    A1 F      80       3        1  9/14/86  9:53:14
      CRIME   THEFT    A1 F      80       4        1  9/14/86  9:50:54
      PICNIC  INGREDTS A1 F      80       3        1  9/14/86  9:48:03
      .
      .
      .

1= Help      2= Refresh  3= Quit     4= Sort(type)  5= Sort(date)  6= Sort(size)
7= Backward  8= Forward  9= FL /n   10=           11= XEDIT      12= Cursor

====>

X E D I T  1 File
```

Cursor

To work on one of your files, follow these steps:

- 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 the XEDIT command from the command line does. The file appears on the screen, replacing the file list.

A more complete description of the FILELIST screen is in Chapter 3. In this chapter, we're using the file list only to get a copy of a file for editing.

The first file you'll be editing is the one labeled **radio waves a**. To get a copy of this file on your screen:

- Move your cursor to the line that has "RADIO WAVES."

- Press PF11.

The RADIO WAVES file appears on your screen.

When you created the file, you used the SET TABS command to set the tabs at columns 1 and 20. Because you're going to use tabs while revising the file, you'll have to enter this command again.

Enter:

```
=---> set tabs 1 20
```

Adding Lines

You've discovered some new stations on your shortwave radio and you want to add them to your list. You want to add Quito, Ecuador, after the listing for Radio Australia. You already know that one way to add a line of information is by entering 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.
```

Now move your cursor to the prefix area of the line that has the entry for Australia, and type the letter "a," like this:

```
a==== AUSTRALIA           6035
```

Press the ENTER key. A blank line is added.

```
==== AUSTRALIA           6035  
====  
==== GREECE              9615
```

The cursor is at the beginning of the blank line. Type the name of the country. Then use PF4 to tab the cursor to the column where you want to type the number.

```
==== quito, ecuador      9560
```

Now let's add three new listings after the listing for Greece. Use the new-line key to move the cursor to the prefix area of the line that has the entry for Greece. Type the **a** prefix command that will add three blank lines:

```
3a==== GREECE           9615
```

When you press the ENTER key, three blank lines are added. The cursor is at the beginning of the first blank line.

Use PF4 for tabbing, and type:

```
==== bbc                 9750  
==== havana             6165  
==== canada            9655
```

Let's add one more entry, after Quito, Ecuador. Move the cursor to the prefix area of the line that contains this entry. Then enter an **a** prefix command.

```
==== QUITO, ECUADOR      9560
```

Type:

```
==== netherlands      9865
```

Moving through a File

If you were to continue adding lines, sooner or later this file wouldn'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.

If your file is many screens long, you press either of these keys repeatedly to go forward or backward through the file.

Moving to the Bottom or Top

Suppose your file is many screens long, and the current screen is somewhere in the middle of the file. To go forward to the end of the file, you could press PF8 repeatedly. Or, you can use the **BOTTOM** command.

The **BOTTOM** command makes the last line of the file the new current line. Its format is:

```
Bottom
```



Your cursor is probably on the last entry you typed. You want to move it to the large arrow command line so that you can enter a BOTTOM command. **To move the cursor to the large arrow command line from any place on the screen, just press the ENTER key** (or the PF12 key). Pressing either key moves the cursor to the command line. When you're finished entering a command, you can press PF12 to move the cursor back to where it was on the screen.

Press ENTER now to move your cursor to the command line. Then enter:

```
====> b      (short for "bottom")
```

The last line in the file is now the current line. To add space after the last line in a file, you could use the **a** prefix command. However, if you don't know how many lines you want to add, it's better to use the INPUT command. Why? Let's say you used the **a** prefix command to add ten blank lines. But you found you needed to type in only five of those lines. The extra five blank lines would remain in your file, taking up space.

Now enter:

```
====> i      (short for "input")
```

Using PF4 for tabbing, type:

```
spain           6090
peking          9780
moscow          9450
```

Now press the ENTER key twice to return to edit mode.

The TOP command makes the Top of File line the new current line. You enter the TOP command on the large arrow command line. Its format is:

```
TOP
```

You might want to use this command when you need to add a line at the beginning of the file.

Enter:

```
====> top
```

Enter an **a** prefix command on the top line:

```
a==== ***Top of File***
```

Then type:

```
===== west germany          9765
```

Moving Down or Up

Suppose that you want to move the display up or down a few **lines** instead of a whole screen full of lines.

Use the DOWN command to move the display one or more lines toward the end of a file. The format of the DOWN command is:

```
Down n
```

You substitute the number of lines you want to move down for "n". Of course, a DOWN command changes which line in the file is the current line on your screen.

The UP command moves the display toward the top of the file. Its format is:

```
Up n
```

You substitute the number of lines you want to move up for "n". An UP command changes 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 the ENTER key is pressed, 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 can enter a / in any position of the prefix area. For example:

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

Now is a good time to practice using the commands that move through a file. Practice entering BOTTOM, TOP, DOWN, and UP from the command line. Enter the / prefix command from a prefix area of any line.

You can't hurt or change the file as you enter these commands. All they do is move the display around.

Now we're finished adding lines to this file. Your file contains the following entries. (Your screen may look different, depending on which commands you used to practice moving through the file. But the entries should be the same.)


```
RADIO WAVES A1 F 80 Trunc=80 Size=12 Line=5 Col=1 Alt=13
```

```
==== * * * Top of File * * *
==== WEST GERMANY 9765
==== VOICE OF AMERICA 6125
==== AUSTRALIA 6035
==== QUITO, ECUADOR 9560
==== NETHERLANDS 9865
|...+...1...+...2...+...3...+...4...+...5...+...6...+...7...
==== GREECE 9615
==== BBC 9750
==== HAVANA 6165
==== CANADA 9655
==== SPAIN 6090
==== PEKING 9780
==== MOSCOW 9450
==== * * * End of File * * *

====> _
```

```
X E D I T 1 File
```

Enter:

```
====> file
```

The revised file now replaces your original file. The file list returns to the screen, with an asterisk (*) in the "Cmd" space next to RADIO WAVES. This is just to remind you that you did something with this file. It does not prevent you from working with this file again.



More Practice in Adding Lines

The next file you're going to change is the one that lists the ingredients for a picnic. Find that file in your file list. Move the cursor to that line in the file list, and press PF11.

First we're going to practice adding lines to this file. Then we'll show you how to take out lines. Your file looks like this:

```
PICNIC  INGREDTS A1  F 80  Trunc=80 Size=3 Line=0 Col=1 Alt=0

===== * * * Top of File * * *
===== |...+...1...+...2...+...3...+...4...+...5...+...6...+...7...
===== A LOAF OF BREAD
===== A JUG OF WINE
===== THOU
===== * * * End of File * * *

=====> _

X E D I T  1 File
```

You decide that a loaf of bread, while nourishing, isn't all that appetizing by itself. You want to include some ham, adding it after the bread entry. Move the cursor to the prefix area of the line containing A LOAF OF BREAD, and enter an **a** prefix command, like this:

```
a===== A LOAF OF BREAD
```

In the blank line, type

```
===== ham
```

The ham sounds so good that you decide to pack a few other things to go with it. You want to add them to your list after the line reading **ham**. To add six more items, each on a separate line, you'll need six extra lines. The cursor is at the end of the ham entry. Move it to the prefix area of that line, and enter the prefix command to add six blank lines, like this:

```
6a=== ham
```

Type

```
===== cheese  
===== hot dogs  
===== mustard  
===== pickles  
===== olives  
===== potato chips
```

You've just remembered that there are also some unpleasant things found at picnics. You want to add them after the last line of the file. Press the ENTER key to move the cursor to the large arrow command line.

Then enter:

```
====> b      (short for "bottom")
```

Enter:

```
====> i      (short for "input")
```

Type these lines:

```
warm orange soda  
hot iced tea  
ants  
sunburn
```



Looking at the list, you decide that two of the most important ingredients for a picnic are missing. You want to add them at the top of the list, before A LOAF OF BREAD. You're going to enter a command, so you must get out of input mode and back into edit mode. (Remember, you can't enter commands from the command line in input mode.) Press ENTER twice to re-enter edit mode.

Enter:

```
====> top
```

Enter "2a" in the prefix area of the Top of File line. Then type:

```
-==== friendly companions  
-==== pleasant surroundings
```

Now you have all the ingredients you need for a perfect picnic. But wait, you just got a phone call. It seems that "Thou" can't come to the picnic after all. What's more, you learn that the picnic area where you plan to go won't allow the consumption of wine. To make your file up to date, you must delete some lines. To learn how, read on.

Deleting Lines

The **d prefix command** deletes the line in which it is entered. (Do not confuse it with the delete **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.
```

You want to delete the line containing A JUG OF WINE and the line containing THOU. Press PF8 to scroll forward. Move the cursor to the line containing A JUG OF WINE, and enter the **d** prefix command that deletes two lines:

```
2d=== A JUG OF WINE
```

When you press the ENTER key, the two lines are deleted.

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 space bar once before pressing the ERASE EOF key. If you don't press the space bar, 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.

Now press the ENTER key to move your cursor to the command line, and enter:

```
====> file
```

The file list returns to the screen. If you needed to work on another file, you would move the cursor to the correct line and press PF11.

The rest of this section doesn't involve any terminal exercises for you to do. Because you're finished editing files for now, press PF3 to exit from the file list.



Deleting a Block of Lines

A block of lines is a group of consecutive lines. Sometimes you may want to delete an entire block of lines, but you don't want to bother counting how many.

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 the ENTER key is pressed, all of the lines are deleted.

What if you want to delete a large section of a file? 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. Remember how to scroll the file? 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:

```
'DD' pending...
```

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

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 last lines that you deleted are the first to be recovered. If, for example, you delete 10 lines and recover 5, the last 5 lines deleted are those recovered.

The recovered line(s) is inserted starting at the current line. The first line recovered becomes the new current line. You can put the lines back where they belong by using the **m** prefix command. This command is discussed under "Moving Lines" on page 57.

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 ". For example:

```
3"=== I want three more copies of this line.  
===== Oh yeah?
```

When the ENTER key is pressed, the file looks like this:

```
===== I want three more copies of this line.  
===== I want three more copies of this line.  
===== I want three more copies of this line.  
===== I want three more copies of this line.  
===== Oh yeah?
```



Copying Lines

Suppose you have a table that appears in one part of a file, and you want it to appear again in another part of the same file. Do you have to retype it? No, 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 would 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 type an “f” or “p”. 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 would be placed before the line containing the “p”. (If the line contained an “f”, the block would go after that line.)

What 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:

```
'CC' pending...
```

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

Moving Lines

Moving lines means to remove them from their original place and put them in another. For instance, if you have a table that appears on one part of a file, and you want to remove the table from that part of the file and put it in another part, you would move (not copy) it.

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", like this:

```
5m=== I want to move this line and the next four lines.  
      .  
      .  
      .  
====f I want them to be moved following this line.
```

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".

RESET Command

You know that scrolling the file while entering different parts of a prefix command causes the status area to display a “pending” notice. A pending notice is also displayed if you type a letter that is not a valid prefix command (like “v”).

To clear the prefix area and remove the pending notice, enter this command (in the large arrow command line):

```
====> RESet
```

(Do not confuse the RESET **command** with the RESET **key**, which is used to get out of insert mode.)

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

Summary

You learned the following commands/PF keys in this section:

Command/PF key	Function
FILEList	Displays a list of your files, from which you can call out a copy of a file for editing.
Bottom	Makes the last line of the file the new current line.
TOP	Makes the ***Top of File*** line the new current line.



Command/PF key	Function
Down	Moves the current line toward the end of a file.
Up	Moves the current line toward the top (beginning) of a file.
RECover	Puts back lines deleted by a "d" prefix command.
RESet	Cancel prefix commands when the status area displays a pending notice.
a (prefix)	Adds blank line(s).
/ (prefix)	Makes this line the new current line.
d (prefix)	Deletes a line(s).
" (prefix)	Duplicates a line(s).
c (prefix)	Copies a line(s). Must show destination with f or p.
m (prefix)	Moves a line(s). Must show destination with f or p.
f (prefix)	Moves/copies line(s) following this line.
p (prefix)	Moves/copies line(s) preceding this line.
PF7 (BACKWARD)	Scrolls backward one screen.
PF8 (FORWARD)	Scrolls forward one screen.

Changing Files (Part Two)

When you're looking at a display of data you have just entered, and you decide to make some changes, it's easy to type over the information to be changed. But it's not always that simple. Typically, you have many files and need to make changes even though you don't know exactly where in a file the data is located. The commands in this chapter help you quickly **locate** and easily **change** the information in your files.

If you're not already logged on, use the logon procedure now.

LOCATE and CHANGE Commands

The first file you're going to change is the one that lists shortwave radio frequencies. Assume you made the list after listening to your shortwave one night. You heard a voice in Spanish mention Havana and you assumed you were listening to Havana, Cuba. But when you tuned in to the same frequency later, you discovered that you had actually been listening to a CB radio in Havana, Illinois. Now that you know the correct frequency for Radio Havana, you want to correct your file.

Now enter:

```
file1      (short for "filelist")
```

You want to work on the file whose label is RADIO WAVES, so follow these steps:

- Move the cursor to the line containing RADIO WAVES.
- Then press PF11.



One way to change the frequency for Havana would be to move the cursor under the first number in the wrong frequency and type the correct number over it. But there's another way to make the change. Let's try it.

First you want to tell the editor to locate the line that contains the frequency you want to change. You use a LOCATE command, whose format is:

```
Locate/your data
```

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/CMS or 06/04/86, you must use a different special character as a separator. For example, LOCATE\$VM/SP or LOCATE%06/04/86. The following special characters cannot be used as separators: \neg (not), + (plus), - (minus), or . (period).

Now enter:

```
====> L/havana      (short for "locate/havana")
```

This command tells the editor to look through the file, starting with the line following the current line, until it finds HAVANA. When it finds the first occurrence of HAVANA, it makes the line that contains it the new current line.

Now that the data is located, we can change it using the CHANGE command. The CHANGE command has the format:

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

The slash (/) is the separator character. Instead of the slash you can use any special character except \neg (not), + (plus), - (minus), or . (period). For example, you might enter: c?6165?9525?

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.

Now enter:

```
====> c/6165/9525/
```

The editor changed the line to read "HAVANA 9525." The message "1 occurrence(s) changed on 1 line(s)" tells you how many changes were made. The HAVANA entry is now in the current line. (Remember, the current line is in the middle of the screen and appears brighter than the other file lines.) What happens if you use the "locate/havana" command now? Let's see. Enter:

```
====> L/havana
```

A message, "Target not found," is displayed. It means the editor could not find the word you are looking for. In other words, it couldn't find the "target" of the search. The editor started looking for HAVANA in the line *following* the current line. It then looked in each subsequent line, until it reached the end of the file. Because HAVANA was in the current line, the editor didn't find it. Notice that the command you typed is redisplayed in the command line. This lets you see if you made a typing mistake. Note also that the End of File line is now the current line.

What happens if you try to use the CHANGE command without first making the line you want to change the current line? Let's try it and see.

Press the ERASE EOF key to clear the command line. Then enter:

```
====> l.op
```



Try to change the frequency for VOICE OF AMERICA by entering:

```
====> c/6125/9590/
```

When you press the ENTER key, the message "No line(s) changed" is displayed. The editor looked in the current line (the Top of File line) and didn't find the frequency "6125."

To make that line current, first press the ERASE EOF key to clear the command line, then enter this LOCATE command:

```
====> L/6125
```

The line containing VOICE OF AMERICA is now the current line.

Enter:

```
====> c/6125/9590/
```

Let's put the file back into permanent storage.

Enter:

```
====> file
```

The revised file is now on your disk. The file list returns to the screen.

Now let's make some changes in the crime file. Move the cursor to the correct line in the file list, and press PF11.

When you created this file, you used the SET CASE MIXED command so that you could have both capital and lowercase letters in the file. Otherwise, all the letters would be in capitals, as they are in the other files you created.

You must enter the SET CASE MIXED command at the beginning of each editing session, or all letters you enter will be capitals.

Enter:

```
====> set case mixed
```

Here is the crime file.

```
CRIME    THEFT    A1  F 80  Trunc=80 Size=4 Line=0 Col=1 Alt=0

==== * * * Top of File * * *
|...+....1...+....2...+....3...+....4...+....5...+....6...+....7...
==== Tom, Tom, the piper's son,
==== Stole a pig and away he run.
==== The pig got loose and scared a goose,
==== And Tom got put in the calaboose.
==== * * * End of File * * *

====> _

X E D I T  1 File
```

Let's say that you've learned that a mistake was made in reporting the theft. The culprit's father was a plumber, not a piper. The current line is now the ***Top of File*** line. First you need to make the line containing "piper" current. Do you remember how? One way is to use the command "locate/piper". Another way is to use "down 1". Still another is to use the / (slash) prefix command. Choose one of these ways to make the line containing "piper" current.

Then enter:

```
====> c/piper's/plumber's/
```

The first line now reads, "Tom, Tom, the plumber's son."



Changing a Word Throughout a File

You now learn that Tom wasn't the thief. It was his older brother Mort. You want to change **Tom** to **Mort** throughout the file. You can change Tom to Mort every time it appears in the file with a single command. It's the same old CHANGE command, with something added.

Enter:

```
====> c/Tom/Mort/ * *
```

The message "3 occurrence(s) changed on 2 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 Tom to Mort. Note the format of this command. You must type a separator, which is a slash (/) above, after the new word (Mort), and you must type two asterisks **with a space between them** (* *). The first asterisk tells the editor to search every line of the file starting from the current line. The second tells it to change every occurrence of the old word in each line.

Notice what happened to the current line. It is now the End of File. After a CHANGE command, the last line searched becomes the new current line. Because you told the editor to search forward in the file for the word "Tom," the End of File line is the new current line.

Keep in mind 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.

Just as you think you have the whole story of the crime, a late flash comes in. The animal stolen wasn't a pig. It was a cow. To change

the word **pig** to **cow** wherever it appears, first make the top line current:

```
====> top
```

Then enter:

```
====> c/pig/cow/ * *
```

Your file looks like this.

```
CRIME    THEFT    A1 F 80 Trunc=80 Size=4 Line=5 Col=1 Alt=3
2 occurrence(s) changed on 2 line(s)

==== * * * Top of File * * *
==== Mort, Mort, the plumber's son,
==== Stole a cow and away he run.
==== The cow got loose and scared a goose,
==== And Mort got put in the calaboose.
==== * * * End of File * * *
      |...+...1...+...2...+...3...+...4...+...5...+...6...+...7...

====> _

X E D I T 1 File
```

Now that you've finished changing the file, enter the command to put the file in permanent storage:

```
====> file
```

Your file list returns to the screen.



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. Let's look at the VACATION CAMPING file. Move your cursor to that line in the file list, and press PF11. Your file looks like this:

```
VACATION CAMPING  A1  F 80  Trunc=80  Size=8  Line=0  Col=1  Alt=0

===== * * * Top of File * * *
|...+...1...+...2...+...3...+...4...+...5...+...6...+...7...
===== A CLEAN, CLEAR STREAM TO FISH IN
===== WOOD NEARBY FOR COOKING
===== NO WILD BEARS
===== NOTHING ELSE TO FEAR
===== SOOTHING SURROUNDINGS FOR THE EYE
===== NOTHING TO OFFEND THE EAR
===== NO EARLY RISERS CAMPED NEARBY
===== EASY ON MY HARD-EARNED MONEY
===== * * * End of File * * *
=====> -
```

X E D I T | File

Looking at the file again, you decide that your nose is more sensitive than your ear. You want to change the sixth line to read NOTHING TO OFFEND THE NOSE. What happens if you use the CHANGE command followed by asterisk-space-asterisk to make the change? Let's try it and see.

Enter:

```
=====> c/ear/nose/ * *
```

Here is the changed file.

```

VACATION CAMPING A1 F 80 Trunc=80 Size=8 Line=9 Col=1 Alt=1
8 occurrence(s) changed on 7 line(s)
===== * * * Top of File * * *
===== A CLEAN, CLNOSE STREAM TO FISH IN
===== WOOD NNOSEBY FOR COOKING
===== NO WILD BNOSES
===== NOTHING ELSE TO FNOSE
===== SOOTHING SURROUNDINGS FOR THE EYE
===== NOTHING TO OFFEND THE NOSE
===== NO NOSELY RISERS CAMPED NNOSEBY
===== EASY ON MY HARD-NOSENEED MONEY
===== * * * End of File * * *
|...+...1...+...2...+...3...+...4...+...5...+...6...+...7...

=====> _
X E D I T 1 File

```

Not quite what you expected, is it? XEDIT has changed the letters EAR to the letters NOSE wherever they appeared. As you can see, 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.

Let's reverse our mistake. First make the top line current by entering:

```
=====> top
```

Then enter:

```
=====> c/nose/ear/ * *
```

All occurrences of NOSE, including those that existed *before* the change, are changed to EAR.



How can we make the change we wanted? First enter:

```
====> top
```

Now let's try another CHANGE command. Make sure you press the space bar to put the spaces exactly as shown below.

Enter:

```
====> c/ ear / nose / * *
```

You added spaces before and after EAR and NOSE. This tells the editor that you want to change EAR to NOSE when EAR is a word, not part of a word. Now the sixth line reads NOTHING TO OFFEND THE NOSE. It's important to include the spaces both before and after the words. If you left out the spaces **after** the words (c/ ear/ nose/ * *) the sixth line would still read NOTHING TO OFFEND THE NOSE. But, it would also have changed the eighth line to read NO NOSELY RISERS CAMPED NEARBY.

Let's forget about vacations and camping and ears and noses for a moment. Suppose you had a file that read:

```
THE QUICK BROWN FOX JUMPED OVER THE LAZY DOG.  
HE THEN RAN A MILE IN ONE MINUTE.  
AFTER THAT HE WAS TOO TIRED TO RUN ANYMORE.  
SO HE SLOWLY WALKED BACK HOME.
```

Suppose you decide to change all the references in the file from HE to SHE. You used the command:

```
====> c/ he / she / * *
```

The command would change HE to SHE in the third and fourth lines, but not in the second line. Do you know why? There is no space before the word HE in the second line, because HE begins that line! In order to make the change in the second line, you would have to make that line current. Then you would enter the command:

```
====> c/he/she/
```

Returning again to the vacation file, let's change NOSE back to EAR in order to practice another command.

First enter:

```
====> top
```

Enter:

```
====> c/ nose / ear / * *
```

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).

In your camping file, the current line is now the End of File line. Let's say you wanted to use a LOCATE command to locate the word EAR. That line is in a backward direction (toward the top of the file) from the current line. You know that you could enter a TOP command so that the search would start on the line following the current line. You can also ask the editor to search in a backward direction from the current line. You do this by typing a minus sign (-) before the slash (/).

Enter:

```
====> L -/ ear /
```

You typed a space and another slash after "ear". This command tells the editor that you want only the word EAR. If you entered "L -/ ear" without a space and a slash at the end of the command, the editor would look for the word EAR and for the letters EAR at the beginning of a word.



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/ WILD BEARS/ in the VACATION CAMPING file. These two words are on the same line. (It is possible to use the LOCATE command to locate two words that aren't on the same line, but that is beyond the scope of this book.)

Let's return this file to permanent storage.

Enter:

```
====> file
```

Your file list returns to the screen.

The = (equal) Command

Call up the stock inventory file by moving the cursor to that line in your file list and pressing PF11. The file looks like this:

```
STOCK      INVEN      A1 F 80  Trunc=80 Size=7 Line=0 Col=1 Alt=0

===== * * * Top of File * * *
|...+...1...+...2...+...3...+...4...+...5...+...6...+...7...
===== 03469  GASKET          AA0306  9432
===== 32554  RATCHET          AA0205  3762
===== 32868  DOOHICKEY         BB0206  5248
===== 64400  GIZMO             CC0101  9942
===== 09263  WIDGET           AA0403  8645
===== 60637  THINGAMABOB      BB0202  5432
===== 26023  WHATCHAMACALLIT BB0306  9321
===== * * * End of File * * *

=====> -                                     X E D I T  1 File
```

Let's say you're the stockroom manager. The third item in each line is a code that tells you where each part is stored. In the first line, the code is AA0306. AA is the aisle, 03 is the shelf, and 06 is the bin. You want to quickly scan through a file, locating the items stored in aisle AA.

Enter the LOCATE command to find AA:

```
=====> L/aa
```

The first line containing AA is now the current line. You want to quickly scan the file, locating AA in every line it appears. Do you have to retype the LOCATE command? No, you can use the PF9 key, which is set to the = (equal) command. The = command



automatically repeats the last command you entered. It saves you from having to retype it. It can be used to repeat any command, not just a LOCATE command.

Now press PF9. The next line that contains AA becomes the current line. Press PF9 again. The same thing happens. Press PF9 a third time. You see the message "Target not found." This means that there are no more lines containing AA.

We're finished working with this file. Press the ERASE EOF key to clear the command line.

Then enter:

```
====> file
```

Now press the PF3 key to exit from the file list.

If you've had enough for this terminal session, log off and turn off your terminal. If not, continue with the next section.

Summary

You learned the following commands/PF keys in this section:

Command/PF Key	Function
Locate	Makes the line containing the data you're looking for the new current line.
Change	Changes one word or group of characters to another, either in the current line or from the current line to the last (if entered with * *).
PF9 (=)	Repeats the last command you executed. PF9 is set to the = (equal) command.



Changing Files (Part Three)

So far, you've learned how to make changes to the data in a file. In this section, you'll learn how to combine files. You'll also learn how to put the lines in a file in alphabetical (or numerical) order. And now that the number of XEDIT commands you've learned is growing, you'll learn how to display information on various XEDIT commands if you've forgotten how to use them.

If you're not already logged on, use the logon procedure now.

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

We don't have any files that logically belong together. So we'll just combine files in order to practice the GET command.

Now enter:

```
file1    (short for "filelist")
```

We have two files on outdoor activities. Let's combine these files into one file. Get a copy of the picnic file on your screen by:

- Moving the cursor to the line containing PICNIC INGREDTS,

- Then pressing PF11.

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. Let's insert the camping file after the last line of the picnic file. We'll make the last line current by using the BOTTOM command.

Enter:

```
====> b      (short for "bottom")
```

The line SUNBURN is now the current line. The GET command has the format:

```
GET filename filetype
```

You substitute the filename and filetype of the file you want to insert.

Now enter:

```
====> get vacation camping
```

The message "EOF reached" is displayed. This means that a copy of the entire camping file has been inserted after the SUNBURN line. Notice that the last line inserted (EASY ON MY HARD-EARNED MONEY) is the new current line. Do you want to see the whole file? Enter a TOP command. Then scroll the screen forward (PF8).

Let's put the file back on disk.

Enter:

```
====> file
```

The revised file is now on your disk. The file list returns to the screen.



What happened to the original camping file? Is it gone from your disk? No, the editor has only made a copy of it and inserted the copy in the picnic file.

Inserting Part of a File

Let's look at the camping file. Move your cursor to the VACATION CAMPING line in the file list and press PF11.

We'll insert part of the radio waves file after the "Top of File" line of the camping file. We don't have to change which line is current, because "Top of File" is 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
```

Let's insert lines 1 through 11. Remember to use the number 1 key instead of the letter L key.

Enter:

```
====> get radio waves 1 11
```

The editor inserted lines 1 through 11 of the shortwave file after the Top of File line in the camping file. The last line inserted (PEKING 9780) is the new current line.

I Don't Want to Keep the Changes!

Because you've been using the camping file only to practice the GET command, you've really created a mess. You don't want to put this version on your disk. How can you get rid of the changes and keep the camping 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.)

Now press PF3. The following message is displayed:

```
File has been changed; type 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.

Now enter:

```
====> qq
```

The file list returns. What happened to the camping file? Nothing. Because you entered the QQUIT command instead of the FILE command, the camping file on disk isn't changed. It remains as it was when you started this editing session.

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.

You would also use QUIT instead of FILE when you're creating a new file and decide not to keep it. Let's create a new file. We'll call it PHONE NUMBERS.

On the command line at the bottom of the file list, enter:

```
====> x phone numbers
```

Now set tabs by entering:

```
====> set tabs 1 30
```

Enter the command that puts you in input mode:

```
====> i
```

Now type these lines, using PF4 for tabbing:

Gable, Clark	212-000-0000
Hari, Mata	312-000-0000
Hun, Attila the	514-000-0000
Ripper, Jack D.	915-000-0000

Press the ENTER key twice to return to edit mode. You suddenly realize that the data you're using is out of date. You want to quit making this file. Press PF3. Then enter:

```
====> qq
```

You can see in your file list that there is no file called PHONE NUMBERS.

SET NUMBER ON Command

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. So you're probably thinking that the GET command is great if you know the numbers of the lines you want to insert. But what if you're working with a long file and you want to get lines from somewhere in the middle? How are you going to count all those lines? You don't have to. You can use the SET NUMBER ON command to number the lines of the file. Its format is:

```
SET NUMBER ON
```

Let's insert part of the radio waves file into the camping file. This time we'll ask the editor to number the file lines.

Move the cursor to the line on the file list that contains RADIO WAVES, and press PF11. Then enter:

```
====> set num 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.

When you no longer need the line numbers, enter:

```
SET NUMBER OFF
```

Let's insert lines from this file into the camping file. We'll insert lines beginning with QUITO, EQUADOR (line 4) and ending with HAVANA (line 8).



Reviewing the GET command format:

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

you see that you would substitute "4" for "first-line" and "5" for "number-of-lines".

Now you're finished with this file. Press PF3 to quit.

Call out the camping file from storage by moving the cursor to that line on the file list and pressing PF11.

Now enter the GET command:

```
====> get radio waves 4 5
```

Remember, the second number in the GET command is the total number of lines you want inserted, not the line number. Remember, too, that the GET command inserts lines after the current line.

We don't want to save these changes, so press PF3.

Then enter:

```
====> qq
```

Sorting a File

The place names in the RADIO WAVES file aren't listed alphabetically. You now decide that the file would be more useful if the listings were in alphabetical order. Is it necessary to retype the file in alphabetical order? No. You can give the editor a command to do it for you.

On your file list screen, move the cursor to the RADIO WAVES line and press PF11. Your file looks like this:

```
RADIO   WAVES   A1 F 80  Trunc=80 Size=12 Line=0 Col=1 Alt=0

===== * * * Top of File * * *
|...+...1...+...2...+...3...+...4...+...5...+...6...+...7...
===== WEST GERMANY      9765
===== VOICE OF AMERICA  9590
===== AUSTRALIA        6035
===== QUITO, EQUADOR   9560
===== NETHERLANDS     9865
===== GREECE          9615
===== BBC             9750
===== HAVANA          9525
===== CANADA          9655
=====> -

X E D I T  1 File
```

Starting with the current line, the SORT command arranges a specified number of lines according to **sort fields**. What's a sort field? Well, a file is a collection of related records. AUSTRALIA 6035 is one record in the shortwave file. The record is composed of two data elements. One data element is the place name - AUSTRALIA. The other data element is the frequency - 6035. The data element we want to work with is place names.

The sort field is determined by the longest data element in the file (in other words, the longest place name). Enter the following command so that you can see the entire file.

```
====> down 5
```

You can see the longest name is VOICE OF AMERICA. Looking at the scale, you can see that VOICE OF AMERICA begins in column one and goes to column sixteen. So the sort field is 1 to 16.



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 top line current.

Enter:

```
====> top
```

Now enter:

```
====> sort * 1 16
```

As you can see, the place names are now in alphabetical order.

```
RADIO WAVES A1 F 80 Trunc=80 Size=12 Line=0 Col=1 Alt=1
```

```
==== * * * Top of File * * *
|...+...1...+...2...+...3...+...4...+...5...+...6...+...7...
==== AUSTRALIA 6035
==== BBC 9750
==== CANADA 9655
==== GREECE 9615
==== HAVANA 9525
==== MOSCOW 9450
==== NETHERLANDS 9865
==== PEKING 9780
==== QUITO, EQUADOR 9560
====> -
```

```
X E D I T 1 File
```

Now you decide that it would be useful to have the file sorted numerically, from the lowest frequency to the highest. The SORT command can do this, too. You will need to know the beginning and ending columns of the sort field. All the frequencies are the same length, so it doesn't matter which one you use to count. Use the scale to help you determine the sort field. The frequency begins in column 20 and ends in column 23. So the sort field is 20 to 23.

Enter:

```
====> sort * 20 23
```

The file has been sorted numerically, from the lowest number to the highest. Let's look at the sorted file.

Enter:

```
====> down 5
```



Your file should look like this:

```
RADIO WAVES A1 F 80 Trunc=80 Size=12 Line=5 Col=1 Alt=2

==== * * * Top of File * * *
==== AUSTRALIA 6035
==== SPAIN 6090
==== MOSCOW 9450
==== HAVANA 9525
==== QUITO, EQUADOR 9560
|...+...1...+...2...+...3...+...4...+...5...+...6...+...7...
==== VOICE OF AMERICA 9590
==== GREECE 9615
==== CANADA 9655
==== BBC 9750
==== WEST GERMANY 9765
==== PEKING 9780
==== NETHERLANDS 9865
==== * * * End of File * * *

====> -

X E D I T 1 File
```

What if you wanted the numbers sorted in descending order, from highest to lowest? That's easy. Just 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.)

First enter a TOP command to make the top line current.

```
====> top
```

Then enter:

```
====> sort * d 20 23
```

↑
means "descending order"

Do you want to see the whole file on one screen?

Enter:

```
====> down 5
```

Your file looks like this:

```
RADIO WAVES A1 F 80 Trunc=80 Size=12 Line=5 Col=1 Alt=3
```

```
==== * * * Top of File * * *
==== NETHERLANDS 9865
==== PEKING 9780
==== WEST GERMANY 9765
==== BBC 9750
==== CANADA 9655
==== |...+...1...+...2...+...3...+...4...+...5...+...6...+...7...
==== GREECE 9615
==== VOICE OF AMERICA 9590
==== QUITO, EQUADOR 9560
==== HAVANA 9525
==== MOSCOW 9450
==== SPAIN 6090
==== AUSTRALIA 6035
==== * * * End of File * * *

====> _
```

```
X E D I T 1 File
```

Sorting Part of a File

In our exercises, we've sorted an entire file. You can also use the SORT command to sort part of a file. Let's review the command format:

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

You substituted an asterisk (*) for "number-of-lines" when you wanted to sort the entire file. But what if you want to sort only part of a file? Instead of substituting an asterisk for "number-of-lines," substitute the number of lines you want to sort.



For example, this command would sort 20 lines, beginning with the current line. The sort field is columns five to ten.

```
====> sort 20 5 10
```

Remember that the sort starts with the current line. If the lines you want to sort are somewhere in the middle of the file, remember to make current the line where you want the sort to begin.

If the number of lines is too long to count, you can use the SET NUMBER ON command discussed earlier in this chapter.

Now let's QUIT this file by pressing PF3 and entering QQ. None of the sorting is saved. The file is the way it was before we sorted it.

Now press the PF3 key to exit from the file list.

Getting Help

If you're editing a file and forget how to use a command, or you'd like to see information about XEDIT commands not covered in this book, press the PF1 key. The PF1 key is set to the HELP command.

When you press PF1, a "menu" appears on the screen. The menu is a list of all XEDIT commands.

XEDIT MENU Menu Help Information line 1 of 24
(c) Copyright IBM 1980, 1986 (adapted from IBM Form SC24-5221)

A subcommand may be selected for viewing by placing the cursor under any character of the subcommand name and then pressing the ENTER key or the PF1 key. A MENU file is indicated when a name is preceded by an asterisk (*). A TASK file is indicated when a name is preceded by a colon (:). For a description of the HELP operands and options, type HELP HELP.

&	CDelete	CP	Help	Next	REFRESH	SOS
*Prefix	CFirst	CRepLace	HEXType	NFind	REnum	SPLit
*QUERY	Change	CURsor	Input	NFINDUp	REPEat	SPLTJOIN
*SET	CInsert	DElete	Join	Overlay	ReplacE	STACK
?	CLast	Down	LEft	PARSE	RESet	STATus
:XEDIT	CLocate	DUPlicat	LOAD	POWerinp	REStore	TOP
=	CMS	EMSG	Locate	PREServe	RGTLEFT	TRAnSfer
Add	CMSG	EXpand	LOWercas	PURge	RIght	Type
PF1= Help	2= Top	3= Quit	4= Return	5= Clocate	6= ?	
PF7= Backward	8= Forward	9= PFkeys	10=	11=	12= Cursor	

====>

Macro-read 1 File

To find out information about a particular command, move the cursor under that command and press the ENTER key. (You can press the key marked \rightarrow to move the cursor quickly to the desired command name.) If the command you want is not listed on the first screen of this menu, press PF8 to go to the next screen. Then select the command as described above.

If the command information fills more than one screen, press PF8 to scroll to the next screen. To return to your file, press PF4.

If you forget how to do various editing tasks, you can use the XEDIT task menu. To see that menu, move the cursor under **:XEDIT** in the list and press ENTER. You can then select which task you'd like to see information about. Remember to press PF4 to return to your file.



Summary

You learned the following commands/PF keys in this section:

Command/PF key	Function
GET	Inserts all or part of one file in another.
SET NUMBER ON/OFF	Puts line numbers in the prefix area (ON) or removes them (OFF).
PF3 (QUIT) QQuit	Ends an editing session without saving any changes. Enter after warning message to quit when file is new or changed.
SORT	Arranges a file in alphabetical or numerical order, according to sort fields.
PF1 (HELP)	Displays a menu of all XEDIT commands.

Protecting Your Files

How can you protect your files from computer or communication line failure? First, you should understand that your files on disk are safe if the system malfunctions (or "goes down," in computer talk). It's the new file you're creating or the changes that you're making to an existing file that might be lost if the system fails. You can minimize this danger 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 the system goes down, all the work you completed before you entered the last SAVE (or FILE) command would be saved on disk. To redisplay your file when the computer begins to function again, you would enter the XEDIT command (or enter FILELIST and select the file from the list). Sometimes you don't even need to do that. The file might appear on the screen when the system comes back on. If you made any changes between the time you entered your last SAVE command and the time the system went down, you'll have to redo them.

If you've entered a SAVE command while working on a file, you still have to enter 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 would be:

```
====> 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 the XEDIT command.

If you've entered a SET AUTOSAVE command and the system goes down, 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 'nnnnnn AUTOSAVE A1'. The part of the message that reads "nnnnnn AUTOSAVE A1" is the new label. The filename will be a number (like 100001) 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. Here's how.

Suppose you were editing the file RADIO WAVES, and the system went down. When the system is available again, your file list might look like this.

```

yourid  FILELIST  AO  V 108  Trunc=108  Size=6  Line=1  Col=1  Alt=0
Cmd  Filename  Filetype  Fm  Format  Lrecl  Records  Blocks  Date  Time
-
100001  AUTOSAVE  A1 F      80      12      1  10/04/86  21:14:04
RADIO   WAVES     A1 F      80      12      1  10/04/86  21:12:04
PICNIC  INGREDTS A1 F      80      22      2  10/04/86  20:58:07
VACATION  CAMPING  A1 F      80       8      1  10/04/86  17:59:00
CRIME   THEFT    A1 F      80       4      1  10/02/86  15:33:05
STOCK   INVEN    A1 F      80       7      1  9/25/86   12:10:03

1= Help      2= Refresh  3= Quit     4= Sort(type)  5= Sort(date)  6= Sort(size)
7= Backward  8= Forward  9= FL /n    10=           11= XEDIT      12= Cursor

====>
X E D I T  1 File

```

Your current RADIO WAVES file would be stored in the file labeled 100001 AUTOSAVE A. You would use the DISCARD command to discard the original RADIO WAVES file. You would then use the RENAME command to rename the AUTOSAVE file from 100001 AUTOSAVE A to RADIO WAVES A. The DISCARD and RENAME commands are explained in Chapter 3.

If you have entered a SET AUTOSAVE command and you want to find out the file label of the AUTOSAVE file and how many changes have been made, you can enter the QUERY AUTOSAVE command. Its format is:

```
Query AUTosave
```

A message like the following is displayed.

```
AUTOSAVE 10; Filename 100001; Alterations : 3
```

"AUTOSAVE 10" is the number you specified in the SET AUTOSAVE command. "Filename 100001" means the filename for this



AUTOSAVE file is 100001. (The filetype is always AUTOSAVE.) If an AUTOSAVE file already exists, the editor picks the next available number. For example, if the file 100001 AUTOSAVE A exists, the editor would name this file 100002 AUTOSAVE A. "Alterations : 3" is the number of the changes that have been made. When the number of alterations reaches ten, the file is saved. Every time the file is saved, this number is set back to zero. The number of alterations is also shown in the file identification line (Alt=n).

With some exceptions, every line changed or added counts as an alteration. If you make a correction by typing over words in an existing line in an existing file, that counts as an alteration. On the other hand, all the changes caused by a single command count as one alteration no matter how many lines are affected.

If you enter a SET AUTOSAVE command while you're creating a file, and then enter QUIT, the file won't be saved. If, however, you've made enough alterations for an automatic save to occur, the AUTOSAVE file will be on your disk.

If you've entered a SET AUTOSAVE command while you're revising an existing file and then you enter QUIT, no revisions will be saved. But if you've made enough alterations for an automatic save to occur, the AUTOSAVE file will be on your disk.

Summary

You learned the following commands in this section:

Command	Function
SAVE	Writes the file you're working on to disk.
SET AUtosave	Automatically issues a SAVE periodically.
Query AUtosave	Displays autosave information.



More About Editing

You have already learned all the XEDIT commands you need to know to work with files. In this section, you'll learn a few more commands that are not only useful, but they can make your job easier. You'll also learn how to create a special file containing commands that tailor your editing sessions to suit your needs.

Power Typing

If you're a good typist, and if you have a large amount of text to enter, power typing is for you! 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
```

Let's create a file to practice power typing. Follow the logon procedure if you aren't already logged on.

Then enter:

```
x marigold story
```

We want the data to be entered in the file in both capital and lowercase letters, so enter:

```
====> set case m
```

Now enter:

```
====> pow
```

Notice that the whole screen is available for typing. The cursor is positioned at the top of the screen. That is where you should start typing the following data. If you come to the end of a line and you're in the middle of a word, just keep on typing. The cursor will move to the beginning of the next line. Later, you'll see that the editor adjusts the text so that only full words are on each line.

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. Now type:

```
Marigolds are annual plants. They are fast growing and quick to produce plentiful flowers. They do not survive over the winter season in most climates. Each year new plants must be started from seeds.#Marigolds produce a profusion of blooms in color ranging from light yellow to deep rust. Some marigold varieties have flowers with two or more colors within the bloom.#Many gardeners use marigolds as border or bedding plants for continuous summer color. Marigolds also make excellent cut flowers.
```

You could fill the whole screen if you wanted to. When you run out of space on the screen, simply 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. Do that now.

Then enter:

```
====> top
```




Your screen looks like this:

```
MARIGOLD STORY  A1 F 80  Trunc=80 Size=8 Line=0 Col=1 Alt=8

=====
|...+...1...+...2...+...3...+...4...+...5...+...6...+...7...
===== Marigolds are annual plants. They are fast growing and quick to produce
===== plentiful flowers. They do not survive over the winter season in most
===== climates. Each year new plants must be started from seeds.
===== Marigolds produce a profusion of blooms in color ranging from light
===== yellow to deep rust. Some marigold varieties have flowers with two or
===== more colors within the bloom.
===== Many gardeners use marigolds as border or bedding plants for continuous
===== summer color. Marigolds also make excellent cut flowers.
===== * * * End of File * * *
=====>
-
X E D I T 1 File
```

As you can see, the editor has adjusted the text so that only full words are on each line. And the sentences you preceded with a pound sign each begin on a new line.

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, just press the insert mode key. You don't have to also press the PA2 key, as you did when in regular input mode or edit mode. Remember to press the RESET key when you are finished using insert mode.

Splitting the Screen

Sometimes, you'd like to be able to look at different parts of a file at the same time, or even at different files at the same time.

You can use the SET SCREEN command to divide the physical screen into multiple logical screens. Each logical screen looks like a separate physical screen, except, of course, it's smaller.

A format of the SET SCREEN command is:

```
SET SCReen n
```

You substitute the number of screens you want for "n." Usually, you want only two or three screens, because the more logical screens, the fewer number of lines in each screen.

Multiple Views of the Same File

Let's practice splitting the screen.

Enter:

```
====> set scr 2
```



Two identical views of the same file appear:

```
MARIGOLD STORY   A1  F 80  Trunc=80 Size=8 Line=0 Col=1 Alt=8

==== * * * Top of File * * *
|...+...1...+...2...+...3...+...4...+...5...+...6...+...7...
==== Marigolds are annual plants. They are fast growing and quick to produce
==== plentiful flowers. They do not survive over the winter season in most
==== climates. Each year new plants must be started from seeds.
====> _
                                X E D I T  1 File
MARIGOLD STORY   A1  F 80  Trunc=80 Size=8 Line=0 Col=1 Alt=8

==== * * * Top of File * * *
|...+...1...+...2...+...3...+...4...+...5...+...6...+...7...
==== Marigolds are annual plants. They are fast growing and quick to produce
==== plentiful flowers. They do not survive over the winter season in most
==== climates. Each year new plants must be started from seeds.
====>
                                X E D I T  1 File
```

You can type over the data and enter commands (and prefix commands) in either screen. Any changes you make appear in both screens. But commands that control the screen display, such as UP, DOWN, TOP, or BOTTOM, affect only the screen from which you enter them. When you use a PF key to move the display, such as PF7 (backward) or PF8 (forward), the screen that is affected is the one that contains the cursor when you press the PF key.

When you decide you've had enough of seeing double, enter:

```
====> set scr 1
```

Multiple Views of Different Files

Now let's look at two different files at the same time.

Enter:

```
====> set scr 2
```

Your cursor is on the command line of the top screen. Enter the XEDIT command to call out a second file:

```
====> x vacation camping
```

Two different files are displayed.

```
VACATION CAMPING  A1  F 80  Trunc=80 Size=8 Line=0 Col=1 Alt=0

.

==== * * * Top of File * * *
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7...
==== A CLEAN, CLEAR STREAM TO FISH IN
==== WOOD NEARBY FOR COOKING
==== NO WILD BEARS
====> -
                                     X E D I T  2 Files
MARI GOLD STORY   A1  F 80  Trunc=80 Size=8 Line=0 Col=1 Alt=8

.

==== * * * Top of File * * *
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7...
==== Marigolds are annual plants. They are fast growing and quick to produce
==== plentiful flowers. They do not survive over the winter season in most
==== climates. Each year new plants must be started from seeds.
====>
                                     X E D I T  2 Files
```

Having two different files displayed is like having two terminals. You edit each file independently. A command entered from one screen does not affect the other file.



Notice the status area in the lower right corner. It reminds you that you are editing two files at the same time. When you QUIT or FILE one of the files, the screen remains split. You will then have two views of the same file. Keeping the cursor in the top screen, press PF3 to QUIT. Remember the command to put the screen back to one view?

Then enter:

```
====> set screen 1
```

Now enter:

```
====> file
```

Creating an XEDIT Profile

You have seen that 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 a special file. 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 lines.

```
/* Sample PROFILE XEDIT file */  
SET CASE MIXED  
SET AUTOSAVE 20  
SET TABS 10 20 30 40  
SET PF12 FILE
```

The first line in this file must be a comment. A comment must start with */** and end with **/*, but the words in between can be anything you like.

Each command must be typed on a separate line. 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. For example, PF3 is set to the QUIT command, PF7 is set to the BACKWARD command, etc. 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.

You must put a separate SET PF command in the file for each PF key setting you want to change.

(You can also enter a SET PF command from the large arrow command line, during an editing session. But in this case, the PF key setting is changed for that editing session only.)

Of course, you can change anything in your PROFILE XEDIT file just as you can in any other file.

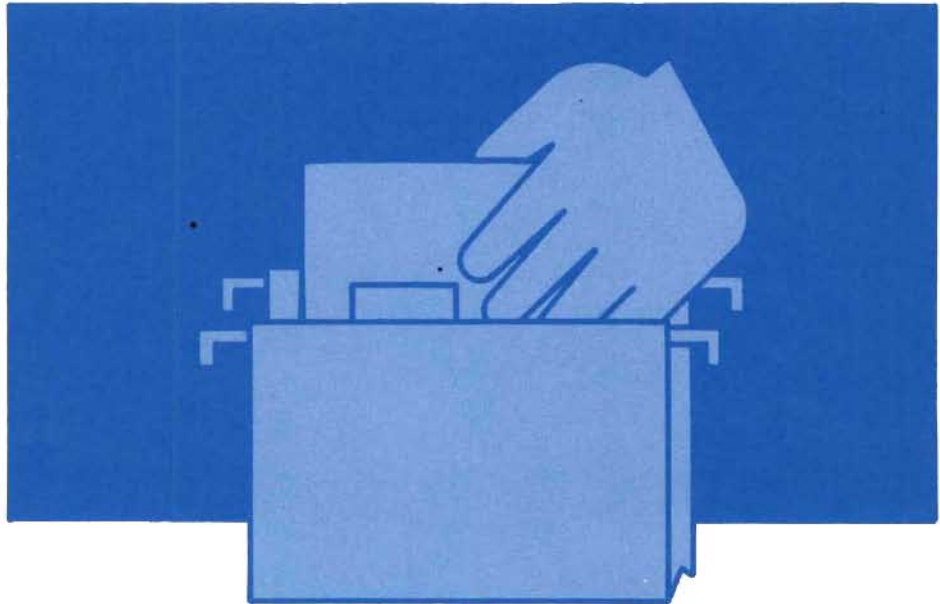


Summary

You learned the following commands/PF keys in this section:

Command/PF key	Function
POWer	Enters power typing mode.
SET SCReen	Splits the screen for multiple views of the same or different files.
SET PFn	Changes the setting for a PF key.

Chapter 3. Managing Your Files



In Chapter 2, you learned how to 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 chapter make it easy for you to manage your files, no matter how many you have.

This chapter teaches you how to:

- Display a list of your files.
- Discard or rename a file.
- Compare or copy files.
- Use another person's files.

More information on the commands discussed in this chapter can be found in the *VM/SP CMS Command Reference* and the *VM/SP CP Command Reference*.

What Files Do I Have?

If you can't remember the exact name of a file you want to work with, you can use the FILELIST command. (If you read the previous chapter, you are already familiar with FILELIST, but let's review it here.) The FILELIST command displays a list of the files you have on your disk. Its format is:

```
FILEList
```

Remember that every file has a three-part label: filename, filetype, and filemode. 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 command displays 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, and so forth.)



If you did the exercises in Chapter 2, your file list looks similar to this:

```
1 — yourid  FILELIST AO V 108 Trunc=108 Size=6 Line=1 Col=1 Alt=0
2 — Cmd  Filename Filetype Fm Format Lrecl  Records  Blocks  Date      Time
      STOCK   INVEN   A1 F      80        7        1  9/14/86  14:48:20
      MARIGOLD STORY  A1 F      80        8        1  9/14/86  14:45:02
      PICNIC   INGREDTS A1 F      80       22        2  9/14/86  13:27:28
      VACATION CAMPING  A1 F      80        8        1  9/14/86  10:50:59
      CRIME    THEFT    A1 F      80        4        1  9/14/86  10:45:19
      RADIO    WAVES    A1 F      80       12        1  9/14/86  10:35:00

3 — 1= Help      2= Refresh   3= Quit     4= Sort(type) 5= Sort(date) 6= Sort(size)
      7= Backward 8= Forward  9= FL /n   10=           11= XEDIT     12= Cursor
      =====>

                                  X E D I T  1 File
```

The file list shows a lot of information about the files on this person's 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 be greater than one. The other fields aren't important to you now.
- 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

These fields give information about the file format but 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. We will discuss them throughout this chapter. 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.

Note: If your terminal has 24 PF keys, PF keys 13 to 24 have the same functions as PF keys 1 to 12 as discussed here.



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 can use PF keys to "scroll" through it. Scrolling is like turning the pages of a book. 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:

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/CRIME THEFT/
```

If CRIME THEFT is located, the line containing it becomes the first line on the screen. If it isn't found, a message will tell you so.

Variations on the FILELIST Command

You've seen how to use the FILELIST command to display a list of **all** the files on your A-disk. You can also use FILELIST to display only **some** of your 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, and so forth.

The FILELIST command has the format:

```
FILEList filename filetype filemode
```

If you don't substitute a letter for "filemode," a filemode of A is assumed. In our examples, all the files are on the A-disk.

Suppose you have several files that concern stock inventory. You know that they all have the same filename, STOCK, but you don't



remember the different filetypes. To see a list of all the files on your A-disk with the filename STOCK, you would enter:

```
FILEL STOCK *
```

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

The file list might look like this:

```
SOMEONE      FILELIST      A0  V 108  Trunc=108  Size=6  Line=1  Col=1  Alt=0
Cmd  Filename  Filetype  Fm  Format  Lrecl  Records  Blocks  Date      Time
STOCK  BOLTS     A1  V      95      34       2  10/04/86  21:15:04
STOCK  GADGETS   A1  V      95      34       2  10/04/86  21:12:04
STOCK  NUTS      A1  V      95      29       2  10/04/86  20:58:07
STOCK  SCREWS    A1  V     107     281     10  10/04/86  17:59:00
STOCK  SURPLUS   A1  V      92     101     4  10/02/86  15:33:05
STOCK  WIDGETS   A1  V      75      28       1  9/25/86   12:10:03

1= Help      2= Refresh   3= Quit      4= Sort(type)  5= Sort(date)  6= Sort(size)
7= Backward  8= Forward   9= FL /n    10=           11= XEDIT      12= Cursor

====> _

X E D I T  1 File
```

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

```
FILEL * DATA
```

The list would contain all files whose filetype is DATA.

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 a list of files containing the file you want.

For example, suppose you know you have a number of files whose file names all start with COM and whose filetype is DATA. To list all those files, you would enter:

```
FILEL COM* DATA
```

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

```
COMBAT    DATA  A
COMMERCE  DATA  A
COMPARE   DATA  A
COMPOST   DATA  A
COMPUTER  DATA  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. For example:

The command:

```
FILEL *DATA DAILY1
```




might show these files:

```
HISDATA    DAILY1  A
HERDATA    DAILY1  A
MYDATA     DAILY1  A
YOURDATA   DAILY1  A
```

And this command:

```
FILEL *DATA DAILY*
```

might list these files:

```
HISDATA    DAILY1  A
HERDATA    DAILY1  A
MYDATA     DAILY1  A
YOURDATA   DAILY1  A
HISDATA    DAILY2  A
YOURDATA   DAILY2  A
ITSDATA    DAILY2  A
```

When you enter a FILELIST command and do not specify a filemode, a filemode of A is assumed. If you want the list to contain files on any other of your disks, you must give that filemode. For example:

```
FILEL * DATA B
```

lists all the files on your B-disk whose filetype is DATA. Similarly:

```
FILEL * DATA *
```

lists all files whose filetype is DATA on all your accessed disks.

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.

To create a file, you type an XEDIT command (XEDIT filename filetype) on the command line next to the large arrow at the bottom of the file list. Then press the ENTER key. When you are finished creating the file and enter a FILE command, the file list returns. However, your new file won't appear in the list until you **press PF2**. PF2 is set to the "refresh" function. This means that the list is refreshed with new information.



Discarding a File

Let's say you want to get rid of some files you've created. If they were on paper, you'd tear them up and throw them away. But how do you destroy files that are cluttering up your disk? You can use the DISCARD command.

```
SOMEONE FILELIST  A0  V 108  Trunc=108  Size=6  Line=1  Col=1  Alt=0
Cmd  Filename  Filetype  Fm  Format  Lrecl  Records  Blocks  Date  Time
STOCK  BOLTS  A1  V  95  34  2  10/04/86  21:15:04
STOCK  GADGETS  A1  V  95  34  2  10/04/86  21:12:04
discardTOCK  NUTS  A1  V  95  29  2  10/04/86  20:58:07
discardTOCK  SCREWS  A1  V  107  281  10  10/04/86  17:59:00
STOCK  SURPLUS  A1  V  92  101  4  10/02/86  15:33:05
STOCK  WIDGETS  A1  V  75  28  1  9/25/86  12:10:03

1= Help      2= Refresh  3= Quit      4= Sort(type)  5= Sort(date)  6= Sort(size)
7= Backward  8= Forward  9= FL /n    10=           11= XEDIT     12= Cursor

====>

X E D I T  1 File
```

To discard a file, you

- Type DISCARD in the "Cmd" space next to the file (or files) you no longer want (see above). If the command name is longer than the space provided, just type right over the information already in that line. In the screen above, the last letter in "discard" is typed over the first letter in the filename (the "S" in "STOCK").
- Press the ENTER key to execute the command.

In the example above, two files would be discarded. After the ENTER key is pressed, a notice appears in the line, telling you the file has been discarded. The lines would then look like this:

```
*      STOCK      NUTS      A1 has been discarded.  
*      STOCK      SCREWS    A1 has been discarded.
```

You must press the ENTER key to execute the DISCARD commands typed on one screen before scrolling to another screen.

If you have many files to discard, you don't have to type out the word "discard" next to each file. You can take a shortcut.

```
SOMEONE FILELIST AO V 108 Trunc=108 Size=6 Line=1 Col=1 Alt=0  
Cmd  Filename Filetype Fm Format Lrecl Records Blocks Date Time  
STOCK BOLTS A1 V 95 34 2 10/04/86 21:15:04  
discard STOCK GADGETS A1 V 95 34 2 10/04/86 21:12:04  
= STOCK NUTS A1 V 95 29 2 10/04/86 20:58:07  
= STOCK SCREWS A1 V 107 281 10 10/04/86 17:59:00  
= STOCK SURPLUS A1 V 92 101 4 10/02/86 15:33:05  
= STOCK WIDGETS A1 V 75 28 1 9/25/86 12:10:03  
  
1= Help 2= Refresh 3= Quit 4= Sort(type) 5= Sort(date) 6= Sort(size)  
7= Backward 8= Forward 9= FL /n 10= 11= XEDIT 12= Cursor  
====> XEDIT 1 File
```

Suppose the only file you want to keep is STOCK BOLTS. To discard the rest, you need type DISCARD only once on a screen. Just type an equal sign (=) in the "Cmd" space next to the other files you want to discard. The equal sign operates only on the lines following the command (see above). It means that you want the same command that you typed to be carried out for this file, too. Then press the ENTER key. The screen would look like this:



```
SOMEONE FILELIST A0 V 108 Trunc=108 Size=6 Line=1 Col=1 Alt=20
Cmd  Filename Filetype Fm Format Lrecl Records Blocks Date Time
* STOCK BOLTS A1 V 95 34 2 10/04/86 21:15:04
* STOCK GADGETS A1 has been discarded.
* STOCK NUTS A1 has been discarded.
* STOCK SCREWS A1 has been discarded.
* STOCK SURPLUS A1 has been discarded.
* STOCK WIDGETS A1 has been discarded.

1= Help      2= Refresh  3= Quit     4= Sort(type)  5= Sort(date)  6= Sort(size)
7= Backward  8= Forward  9= FL /n   10=           11= XEDIT     12= Cursor

====> _
```

X E D I T 1 File

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

What if you decide that a file label isn't quite right? Perhaps it's too similar to the label of another file. Or maybe you want to create a new file with that label. You can change the filename or filetype (but not the filemode) 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 the ENTER key. 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 the ENTER key to carry out the RENAME command.

Let's say you want to rename VACATION CAMPING A to GREAT OUTDOORS A. (If you did the exercises in Chapter 2, you can rename your file by following the instructions below.)

This line in the file list looks like this:

```
VACATION CAMPING A1 F      80      8      1  9/14/86 14:48:20
```

You type the RENAME command right over this line, as shown below. Remember to type all **three** parts of the new file label: filename filetype filemode.

```
rename / great outdoors a1 F      80      8      1  9/14/86 14:48:20
```

stands for VACATION CAMPING A1

When you press the ENTER key, your command is executed. It is replaced with this notice:

```
*      VACATION CAMPING A1 ** Discarded or renamed **
```

The notice means that your file VACATION CAMPING A has been renamed. Its new label, GREAT OUTDOORS A, replaces the old label in the file list when you **press PF2**. PF2 is set to the "refresh" function. This means that the list is "refreshed" with new information. In this case, the new information is the new file label.



Copying a File

Sometimes you may want to make changes in a file, but you want to keep both the original file and the revised one. 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 the ENTER key 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 the ENTER key to carry out the COPYFILE command.

For example, you have a file RADIO WAVES A. You want to make changes in the file, but you want to keep the original. (If you did the exercises in Chapter 2, you can copy your file by following the instructions below.) The line in the file list looks like this:

```
RADIO WAVES A: F      80      12      1  9/14/86 10:35:00
```

You want to make a copy of this file and call the copy RADIO NEWAVES A. You would type the following command right over this line:

```
copyfile / radio newaves a F 80 12 1 9/14/86 10:35:00
```

stands for RADIO WAVES A1

newfilename

newfilemode

newfiletype

When the ENTER key is pressed, the COPYFILE command is executed. To see the entry for the new file in your file list, **press PF2**. The new file appears at the top of the file list. Now you have two copies of the same file. Each has its own file label. You can make changes in RADIO NEWAVES, and still have the original file, RADIO WAVES.

If you have two copies of the same file, you might want to compare them. You could use the editor to look at each file, noting the differences yourself. Or you can use the COMPARE 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 compare. Press the ENTER key 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 the ENTER key to carry out the COMPARE command.

For example, suppose you made changes to the file RADIO NEWAVES A, and you want to compare it with RADIO WAVES A. The line in the file list looks like this:

```
RADIO    NEWAVES A1    F            80      12      1 10/26/86 12:10:03
```

Here’s how your COMPARE command would look:

```
compare / radio waves a    F    80    12    1 10/26/86 12:10:03
```

Diagram illustrating the components of the COMPARE command:

- stands for RADIO NEWAVES A (points to the slash /)
- filename of file to be compared (points to radio waves a)
- filetype of file to be compared (points to F)
- filemode of file to be compared (points to 80)

After you press the ENTER key, this message appears on a cleared screen:

```
Comparing RADIO NEWAVES A with RADIO WAVES A
```

If the files are identical, nothing else is displayed. The status area (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 (in this case, RADIO NEWAVES A), 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 file *filename filetype filemode*. The screen will display MORE... in the status area. Press the CLEAR key to return to the file list. A number will appear next to the asterisk in the "Cmd" space (*40). This number is a return code; you can ignore it.

Using Other People's Files

One way to look at another person's files is to ask him to send you a copy for you to store on your disk. You can learn how to do this in Chapter 5 ("Receiving Notes and Files" section).

You can use another method if the files you want to see are very large and would take up a lot of space on your disk. You can use this method only if the other person is using the same computer that you are.

The commands discussed here can be entered at the bottom of the file list (next to the large arrow), or on the CMS command line. For purposes of illustration, let's assume you're entering the commands on the file list large arrow command line.

First, you need to understand more about how your files are stored. You can think of all your files as being stored in a filing cabinet that has several drawers. The top drawer is the "A" drawer. You have other drawers labeled with different letters, but the files you create and change are in the A-drawer. We call this collection of files your **A-disk**. You already know that all the files on the A-disk have a filemode of A.

The A-disk also has a three-digit number, like 191. Normally, you don't need to know this number. But you will need to know it in order to use the commands that allow you to see another person's files.



To find out what disk letters and disk numbers you have, enter the following command in the large arrow command line.

```
====> Query SEARCH
```

This command clears the file list from the screen and displays the disk numbers and filemode letters that are assigned to you. Your display might look like this:

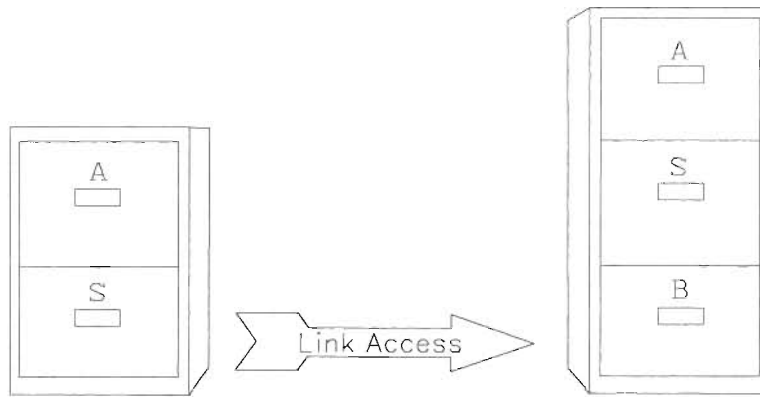
```
USE191  191  A  R/W  
LABEL1  190  S  R/O
```

The screen will display a MORE... notice. Press the ENTER key to hold this information on the screen. (The status area will then display the notice HOLDING.) Each line gives information about the "drawers" (disks) in your filing cabinet. The first field is the disk label. It's not important now. The next field is the disk number. The third field is the filemode letter. The last field isn't important now. Write down the disk numbers and letters. You'll need to know them to use the commands discussed below. Now you press the CLEAR key to get the file list back on the screen.

Before using someone's files, you have to ask him for three things:

- His userid
- The number of his disk that has the files you want to see
- His "read" password. (This password is different from the one he uses when he logs on.)

The LINK and ACCESS commands, discussed below, temporarily add another "drawer" to your filing cabinet. They allow you to look at the files in this new drawer.



First you enter a LINK command, using the information you got from the other person. For example:

```
====> LINK TO JACKIE 191 196 RR
```

JACKIE → his userid
 191 → his disk number
 196 → a disk number not in your list
 RR → means "read only"

In the example above, JACKIE is the other person's userid. 191 is his disk number that he told you. 196 is a disk number that you select. You can use any three-digit number that didn't appear in the list when you entered QUERY SEARCH. RR means that this new disk is "read only." You can look at the files but not change them. Later, you'll learn what to do if you decide you want to change one or more of the files.

When you press the ENTER key, this message appears on a cleared screen: ENTER READ PASSWORD. You type his read password, and press the ENTER key. The screen displays MORE... in the status area. Press the CLEAR key to get the file list back on the screen.



Then you enter an ACCESS command:

```
====> ACCESS 196 B
```

 | |
 | |→ next available disk letter
 | |→ your disk number you used in the LINK
 | | command

The disk letter can be any letter that didn't appear in the list when you entered QUERY SEARCH. It is best to choose an available letter closest to the beginning of the alphabet. In our example, the QUERY SEARCH command showed that disk letters A and S were in use. Therefore, the best letter to select is B. When you enter the ACCESS command, a message like the one below is displayed on a cleared screen:

```
B (196) R/O
```

This simply means that you now can read the files on this new (B) disk.

To see a list of files in this new "drawer," enter:

```
====> FILELIST * * B
```

 |
 | the letter you used in the
 | ACCESS command

This file list contains all files on the B-disk. To look at the contents of a file, move the cursor to the line that describes it, and press the PF11 (XEDIT) key.

Can you change these files? Yes, but first you must copy each file you want to change onto your A-disk. In other words, you must put a copy of that file(s) in your A-drawer. You can use the COPYFILE command to do this. In the COPYFILE command, you would change the filemode from the letter you specified in the ACCESS command to "A". For example, if one of the files on the B-disk is GOOD

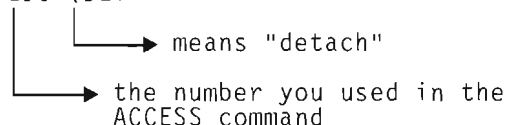
STUFF, you can type this command in the "Cmd" space next to that file:

```
COPYFILE / GOOD STUFF A
```

The COPYFILE command puts a copy of the file GOOD STUFF on your A-disk. When you have used COPYFILE to copy all the files you want to change to your A-disk, you can press PF3 to get your A-disk file list back on the screen. Then press PF2 to refresh the screen. The file GOOD STUFF is the first entry in the file list. Now you can use the file GOOD STUFF as if you had created it yourself. You can make changes to it using XEDIT, discard it, and so forth.

When you no longer want access to the other person's disk, enter the command

```
====> RELEASE 196 (DET
```



means "detach"

the number you used in the ACCESS command

The message DASD 196 DETACHED is displayed on a cleared screen. (Press the CLEAR key to get your file list back on the screen.) Your LINK and ACCESS commands are good only for one terminal session. The next time you log on, you'll have to enter these commands again to gain access to the other person's files.

Press PF3 to leave your file list.



Summary of How to Share Files

To Gain Access to Another Person's Files:

- Ask other person for his userid, disk number, and read password.
- Enter a QUERY SEARCH command.
Write down your disk numbers and filemode letters displayed.
- Enter a LINK command.
- Enter an ACCESS command.

To See a List of These Files:

- Enter a FILELIST command.

To Make Changes in these Files:

- Use the COPYFILE command (on the file list of these files).

To End Access to these Files:

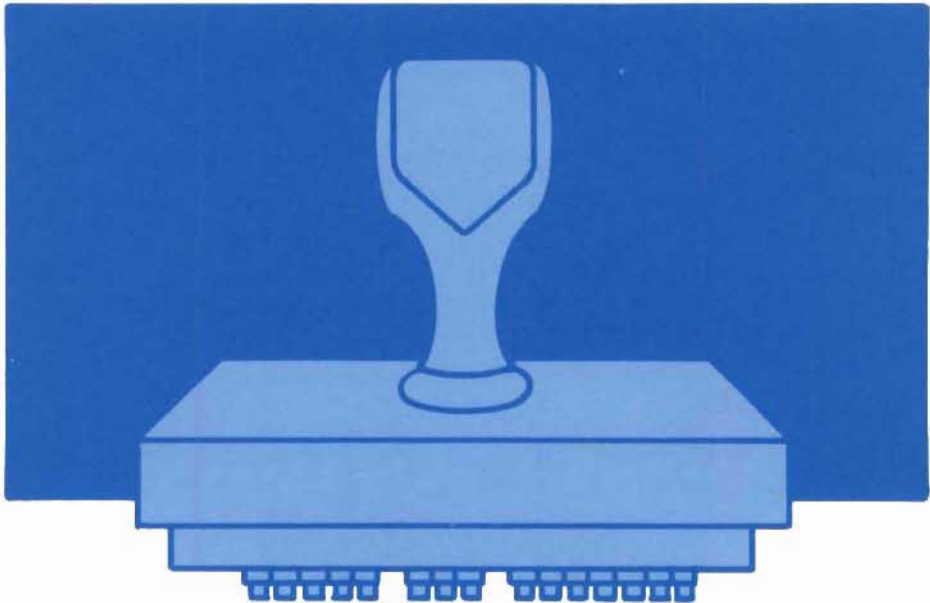
- Enter a RELEASE command.

Summary

You learned the following commands in this chapter:

Command	Function
FILEList	Displays a list of your files.
DISCARD	Discards a file.
Rename	Renames a file.
COPYfile	Copies a file.
COMpare	Compares two files.
Query SEARCH	Displays your disk numbers and letters.
LINK, ACcess	Lets you see someone else's files.
RELease	Ends access to someone else's files.

Chapter 4. Printing Files



With VM, it's possible to manage your files and never use a piece of paper. But what if ten different people need to read one of your files? Unless you want to send them copies for their permanent disk storage (discussed in Chapter 5), you'll want to have ten copies printed on paper. To have a file printed, you don't need to have it displayed on your screen.

This chapter teaches you how to:

- Tell the system how many copies you want printed.

- Use the PRINT command to print any file.
- Use the SCRIPT command to format and print SCRIPT files.

For more information on the PRINT command, see the *VM/SP CMS Command Reference*. For more information on the SCRIPT command, see the *Document Composition Facility: Generalized Markup Language Starter Set User's Guide*.

Where Do I Enter These Commands?

The PRINT and SCRIPT commands can be entered either on the CMS command line, or directly on a FILELIST screen. Both ways are explained here.

PRINT Command

To print one copy of any file, your command (entered on the CMS command line) would have the following form:

```
PRINT filename filetype filemode
```

When you press the ENTER key, your command moves up on the screen and you get a ready message. Your file is now at the printer.

If you want to print more than one copy of a file, you'll need two commands. Suppose you want to print three copies of a file. Your first command would be:

```
SPOOL PRINT COPY 3
```



When you press the ENTER key, the command moves up on the screen and you get a ready message. Then you enter the PRINT command:

```
PRINT filename filetype filemode
```

When you press the ENTER key, you get a message similar to the following:

```
PRT FILE 765 FOR userid COPY 003 NOHOLD.
```

The number 765 is the number that VM assigned to the file in the example. Your userid is listed in the message. COPY 003 means that you will get three copies. NOHOLD isn't important. The message means your file is at the printer.

If you want to print more files during the same terminal session and you need three copies of each, you don't have to enter a new SPOOL PRINT command. Just enter a PRINT command for each file. But if you need a different number of copies, you'll have to enter a new SPOOL PRINT command indicating a different number of copies (for example, SPOOL PRINT COPY 1).

Using PRINT on the FILELIST Screen

You can enter the PRINT command on a FILELIST screen, in the "Cmd" space next to the file you want printed.

To print one file, simply type PRINT in the "Cmd" space next to the file you want printed. Then press the ENTER key.

To get more than one copy, first enter a SPOOL PRINT COPY n command. Type it on the large arrow command line (at the bottom of the file list) and then press the ENTER key.

Here's a sample FILELIST screen with a PRINT command typed next to a file:

```
SNOWHITE FILELIST A0 V 108 Trunc=108 Size=10 Line=1 Col=1 Alt=0
Cmd  Filename Filetype Fm Format Lrecl Records Blocks Date Time
ALL   NOTEBOOK A0 V      95      34      2 10/04/86 21:15:04
WISTFUL DATA A1 V      95      34      2 10/04/86 21:12:04
BOSS  DATA A1 V      95      29      2 10/04/86 20:58:07
DUMMY DATA A1 V     107     281     10 10/04/86 17:59:00
print GROUCHY DATA A1 V      92     101      4 10/02/86 15:33:05
PRINCE DATA A2 V      75      28      1  9/25/86 12:10:03
SNOOZY DATA A2 V     120     277     10  9/24/86  9:14:02
SNIFFLES DATA A1 V      26       7      1  9/23/86  8:50:06
MINING SCRIPT A1 V     132     489     30  8/26/86  7:05:08
KITCHEN SCRIPT A1 V     132     489     30  8/26/86  6:05:08

1= Help      2= Refresh  3= Quit     4= Sort(type) 5= Sort(date) 6= Sort(size)
7= Backward 8= Forward  9= FL /n   10=          11= XEDIT    12= Cursor

====>

X E D I T 1 File
```

When you press the ENTER key, a message like this appears on a cleared screen: PRT FILE 785 FOR SNOWHITE COPY 001 NOHOLD. This message means that the file GROUCHY DATA is at the printer.

Press the CLEAR key to get the list back on the screen. An asterisk (*) appears in the "Cmd" space. If you wanted to print this file again, you would just type PRINT right over the asterisk, and press the ENTER key.

To print more than one file, type PRINT in the "Cmd" space next to each file you want printed. Then press the ENTER key.

You must press the ENTER key to execute all PRINT commands typed on one screen before scrolling to another screen.



SCRIPT Command

In Chapter 7, you learn how to create SCRIPT files containing formatting instructions called GML tags. When you want to print such a file, you can use the following command:

```
SCRIPT filename (PROFile(DSMPROF3) CONTINUE PRINT
```

The first part of the command, SCRIPT, tells VM to use the Script formatter. The second part is the filename of the file you want printed (the filetype must be SCRIPT). PROFile(DSMPROF3) means a special file will handle the GML tags. For DSMPROF3, you may need to substitute the name of a different profile. Your system administrator can tell you whether you need to make a substitution. CONTINUE tells Script to continue processing your file unless it finds a severe error. PRINT means you want the file printed on your system printer.

If you want to print more than one copy of a Script file, you'll need two commands. Suppose you want five copies of a file. Your first command would be:

```
SPOOL PRINT COPY 5
```

When you press the ENTER key, this command moves up on the screen and you get a ready message. Then you enter the SCRIPT command. When you press the ENTER key, VM gives you a message similar to the following: PRT FILE 333 FOR userid COPY 005 NOHOLD. The number 333 is the number that VM assigned to the file in our example. Your userid appears in the message. COPY 005 means that you will get five copies, as you requested in the SPOOL PRINT command. NOHOLD isn't important to you. The message means that your file is now at the printer.

Using SCRIPT on the Filelist Screen

You can enter the SCRIPT command on a FILELIST screen, in the "Cmd" space next to the file you want printed.

If you want only one copy of each file you print, you don't need a SPOOL PRINT command. But if you want more than one copy, you enter a SPOOL PRINT command on the large arrow command line, at the bottom of the file list.

To print one file, you begin typing the SCRIPT command in the "Cmd" space next to the file you want printed. The command has the form:

```
SCRIPT / (PROFile(DSM PROF3) COntinue PRint
```

SCRIPT is the command name. The slash (/) stands for the file on that line. You then type the left parenthesis and the options. (Remember that you may need to substitute the name of another profile for DSM PROF3.) Then press the ENTER key.

Here is a line from the filelist screen shown previously:

```
MINING  SCRIPT  A1 V      132      489      30  8/26/86  7:05:08
```



Here's how that line looks on a filelist screen with a SCRIPT command typed over it:

```
SNOWHITE FILELIST AO V 108 Trunc=108 Size=10 Line=1 Col=1 Alt=3
Cmd  Filename Filetype Fm Format Lrecl Records Blocks Date Time
ALL  NOTEBOOK AO V          95      34      2 10/04/86 21:15:04
WISTFUL DATA A1 V          95      34      2 10/04/86 21:12:04
BOSS  DATA A1 V          95      29      2 10/04/86 20:58:07
DUMMY DATA A1 V         107     281     10 10/04/86 17:59:00
* GROUCHY DATA A1 V          92     101      4 10/02/86 15:33:05
PRINCE DATA A2 V          75      28      1  9/25/86 12:10:03
SNOOZY DATA A2 V         120     277     10  9/24/86  9:14:02
SNIFFLES DATA A1 V          26       7      1  9/23/86  8:50:06
script / (prof(dsmprof3) co pr 132 489 30 8/26/86 7:05:08
KITCHEN SCRIPT A1 V         132 489 30 8/26/86 6:05:08

1= Help      2= Refresh  3= Quit     4= Sort(type) 5= Sort(date) 6= Sort(size)
7= Backward 8= Forward  9= FL /n   10=          11= XEDIT    12= Cursor

====>

X E D I T 1 File
```

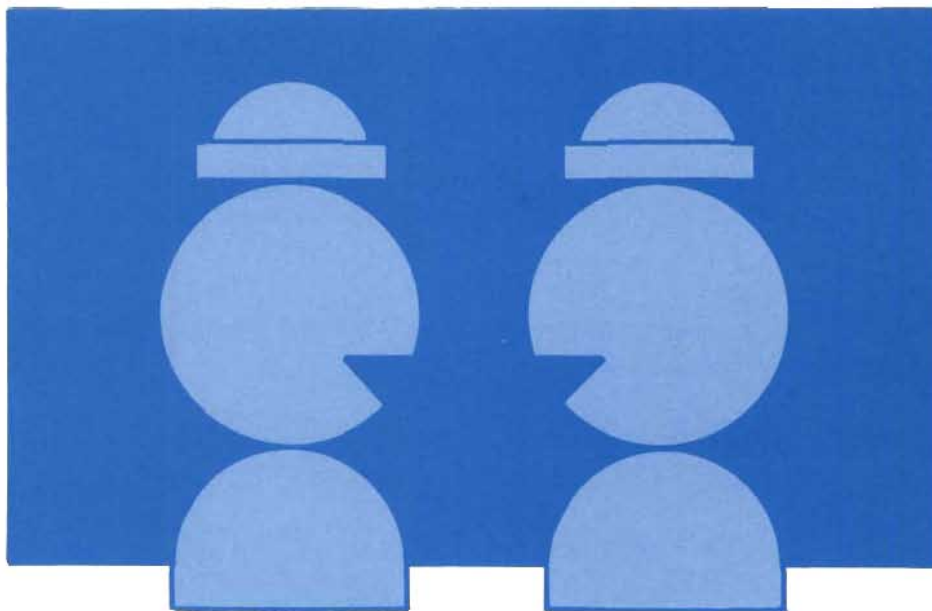
Note: To display the formatted file at your terminal instead of printing it, just omit the "PR" option on the SCRIPT command.

Summary

You learned the following commands in this chapter:

Command	Function
SPOOL	Specifies number of copies to be printed.
PRint	Prints a file.
SCRIPT	Formats and prints a SCRIPT file.

Chapter 5. Communicating With Other People (Who Use Computers)



Communicating by computer means sending and receiving messages, notes, and files. You can communicate with people who share your computer, or who use a computer that is connected to yours.

This chapter teaches you how to:

- Put information about the people with whom you communicate in a special file of **names**.

- Send **messages** to someone who is logged on to a computer.
- Prepare and send **notes** to one person or to a number of people on a list.
- Send **files** to other people.
- **Receive** notes and files from other people.

You should know how to edit files using XEDIT (Chapter 2) before reading this chapter.

For more information on the commands discussed in this section, see the *VM/SP CMS Command Reference*.

Keeping Track of Names

You need to know two things about another user to communicate with him:

- His **userid** (user identification)
- His **node** (computer identification).

The node of a computer is similar to an area code. A userid is similar to a phone number.

To find out a person's userid, ask him. (A userid is sometimes called a logon id.)

If a person shares your computer system, his node is the same as yours. If he uses a computer that is connected to yours, his node is different from yours.



You're probably thinking that's a lot of information to keep track of. But you can keep all this data in a special file, called a names file.

What Is a Names File?

A names file is your personal directory for computer communication. It contains the userids and nodes of the people you communicate with. It can also contain other information, like mailing addresses and phone numbers. Once you make an entry for a person in your names file, you no longer have to think about his userid or node.

Another good thing about having a names file is that you can assign a nickname, any name you want, to each person. Then, when you use the commands (discussed later in this chapter) to send messages, notes, or files to someone, you can identify him just by using his nickname.

How to Create a Names File

The NAMES command makes it easy for you to create a names file. It displays a special screen for data entry. You type the required information directly on this screen. All the information you type on one screen is called an "entry" in the names file. The names file is the collection of all the entries you create.

Enter this command on the CMS command line to display the names screen:

```
NAMES
```

The screen is shown below. (When you enter NAMES, your userid appears (automatically) in the first line instead of "userid.")

```

====> userid NAMES <=====> NAMES FILE EDITING <===
Fill in the fields and press a PFkey to display and/or change your NAMES file.
Nickname:      Userid:      Node:      Notebook:
              Name:
              Phone:
              Address:
              :
              :
              :
List of Names:
              :
              :
              :

You can enter optional information below. Describe it by giving it a "tag".
Tag:           Value:
Tag:           Value:
1= Help       2= Add       3= Quit      4= Clear     5= Find      6= Change
7= Previous   8= Next      9=          10= Delete   11=         12= Cursor
====>

```

Macro-read 1 File

You type the data in the space provided next to each labeled field. You can press the key marked \rightarrow to move the cursor quickly to each field.

Creating a Sample Names File

Once upon a time, there was a computer user named Snow White. Her userid was SNOWHITE, and her node was FOREST (probably someplace in southern California).

She used her computer to communicate with Prince Charming (for love messages), the Bad Queen (for hate notes), the Seven Dwarfs (for domestic files).

All these people had userids.

To create a names file, **fill in a screen for each entry**. The first entry you create should be for yourself. (You probably don't want to

fields in capital or lowercase letters or a combination of the two. They remain the same way you type them.

Notebook:

We'll discuss this field later in this chapter.

List of Names:

We'll discuss this field later in this chapter.

Tag: Value:

These fields are optional. You can use them to save additional information about someone. **Tag** is a label you choose. **Value** is its value. For example, in Snoozy's entry (see the next page), you might add:

Tag: Job title Value: Second class miner

After filling in the fields, **press PF2**. This enters the data in the names file. A message appears near the bottom of the screen (on the line above the large arrow line), telling you the entry has been made.

To type in another entry, first **press PF4**. This clears the fields, and you can type the information for the next entry.

When you are finished adding entries, **press PF3** to exit from the screen.

Here are a few more entries in Snow White's names file.



```
====> SNOWHITE NAMES <=====> N A M E S   F I L E   E D I T I N G <===  
Fill in the fields and press a PFkey to display and/or change your NAMES file.  
Nickname: SNOOZY  Userid: SNOOZY  Node: COTTAGE  Notebook:  
      Name: I. M. Dozing  
      Phone: 777-7777  
      Address: Dwarf Cottage  
              : Forest  
              :
```

```
====> SNOWHITE NAMES <=====> N A M E S   F I L E   E D I T I N G <===  
Fill in the fields and press a PFkey to display and/or change your NAMES file.  
Nickname: GROUCHY  Userid: GROUCHY  Node: COTTAGE  Notebook:  
      Name: E. B. Scrooge  
      Phone: 777-7777  
      Address: Dwarf Cottage  
              : Forest  
              :
```

```
====> SNOWHITE NAMES <=====> N A M E S   F I L E   E D I T I N G <===  
Fill in the fields and press a PFkey to display and/or change your NAMES file.  
Nickname: WITCH  Userid: QUEEN  Node: CASTLE  Notebook:  
      Name: Bad Queen  
      Phone: UGLY-1111  
      Address: Vanity Lane  
              : Mirror City  
              :
```

Using the Notebook Field

In this entry, the Notebook field is filled in:

```
====> SNOWHITE NAMES <=====> N A M E S   F I L E   E D I T I N G <===
Fill in the fields and press a PFkey to display and/or change your NAMES file.
Nickname: GORGEOUS  Userid: PRINCE   Node: ATTLARGE  Notebook: PRIVATE
                Name: Prince Charming
                Phone: Area 111 111-1111
                Address:
                :
                :
                :
                List of Names:
                :
                :
                :

You can enter optional information below. Describe it by giving it a "tag".
Tag:                Value:
Tag:                Value:

1= Help      2= Add      3= Quit      4= Clear      5= Find      6= Change
7= Previous  8= Next      9=          10= Delete    11=          12= Cursor

====>

Macro-read 1 File
```

Normally, a copy of all notes that you send and receive are saved automatically. They are saved in a file whose file label is ALL NOTEBOOK. Fill in the Notebook field only if you want a copy of all notes to and from this person to be saved in a separate file. Its filename will be the name you type in the Notebook field. Its filetype will be NOTEBOOK.

With this entry, all notes that Snow White sends to and receives from the prince are saved in a file called PRIVATE NOTEBOOK. Otherwise, all notes to and from the prince would be saved in the file ALL NOTEBOOK, along with everyone else's notes.

What if a person in the list is not in the names file? He can still be in this list if he shares your computer. In this case, you type his userid in the list. Remember to press PF2 to add the entry to the file.

You can communicate either with each person individually, or you can communicate with the group as a whole. It all depends on which nickname you use. When Snow White communicates with one dwarf, she uses that dwarf's nickname. When she communicates with all the dwarfs, she uses the nickname of the list. The note or file will go to everyone on the list.

Of course, you can have more than one list entry, each with its own nickname. For example, you might have a list named MANAGERS or FRIENDS.

Keeping Your Names File up to Date

So far, we've been discussing how to create new entries. But what if someone moves? Or they get a new phone number? Or even a new name?

The PF keys on the names screen are set to functions that make it easy to keep your names file up to date.

How to Change an Entry

To change an entry, first display the screen for that entry. You do this by entering the NAMES command with that person's nickname, like this:

```
NAMES nickname
```

For example,

```
NAMES SNOOZY
```



Type over the data you want to change, and **press PF6**.

Remember to press PF6, not PF2, when changing an entry. If you press PF2, you'll have two entries—the original and the one with the changes you just typed.

How to Delete an Entry

Display the entry that you want to remove from the file by entering:

```
NAMES nickname
```

Then **press PF10**. The entry is deleted from the names file.

How to Find Information

After you enter a NAMES command, you can use the PF7 and PF8 keys to display the entries, one by one.

Press PF8 to display the next entry in the file. Press PF7 to display the previous entry in the file.

By pressing these keys repeatedly, you can “scroll through” the whole file.

Suppose you want to find a specific piece of information. Let's say you forgot someone's nickname and want to see the entry for that person. You need to know only one unique field for that entry, like the person's name. Then follow these steps:

1. First, get a cleared names screen. You can do this by entering the NAMES command. Or, if you've already entered NAMES and have an entry displayed, press PF4 to clear the fields.
2. Fill in the field you can remember.

3. Then **press PF5**. The first entry that contains the field you typed will appear. If your names file does not contain the field you typed, you'll get the message, "Entry not found."

Summary of PF Keys on the Names Screen

PF1 Help

Display a detailed description of the NAMES command.

PF2 Add

Add this entry to the names file.

PF3 Quit

Exit from the NAMES screen.

PF4 Clear

Blank out the fields, so you can create a new entry.

PF5 Find

Fill in one of the fields on a cleared screen, and press the PF5 key. If a matching field is found, that entry is displayed.

PF6 Change

Change an existing entry. For example, if someone's address or phone number changes, type the new data on their screen and press this key.

PF7 Previous

Display the previous entry.

PF8 Next

Display the next entry.



PF9

No function assigned.

PF10 Delete

Remove this entry.

PF11

No function assigned.

PF12 Cursor

Move the cursor to the command line or back to its previous location on the screen.

Creating Your Own Names File

Now is a good time to create your own names file. Start with an entry for yourself. Here is a summary of the steps to follow:

1. Enter NAMES.
2. Fill in the fields on the names screen.
3. Press the PF2 key to add this entry to the names file.
4. To add the next entry, press PF4 and go back to step 2.
5. If you're finished for now, press PF3 to quit.

Once you have a names file, you can use the commands discussed in the rest of this chapter to send messages, notes, and files to these people just by using their nicknames. You can also use the commands to communicate with people who do not have entries in your names file, as long as they share your computer.

Sending and Receiving Messages

What Is a Message?

A message is a short communication, usually a sentence or two. It's the kind of information you usually tell someone over the phone. You can use the TELL command to send a message to another person, as long as that person is logged on.

TELL Command

The TELL command looks like this:

```
TELL name message
```

You can enter a TELL command on the CMS command line, on the FILELIST large arrow command line, or on the XEDIT large arrow command line. The name can be a nickname from your names file. Or, if the person shares your computer, the name can be his userid. After the name, you type the message. For example,

```
TELL SNOOZY Wake up! It's lunch time!
```

After you type your message, press the ENTER key.

Shortly after you send the message, Snoozy sees the following on his screen:

```
MSG FROM SNOWHITE: Wake up! It's lunch time!
```

The first part tells Snoozy who the message is from. It's always the sender's **userid**, not his nickname. Now Snoozy knows who sent the message and can send one back, if he wants to.



If Snoozy were not logged on, Snow White would see this message:

```
SNOOZY not logged on
```

The message wouldn't be saved and given to Snoozy when he does log on. You have to send it again when he does.

What happens if someone sends you a message while you are editing a file? You wouldn't get the message until you press the ENTER key. The screen clears, and the message appears. The notice HOLDING appears at the bottom right corner of the screen. After you read the message, press the CLEAR key, and your file returns to the screen.

I Don't Want to Receive Messages

If you don't want to be interrupted by messages while you're working at your terminal, enter this command:

```
SET MSG OFF
```

This command can be entered on the CMS command line, the FILELIST large arrow command line, or the XEDIT large arrow command line. Now, you won't receive messages either from other users or from the computer.

If you decide later that you do want to receive messages, enter this command:

```
SET MSG ON
```

The SET MSG OFF command is good for one terminal session only. When you log off, it's cancelled automatically.

Note: If you try to send a message to someone who has used a SET MSG OFF command, you will receive a message telling you that person is not receiving messages.

For Practice

If you'd like to practice sending messages, send some to yourself.
Type

```
TELL name anything you want
```

For the name, substitute your userid. If you have a names file with an entry for yourself, use your nickname.

Sending Notes

What Is a Note?

A note is the kind of communication you usually do by letter. It is generally longer than a message. A more important difference between messages and notes is that a person does **not have to be logged on** to receive a note, as he does with messages. A note is stored by the system until the person is ready to read it.



NOTE Command

You can use the NOTE command to prepare and send a note. A note can be sent to one person or to a list of people. You can even specify that certain people receive a “complimentary copy” of a note. Each recipient can then place a copy of the note on his disk. All this can happen with no copies printed on paper unless you want them to be.

The format of the NOTE command is:

```
NOTE name
```

where “name” can be one or more nicknames in the names file, or it can be one or more userids of people who share your computer.

For example,

```
NOTE SNOOZY
```

Or,

```
NOTE SNOOZY SMILEY BOSS
```

You **must specify a name** when you begin a new note.

When you enter the NOTE command, a special screen is displayed. The following note screen was created by entering the command:

```
NOTE SNOOZY
```

```

SNOWHITE NOTE      AO  V 132  Trunc=132 Size=9 Line=9 Col=1 Alt=0
* * * Top of File * * *
OPTIONS: NOACK     LOG      SHORT   NOTEBOOK ALL

Date: 11 February 1986, 11:04:52 EST
From: Snow White   ZZZ-ZZZZ           SNOWHITE at FOREST
To:  SNOOZY at COTTAGE

* * * End of File * * *

1= Help      2= Add line  3= Quit      4= Tab      5= Send     6= ?
7= Backward  8= Forward   9= =        10= Rgtleft 11= Spltjoin 12= Power input

====> _

X E D I T I File

```

The note screen has headings that show the date and time, and information about the sender (you) and the recipient(s). You do not have to type these headings. The information is collected from your names file and is placed in the note automatically. (You can ignore the line that begins with "OPTIONS.")

If you want the headings to contain more information than is shown in the headings above, you can enter this form of the NOTE command:

```
NOTE  nickname  (LONG
```

You must type the left parenthesis before the word LONG, exactly as shown above.

These headings were produced by entering the following command:

```
NOTE  SNOOZY  (LONG
```



```
Date: 11 February 1986, 11:04:52 EST
From: Snow White          ZZZ-ZZZZ          SNOWHITE at FOREST
      201 Redwood Lane
      Forest Primeval, California
To:   I. M. Dozing        777-7777          SNOOZY   at COTTAGE
```

The names and addresses are collected from your names file. If you use the LONG option and someone's name or phone number is not in his entry, it won't appear in the note heading.

Typing the Body of the Note

After entering NOTE, you are in edit mode, as you are when you XEDIT a file. However, the screen looks different from the usual XEDIT screen layout. There is no prefix area or scale.

Before you start typing the note, enter the INPUT command on the command line (next to the large arrow at the bottom of the screen). Then you can type the body of the note, in the space provided after the headings.

```
SNOWHITE NOTE      A0  V 132  Trunc=132 Size=9 Line=9 Col=1 Alt=0
* * * Top of File * * *
OPTIONS: NOACK     LOG      SHORT   NOTEBOOK ALL

Date: 11 February 1986, 11:04:52 EST
From: Snow White          ZZZ-ZZZZ          SNOWHITE at FOREST
To:   SNOOZY at COTTAGE

Dear Sir:
  Would you be interested in hiring domestic help?
  I understand the cottage is a mess, what with your having
  to work in the mines and sing "Heigh-Ho" and all that.
  In addition to being a hard worker, my lips are red as blood
  and I have black hair.

                                Thank you,
                                S. White
                                4= Tab          5= Send          6= ?
1= Help          2= Add line  3= Quit
7= Backward     8= Forward  9= =
Input mode:
====> * * * Input Zone * * *

                                10= Rgtleft  11= Spltjoin 12= Power input

                                Input-mode 1 File
```

All the PF keys but PF5 are set to commands that help you edit your note. If they look familiar, it's because they are XEDIT commands. For example, if your note doesn't fit on one screen, you press PF8 (FORWARD) to continue typing on a new screen.

If you want to use XEDIT commands that aren't assigned to PF keys, you must go back to edit mode. Press the ENTER key twice, just as you do in a regular file. Then you can use both the PF keys and enter XEDIT commands from the command line while composing your note. If you don't need to use any XEDIT commands, you can stay in input mode until you finish typing.

If you want to use insert mode, just press the insert mode key, insert the characters or words, and press the RESET key to resume normal typing.

Sending the Note

When you are finished typing your note, **press PF5** to send it.

Sending Notes to People on a List

The nickname you specify in the NOTE command can be the nickname assigned to a list in your names file. In this case, a copy of the note is sent to each person in the list.

For example, Snow White's names file contains an entry for a list whose nickname is DWARFS. The List of Names field has the nicknames of the seven dwarfs. (You can find this entry under "Creating an Entry for a List of Names" on page 145.) The command

```
NOTE DWARFS
```

sends a copy of the note to each of the seven dwarfs.



The note screen would look like this:

```
SNOWHITE NOTE      AO V 132  Trunc=132 Size=9 Line=9 Col=1 Alt=0
*** Top of File ***
OPTIONS: NOACK     LOG      SHORT   NOTEBOOK ALL

Date: 11 February 1986, 11:04:52 EST
From: Snow White           ZZZ-ZZZZ           SNOWHITE at FOREST
To:  SNOOZY at COTTAGE, DUMMY at COTTAGE, BOSS at COTTAGE, SMILEY at COTTAGE
     GROUCHY at COTTAGE, SNIFFLES at COTTAGE, WISTFUL at COTTAGE

Dear Sir,
  Would you be interested in hiring domestic help?
  I understand the cottage is a mess, what with your having
  to work in the mines and sing "Heigh-Ho" and all that.
  In addition to being a hard worker, my lips are red as blood
  and I have black hair.

                                Thank you,
                                S. White
                                4= Tab           5= Send           6= ?
1= Help           2= Add line  3= Quit           10= Rgtleft  11= Spltjoin 12= Power input
7= Backward      8= Forward  9= =

====>

                                X E D I T 1 File
```

If you want the long headings, you would enter:

NOTE DWARFS (LONG

Then the headings would look like this:

```
From: Snow White           ZZZ-ZZZZ           SNOWHITE at FOREST
      201 Redwood Lane
      Forest Primeval, California
To:   I. M. Dozing         777-7777
      S. A. What           777-7777
      T.C.P. Banana       777-7777
      H. A. Haas          777-7777
      E. B. Scrooge       777-7777
      A. H. Choo          777-7777
      R. U. Shy           777-7777
      SNOOZY at COTTAGE
      DUMMY at COTTAGE
      BOSS at COTTAGE
      SMILEY at COTTAGE
      GROUCHY at COTTAGE
      SNIFFLES at COTTAGE
      WISTFUL at COTTAGE
```

Sending Complimentary Copies

Sometimes you may want someone to see a copy of a note, but you want him to know that he's not the main recipient. If you were using paper, you'd send him a carbon copy. With the computer, you can send him a "complimentary copy." To send a complimentary copy, you could enter the NOTE command like this:

```
NOTE BOSS CC: SNIFFLES SMILEY
```

"CC" stands for complimentary copy. The name following "CC:" can be one or more names of people who have entries in the names file, or one or more userids of people who share your computer.

In this example, BOSS is the main recipient. SNIFFLES and SMILEY are nicknames of people who have entries in the names file. The note header will show that they are complimentary copy recipients.

```
Date: 11 February 1986, 11:04:52 EST
From: Snow White          ZZZ-ZZZZ          SNOWHITE at FOREST
To:   BOSS at COTTAGE
cc:   SNIFFLES at COTTAGE, SMILEY at COTTAGE
```

The name following "CC:" can also be the nickname of a list.

```
NOTE WITCH CC: DWARFS
```

DWARFS is the nickname of an entry that contains the names of the people in the list. (You can find this entry under "Creating an Entry for a List of Names" on page 145.) Each person in the list will be a complimentary copy recipient.



Keeping Copies of Notes

A copy of every note you send is saved automatically in a file called ALL NOTEBOOK. Each time a note is sent, a copy is added to the end of this file. A line of equal signs separates each note. If you ever need to look up a note, you can use the editor to examine this file. You would enter XEDIT ALL NOTEBOOK.

Sometimes you may want to keep all the notes you send to one person in a separate file. To do this, you can specify a filename in the Notebook field of that person's names file entry. A copy of each note you send to that person is saved automatically in a file whose filename is the name you specified and whose filetype is NOTEBOOK. In addition, all notes that you **receive** from that person are also kept in this file. You'll see how to receive notes in "Receiving Notes and Files" on page 171.

If you ever need to look up a note that you sent, you can use the editor to examine that file. For example, suppose you typed "snoozy" in the Notebook field of Snoozy's names file entry. You could enter XEDIT SNOOZY NOTEBOOK to look at the notes you sent to Snoozy.

Continuing Notes

Suppose you're typing a note and have to stop before you're finished. You can save the note and finish it later by following these steps:

- Enter FILE from the XEDIT command line at the bottom of the note screen. Your note is saved under the file label "userid NOTE," where "userid" is your userid.

- Later, when you want to finish the note, enter the NOTE command, but **do not specify a name**. Your unfinished note appears on the screen. You can then complete the note and send it.

Summary of PF Keys on the NOTE Screen

PF1 Help

Display a detailed description of the NOTE command.

PF2 Add line

Add a blank line after the line containing the cursor.

PF3 Quit

End the note session without saving any changes.

PF4 Tab

Use this key to tab the cursor.

PF5 Send

Send the note.

PF6 ?

Display the last command that was executed.

PF7 Backward

Scroll back one screen.

PF8 Forward

Scroll forward one screen.

PF9 =

Execute again the last command that was executed.



PF10 Rgtright

View data to the right of the screen. Press PF10 again to return to original view.

PF11 Spltjoin

Split a line or join two lines at the cursor.

PF12 Power input

Enter power typing mode.

For more information on the various XEDIT commands, see Chapter 2.

Summary of How to Send a Note

Follow these steps to prepare and send a note.

- Enter the command NOTE in one of its forms:

NOTE name

NOTE name CC: name

NOTE name (LONG

NOTE name CC: name (LONG

In each form, "name" can be one or more names. A name can be a nickname of an entry in the names file, or it can be the userid of someone who uses the same computer system that you use, or it can be a combination of nicknames and userids.

- Type the body of the note.
- Press PF5 to send the note.

Sending Files

Suppose you have a file that you want other people to see. This section shows you how to send people a copy of a file for their own personal use. Each recipient can then place the copy on his disk, so it will be available when he needs it.

SENDFILE Command

When you enter the SENDFILE command, a special screen for data entry is displayed, like the one shown below. You type the name of the file to be sent and the name of the recipient directly on this screen.

An abbreviation for SENDFILE is SF.

```
----- SENDFILE -----
File(s) to be sent      (use * for Filename, Filetype and/or Filemode
                        to select from a list of files)
Enter filename : _
      filetype : _
      filemode :
Send files to :
Type over 1 for YES or 0 for NO to change the options:
  0   Request acknowledgement when the file has been received?
  1   Make a log entry when the file has been sent?
  1   Display the file name when the file has been sent?
  0   This file is actually a list of files to be sent?
1= Help          3= Quit          5= Send          12= Cursor
====>
Macro-read 1 File
```



Sending One File

To send one file, follow these steps:

- Type the filename in the space provided.
- Type the filetype in the space provided.
- Type the filemode if it is not "A." You can leave this field blank if the file is on your A-disk.
- Type the name(s) of the recipient(s). You can use nicknames from your names file. The nickname can represent a single user. For example,

```
Send files to: snoozy
```

Or you can type more than one name on this line. (Do not separate the names with commas.) For example,

```
Send files to: snoozy grouchy prince
```

The nickname can also represent a list. In this case, the file is sent to each person in the list. For example,

```
Send files to: dwarfs
```

If the recipient uses the same computer system that you use, you can type his userid.

- Choose the "0/1" options.

You can select certain options from the list on the screen. They are described below.

- Press PF5 to send the file and exit from the sendfile screen. Pressing the ENTER key sends the file but keeps the screen (so that you can use it to send another file).

Choosing the Options

Each option is in the form of a question, with either 0 or 1 displayed to its left. A 0 represents NO, while a 1 represents YES. If you're satisfied with the answer shown, don't do anything. If you want a different answer, type 1 over the 0 (or 0 over the 1).

The options are as follows:

0 Request acknowledgement when the file has been received?

Type 1 only if you want to get an acknowledgment when the person receives (or discards) your file. The acknowledgment shows the date and time you sent the file, and the recipient's userid and node.

When you get an acknowledgement, it appears in your reader and a message appears on your screen. The message informs you that you have been sent a file. (Your reader is a part of a disk where incoming files are kept. You'll learn how to handle these files in "Receiving Notes and Files" on page 171.) If you choose to receive the acknowledgment, an entry is made in a "userid NETLOG" file, which is explained below.

1 Make a log entry when the file has been sent?

Each time you send a file, an entry is automatically made in a special log file. Its file label is "userid NETLOG," where the userid is your userid.

A typical entry might look like this:

```
File MY DATA A1 sent to SNOOZY at COTTAGE on 10/10/86 11:30:25
```



If you specified 1 (YES) on the first option (acknowledgment), an entry is also made when you receive the acknowledgment.

Type 0 if you don't want an entry made in the log file.

1 Display the file name when the file has been sent?

The names of the file(s) and the userid(s) and node(s) of the recipients are displayed, on a cleared screen. Type 0 for NO if you don't want this information displayed.

0 This file is actually a list of files to be sent?

This option provides one way to send many files at once. In order to use it, you need to learn how to use the FILELIST command, which is explained in Chapter 3. Type 1 for YES if you have saved a list of files produced by the FILELIST command. This option is discussed more fully below.

Sending More than One File

What if you want to send a lot of files to someone? One way to do this is by filling in sendfile screens, one for each file you want to send. But there are easier ways. One way is to save a list of the files you want to send using the FILELIST command. Another way is to select the files you want to send from a list that contains a subset of your files. Both ways allow you to send all the files at once.

Using FILELIST to Save a List of Files to be Sent

You can use the FILELIST command to create a list of the files you want to send. Then you FILE that list. (If you are not familiar with the FILELIST command, read Chapter 3.)

For example, if you want to send all files on your A-disk whose filetype is **payroll**, you would enter:

```
filelist * payroll
```

The list might look like this:

```
SNOWHITE FILELIST A0 V 108 Trunc=108 Size=7 Line=1 Col=1 Alt=0
Cmd Filename Filetype Fm Format Lrecl Records Blocks Date Time
WISTFUL PAYROLL A1 V 95 34 2 10/04/86 21:12:04
BOSS PAYROLL A1 V 95 29 2 10/04/86 20:58:07
DUMMY PAYROLL A1 V 107 281 10 10/04/86 17:59:00
WITCH PAYROLL A2 V 98 49 2 10/03/86 15:17:01
GROUCHY PAYROLL A1 V 92 101 4 10/02/86 15:33:05
PRINCE PAYROLL A2 V 75 28 1 9/25/86 12:10:03
SNOOZY PAYROLL A2 V 120 277 10 9/24/86 9:14:02

1= Help      2= Refresh  3= Quit     4= Sort(type)  5= Sort(date)  6= Sort(size)
7= Backward  8= Forward  9= FL /n   10=           11= XEDIT     12= Cursor

----> file my list

X E D I T 1 File
```

Enter a FILE command on the large arrow command line, changing the file label. Remember that the list is a file, whose file label is "userid FILELIST." Change either the filename or filetype (or both) in your FILE command. In the screen above, the FILE command changes the label to MY LIST when the ENTER key is pressed. The



list would now be on your A-disk. Its file label would be MY LIST A.
(If you don't specify a filetype, it will be FILELIST.)

Enter the SENDFILE command.

```
----- SENDFILE -----
File(s) to be sent      (use * for Filename, Filetype and/or Filemode
                        to select from a list of files)
Enter filename : my
                filetype : list
                filemode :

Send files to :

Type over 1 for YES or 0 for NO to change the options:
  0   Request acknowledgement when the file has been received?
  1   Make a log entry when the file has been sent?
  1   Display the file name when the file has been sent?
  1_  This file is actually a list of files to be sent?
  1= Help          3= Quit          5= Send          12= Cursor
====>

Macro-read 1 File
```

Type "1"
Over the
"0"

On the sendfile screen, you

- Type the filename and filetype of the file list you saved.
- Type the name of the recipient(s).
- Type "1" for the last option.
- Then press PF5 to send the files.

Selecting Files from a List

Another way to send many files at once is to select the files from a list of some or all of your files. You would probably choose this method if you wanted to send only some, but not all, files with the same filename or filetype. Or perhaps you'd like to browse through a list before deciding which files to send.

First enter the SENDFILE command to display a sendfile screen.

```
----- SENDFILE -----
File(s) to be sent      (use * for Filename, Filetype and/or Filemode
                        to select from a list of files)
Enter filename : *
filetype : data
filemode :
Send files to : sleepy_
Type over 1 for YES or 0 for NO to change the options:
  0      Request acknowledgement when the file has been received?
  1      Make a log entry when the file has been sent?
  1      Display the file name when the file has been sent?
  0      This file is actually a list of files to be sent?
1= Help          3= Quit          5= Send          12= Cursor
====>
Macro-read 1 File
```

On the sendfile screen, you type an asterisk (*) next to the filename, filetype, and/or filemode fields, as shown above. An asterisk in any of these fields means that you want the list to contain **all** files with a certain filename, filetype, and/or filemode. Of course, if you wanted the list to contain **all** files on your A-disk, you'd type an asterisk for filename **and** filetype.

In the screen above, an asterisk is typed in the filename field and "data" is typed in the filetype field. This means that the list will contain all files on Snow White's A-disk whose filetype is DATA. To display the list, first finish filling out the sendfile menu, and then press PF5.



```
SNOWHITE FILELIST A0 V 108 Trunc=108 Size=9 Line=1 Col=1 Alt=0
Cmd Filename Filetype Fm Format Lrecl Records Blocks Date Time
s WISTFUL DATA A1 V 95 34 2 10/04/86 21:12:04
BOSS DATA A1 V 95 29 2 10/04/86 20:58:07
s DUMMY DATA A1 V 107 281 10 10/04/86 17:59:00
MISC DATA A2 V 98 49 2 10/03/86 15:17:01
s GROUCHY DATA A1 V 92 101 4 10/02/86 15:33:05
PRINCE DATA A2 V 75 28 1 9/25/86 12:10:03
SNOOZY DATA A2 V 120 277 10 9/24/86 9:14:02
SNIFFLES DATA A1 V 26 7 1 9/23/86 8:50:06
WITCH DATA A1 V 80 489 30 8/26/86 8:05:08
```

```
1= Help      2= Refresh   3= Quit      4= Sort(type) 5= Sendfile   6= Sort(size)
7= Backward  8= Forward   9= FL /n    10=           11= XEDIT    12= Cursor
Type 'S' in front of each file to be sent and press ENTER
====>
```

```
X E D I T 1 File
```

To mark the files you want sent, type the letter "s" in front of the filename of each file you want sent. Then press the ENTER key to send the files. In this example, three files would be sent to Sleepy (the recipient specified on the sendfile screen). They are the files marked with an "s." When the files are sent, an asterisk (*) appears in the "Cmd" space next to each file sent.

If your file list extends for more than one screen and you also want to send some of these files, first scroll the screen (using PF8). Then type an "s" next to the files you want to send and press ENTER.

When you are finished sending files, press PF3 to exit from the file list (and sendfile) screens.

PF Keys on the File List Screen

The PF keys are set to commands that help you find information or rearrange the files in the list. Except for PF5, they are the same as those on the regular FILELIST screen. (You can use PF5 to send a file by placing the cursor on that line and then pressing PF5.) For more information on these PF keys, see the FILELIST command, in Chapter 3.

Summary

Here are the steps to follow when sending many files by selecting them from a list:

- Type SF and press the ENTER key to display the sendfile screen.
- Type * for filename, filetype, and/or filemode to define the subset of all your files you want displayed in the list.
- Type the name of the recipient(s).
- Type over the 0|1 options, if you want something other than what is shown on the screen.
- Press PF5 to display the list.
- Type "S" in front of each filename you want sent.
- Press the ENTER key to send the file(s).
- Press PF3 to quit.



For Practice

For practice, you can send yourself a file. After entering SF, fill in the sendfile screen. Type in the filename and filetype of a file that you already have on your disk. Type in your nickname (if you have an entry for yourself in a names file), or your userid. Then press PF5.

The next section shows you how to receive files and notes that are sent to you. If you send yourself a file, you'll have a "real" example to follow.

Receiving Notes and Files

All notes and files that other people send you are delivered to your **reader**. Your reader is a temporary place on a disk where incoming notes and files are kept until you decide what you want to do with them. In this section, both notes and files are referred to as files.

After you log on, a message may appear on your screen, notifying you that a file is in your reader. It might be similar to the following message:

```
FILES: 003 RDR, NO PRT, NO PUN
```

The part of the message that reads FILES: 003 RDR means that one or more people sent you a total of three files, and they are currently in your reader.

You might also receive a message that files are in your reader while you're editing a file. The message appears on a cleared screen. When you press the CLEAR key, your file returns.

You can use the RDRLIST command to display information about the files in your reader. Then you can decide if you want to keep the files or discard them.

RDRLIST Command

The RDRLIST command displays information about files in your reader. An abbreviation for RDRLIST is RL.

If you sent yourself a file, as suggested in the previous section, you can enter RL now. A sample rdrlist screen is shown below:

```
SNOWHITE RDRLIST      AO  V 108  Trunc=108 Size=3 Line=1 Col=1 Alt=0
Cmd  Filename Filetype Class User   at Node Hold  Records  Date   Time
-   MINING   NEWS      PUN A SNOOZY COTTAGE NONE    10  07/07 10:41:38
    SNOOZY   NOTE      PUN A SNOOZY COTTAGE NONE    10  07/07 10:39:11
    MIRROR   MIRROR    PUN A QUEEN  CASTLE  NONE     7   07/07 10:20:50

1= Help      2= Refresh   3= Quit      4= Sort(type) 5= Sort(date) 6= Sort(user)
7= Backward  8= Forward   9= Receive  10=           11= Peek      12= Cursor

====>

                                X E D I T  1 File
```

As you can see, the rdrlist screen lists a lot of information about the files in Snow White's reader:

Cmd is an area where you can enter commands. We'll discuss this later. **Filename** is the filename; **Filetype** is its filetype. **Class** isn't important to you right now. **User** tells you the userid of the person who sent you the file. **At Node** is his node. **Hold** isn't important to



you right now. **Records** tells you how many records are in the file. **Date** is the date the file was sent. **Time** is the time it was sent.

What Should I Do with the Files in My Reader?

You can do any of the following things with each of the files in your reader:

- Receive the file, that is, move it from the reader to your disk.
- Discard the file.
- Peek at the file, that is, look at its contents without moving it to your disk. Then decide if you want to receive it or discard it.

Receiving a File

To receive a file, move the *cursor* to the line that describes the file you want to receive. Then *press PF9*. A notice will appear on that line, telling you that the file has been received, as follows:

```
SNOWHITE RDRLIST      A0  V 108  Trunc=108 Size=3 Line=1 Col=1 Alt=3
Cmd  Filename Filetype Class User      at Node Hold  Records  Date  Time
*   MINING   NEWS      received from SNOOZY at COTTAGE.
SNOOZY NOTE      PUN A SNOOZY COTTAGE NONE      10  07/07 10:39:11
MIRROR MIRROR     PUN A QUEEN  CASTLE  NONE      7   07/07 10:20:50
```

Notice that an asterisk appears in the "Cmd" space next to the received file. It's just a signal to remind you that you did something with this file.

If more than one file is in your reader, move the cursor to the appropriate line and press PF9 for each file you want to receive.

Replacing or Renaming a File

What happens if the file in your reader has the same file label as one you already have on your disk? In this case, when you press PF9, the following message appears at the bottom of your rdrlst screen:

```
File fn ft fm already exists; specify REPLACE option
```

This means that the computer won't move the file to your disk, because two files can't have the same file label.

There are two ways to handle this.

If you want the file in your reader to replace the file on your disk, type the following command in the "Cmd" space next to the file you want to receive. Then press the ENTER key.

```
RECEIVE / (REPLACE
```

You would probably want to replace the file on your disk if, for example, someone sent you an updated copy of the file.

However, if you want to keep the file on your disk, you can give the file in your reader a different file label. Type the following command in the "Cmd" space. Then press the ENTER key.

```
RECEIVE / newfilename newfiletype
```

You need to specify either a new filename *or* a new filetype for the file label to be unique. For example, if a file label is MINING NEWS, you might specify either RECEIVE / MINING1 NEWS or RECEIVE / MINING TIDBITS.



Receiving Notes

You can recognize a note because its filetype is NOTE. The second file in Snow White's reader is a note. You receive a note file the same way you do other kinds of files. Move the cursor to the line that describes the note and **press PF9**.

```
SNOWHITE RDRLIST      A0  V 108  Trunc=108 Size=3 Line=1 Col=1 Alt=6
Cmd  Filename Filetype Class User   at Node Hold  Records  Date  Time
*   MINING   NEWS      received from SNOOZY at COTTAGE.
*   SNOOZY   NOTE      added to ALL NOTEBOOK A0.
MIRROR MIRROR    PUN A QUEEN CASTLE NONE          7 07/07 10:20:50
```

Note files are added to the file ALL NOTEBOOK, as the notice indicates. Of course, if you specified a different NOTEBOOK filename in your names file entry for that person, the note is added to that file.

Discarding a File

To discard a file, type DISCARD in the "Cmd" space next to the filename. Then press ENTER. The notice "has been discarded" appears in that line.

Sometimes you can't decide whether you want to receive or discard a file unless you "peek" at its contents. Let's peek at the last file in Snow White's reader and then decide if we want to keep it or discard it.

Peeking at a File

By placing the **cursor** on the appropriate line and **pressing PF11**, you can "peek" at the file. This means that you can see what's in the file without first moving it to your disk. Let's peek at the last file in Snow White's reader. It is displayed on a peek screen, as shown below:

```
3001      PEEK      A0 V 255 Trunc=255 Size=7 Line=0 Col=1 Alt=0
FILE MIRROR MIRROR from QUEEN at CASTLE. Format is DISK-DUMP.
* * * Top of File * * *
      Apples, Apples, Apples

I have recently developed a new variety of apple. It is
crunchier, sweeter, and larger than the run-of-the-mill varieties.
I am leaving samples on doorsteps throughout the forest, free
of charge. I guarantee you'll like these apples. After just one
bite, you'll feel tranquil, totally at peace.
* * * End of File * * *
```

1= Help 2= Add line 3= Quit 4= Tab 5= Clocate 6= ?/Change
7= Backward 8= Forward 9= Receive 10= Rgtleft 11= Spltjoin 12= Cursor

====>

X E D I T 1 File

PF Keys on the Peek Screen

You can locate information by typing the XEDIT command CLOCATE/data/ in the XEDIT command line (at the bottom of the screen) and pressing PF5. Each time you press PF5, the next occurrence of the string is located. This means that the cursor moves under the string, and the line that contains it is highlighted.

If the file takes up more than one screen, you can use PF7 and PF8 to scroll through the file.



If you decide you want to receive the file, press PF9. You will then go back to the rdrlist screen automatically.

If you decide to discard the file, type DISCARD in the large arrow command line on the peek screen, and press ENTER. Snow White decides that no news from the Bad Queen can be good news, so she discards the file.

Now her rdrlist screen looks like this:

```
SNOWHITE RDRLIST      AC V 108 Trunc=108 Size=3 Line=1 Col=1 Alt=9
Cmd  Filename Filetype Class User   at Node Hold  Records  Date  Time
*    MINING   NEWS   received from SNCOZY at COTTAGE.
*    SNOOZY   NOTE   added to ALL NOTEBOOK AC.
*    MIRROR   MIRROR has been discarded.
```

Using PF Keys to Scroll through or Rearrange the List

Snow White's reader contains only three files. However, your reader can contain many files, and the rdrlist display can occupy more than one screen.

By using PF keys, you can scroll through the rdrlist screens and rearrange the list of files. PF7 and PF8 scroll the screen backward and forward, respectively. PF4, PF5, and PF6 sort the list according to the filetype, date, and the userid from whom the file was sent.

When you're finished using the rdrlist screen, press PF3 to quit.

Is There a File in Your Reader?

If you sent yourself a file, as suggested in the last section (“Sending Files”), enter the command RL (if you haven’t already). If you sent yourself a file that you already have on your disk, you can discard the one in your reader. Type DISCARD in the “Cmd” area on the rdrlst screen, and press ENTER. Then press PF3 to quit.

Summary

You learned the following commands in this chapter:

Command	Function
NAMES	Displays names file data entry screen.
TELL	Sends a message to another person who is logged on.
SET MSG OFF	Prevents messages from being displayed.
SET MSG ON	Resumes message display.
NOTE	Prepares a note for another user, who may or may not be logged on.
SENDFile or SF	Displays data entry screen for sending a file.
RDRLst or RL	Displays information about files in your reader.
RECEIVE	Receives a file that is in your reader, that is, moves it from your reader to your A-disk.
DISCARD	Discards a file that is in your reader.

Chapter 6. Asking Questions



Do you have a question? Well, VM may be able to answer it. You can use commands to ask various questions.

This section teaches you how to ask VM:

- Who am I, where am I, and what's the time and date?
- Who's using the computer?
- How much disk space have I used?

- Which command should I use?

For more information on the commands discussed in this chapter, see the *VM/SP CP Command Reference* and the *VM/SP CMS Command Reference*.

Where Do I Enter These Commands?

The commands discussed in this chapter can be entered on:

- The CMS command line
- The FILELIST large arrow command line
- The RDRLIST large arrow command line
- The XEDIT large arrow command line.

When you enter a command from the FILELIST, RDRLIST, or XEDIT command line, the answer to your question will be displayed on a cleared screen, with MORE... displayed in the status area. You can press the CLEAR key to get the file list, reader list, or file back on the screen.



Who Am I, Where Am I, What's the Date and Time?

When you're using VM, you'll never have an identity crisis, and you won't need a calendar or a watch! To find out your userid and node, and the date, time, and day of the week, use this command:

```
IDentify
```

You'll get a message like this one:

```
SNOWHITE AT FOREST VIA NET1 02/14/86 12:01:00 EST FRIDAY
```

Who's Using the Computer?

You can use three different QUERY commands to find out who else is using your computer.

If you want to know how many people are using your computer, which can affect the response time (how long it takes VM to do the various things you ask it to do), enter

```
====> Query Users
```

The number of people currently logged on is displayed.

If you want to know which of your fellow users are currently logged on, enter

```
====> Query Names
```

You'll see a list of the userids of all the people who are currently logged on.

If you want to know if a particular person is logged on and you don't want to see the whole list of users, enter the QUERY command with that person's userid. For example,

```
====> Q DOE
```

VM responds with a message. If DOE is logged on, VM gives the address of the terminal he is using. If he's not logged on, you see this message: DOE not logged on.

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. How can you find out how much disk space is left? With this command:

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

The response is a message similar to this:

```
LABEL  CUU M  STAT CYL  TYPE  BLKSIZE  FILES  BLKS  USED-(%)  BLKS  LEFT  BLK  TOTAL
CS1191 191 A  R/W   5   3330   800     53   576-43     754   1330
```

The disk label is CS1191. The "address" of the disk (where the disk is) is 191. 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 43%



of this disk is used. Therefore, 57% of this A-disk is still available. The rest of the fields aren't important to you now.

Which Command Should I Use?

Have you ever forgotten the name of a command? If you know what you want to do, but can't remember how to do it, use the HELP command. You can enter the HELP command from any large arrow command line.

To use the HELP command, type:

```
====> help
```

Then press the ENTER key.

When you type HELP from the XEDIT, RDRLIST, or FILELIST command line, you will see a "menu." The menu lists all the XEDIT commands. (Help in XEDIT was described in Chapter 2.)

When you type HELP from the CMS command line, your screen should look like this:

```
HELP TASKS          Task Help Information          line 1 of 18
To use VM Help, move the cursor to any topic below,
then press the ENTER key or the PF1 key.

TASKS      - Helps if you don't know VM commands.
            Good choice for beginners.
MESSAGE    - Explains how to get help for VM messages.
HELP       - Explains some ways for using HELP.
MENUS      - Lists the HELP component MENUS.
COMMANDS   - Lists VM commands that you can use.
CMS        - Shows only CMS commands.
CP         - Shows only CP commands.
XEDIT     - Lists System Product Editor items.
REXX      - Helps you use the REXX language.
DEBUG     - Helps you debug programs.
SQLDS     - Shows SQL/Data System items.
SRPI      - Lists the SRPI subcommands.
PF1= Help  2= Top    3= Quit   4= Return  5= Clocate  6= ?
PF7= Backward 8= Forward 9= PFkeys 10=      11=      12= Cursor

====> _

Macro-read 1 File
```

This is a "menu" of the different types of help you can get.



If you move the cursor under TASKS and press the ENTER key, you will see:

```
TASK TASK          Task Help Information          line 1 of 18
Each VM task below leads to a more detailed
list of tasks for you to choose from.

Place the cursor under the task of your
choice and press the ENTER key or the PF1 key.

Create or change (edit) files
Manage files (copy, rename, erase, etc.)
Work with windows in fullscreen CMS
Communicate with other users
Develop and debug programs and EXECs
Ask questions about the system
Customize your system
Print files
Use disks
Use tapes
PF1= Help          2= Top          3= Quit          4= Return          5= Clocate          6= ?
PF7= Backward      8= Forward      9= PFkeys       10=                11=                12= Cursor

====> -

Macro-read 2 Files
```

Now move the cursor so it is under the task you want to do. Press the ENTER key. The next screen you see will provide more information about the task. Just follow the directions on the screen.

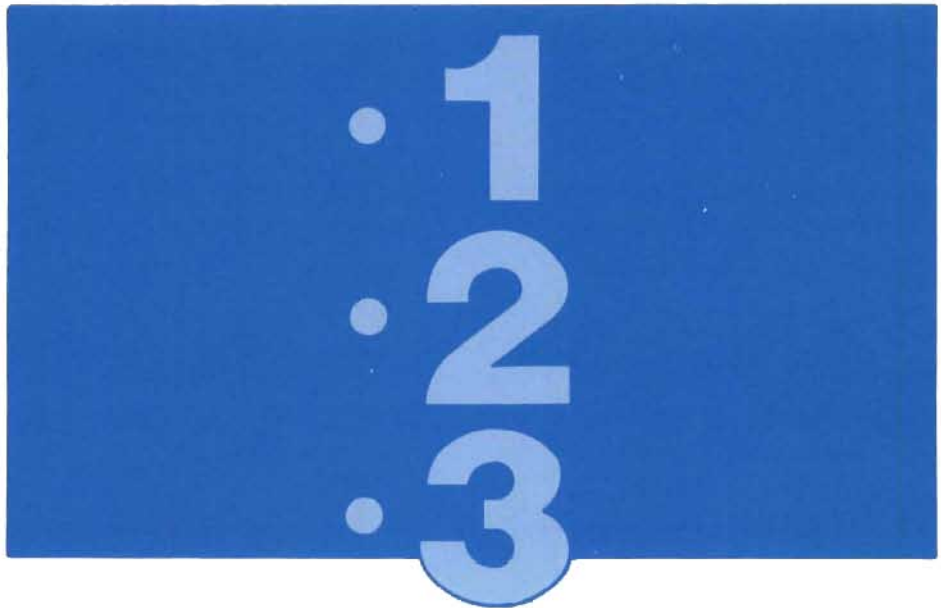
To leave the HELP screens at any time, press the PF4 key. To find out more about using the Help facility, enter the command HELP SELF.

Summary

You learned the following commands in this chapter:

Command	Function
HELP	Displays information about commands.
IDentify	Displays userid, node, time, and date.
Query Users	Displays how many people are logged on.
Query Names	Displays userids of people who are logged on.
Query userid	Tells you if a particular userid is logged on.
Query DISK A	Displays how much disk space you have used.

Chapter 7. Using Script to Format Your Files



If you want to print your files, you can use `SCRIPT/VS` to make them look attractive. `SCRIPT/VS` is part of the Document Composition Facility (DCF), an IBM program product. It is often referred to simply as Script.

You can use Script to format a wide range of printed materials, from letters to full-length documents. You do this by inserting formatting directions, called "markup," in your files. The markup taught in this section consists of shorthand-like notations called GML (Generalized Markup Language) tags.

As you create your file, you insert GML tags that identify the various parts of the document (for instance, a numbered list), and that's it! Script will take care of the formatting, like numbering the list, leaving spaces between list items, and so forth. Script will even check your spelling.

The Document Composition Facility comes with a "starter set" of GML tags. This chapter teaches you some of these GML tags and some of the capabilities of Script. You should do this section at your terminal.

More information on Script can be found in the *Document Composition Facility: Generalized Markup Language Starter Set User's Guide*.

Learning Markup

What Kind of Files Can I Use?

You can use Script to format any file whose filetype is SCRIPT. The file can contain any information you want. You create and change a Script file by using XEDIT commands, just as you have done with all the other files you practiced in this book.

There is one difference between Script files and other filetypes. The editor (XEDIT) doesn't change the contents of Script files to capital letters. The case is automatically "mixed." If you typed capital letters, you get capital letters. If not, the editor won't change the letters to capitals. Therefore, when using a LOCATE or CHANGE command in a Script file, be sure to type the words you want located



(or changed) exactly as they appear in the file, INCLUDING CAPITAL LETTERS.

For example, if you wanted to locate the words “New Year”, your LOCATE command would look like this:

```
====> locate/New Year/
```

If you typed locate/NEW YEAR/, the editor wouldn’t be able to find the words.

Would you like XEDIT to ignore the case of letters when locating words? You can use this command:

```
====> SET CASE MIXED IGNORE
```

Now, you could enter **locate/new year/**, and XEDIT would locate **New Year**, **NEW YEAR**, or any other combination of upper and lowercase letters.

Note: You must enter a SET CASE MIXED IGNORE command each time you edit a file. If you want this command to be in effect all of the time, you can add it to your PROFILE XEDIT file (see Chapter 2).

Now you’re ready to learn the GML tags to insert in your Script files. You’ll learn tags for paragraphs, headings, lists, and highlighting (various kinds of emphasis). The GML starter set contains more tags, but these are the ones you’ll use most often.

Paragraphs

The tag you'll use most often is **P**, for "paragraph."

Like all tags, you start it with a colon (:) and end it with a period (.). You enter this paragraph like this:

```
:p.Like all tags, you start it with...
```

Or you can put the tag on a separate line, like this:

```
:p.  
Like all tags, you start it with...
```

You don't have to skip a line before the paragraph tag. When the file is formatted, a blank line is inserted automatically between paragraphs.

If you start each sentence within the paragraph on a new line, Script automatically inserts spaces between sentences when the file is formatted. (You don't have to worry about the length of the lines you type.)

Headings

Headings identify different parts of your document. For example, a heading may be a chapter title, a topic, and so forth. GML gives you seven levels of headings, H0 through H6. Each level causes a different type of formatting, so that your document can be arranged in a consistent hierarchy, like an outline.



The tags for the various heading levels are described below.

:h0.

This heading level is not used often. You use it only when you want to divide a large document into major parts, each part having several chapters. Script starts a new page when it finds this tag. You would enter a level 0 heading like this:

```
:h0.Part One
```

Notice that the tag starts with a colon (:) and ends with a period (.). A heading must fit on one line.

:h1.

This heading level can be used for chapters. Script starts a new page when it finds this tag. For example,

```
:h1.Chapter One
```

Note: When you choose either a level 0 or level 1 heading, Script puts a “running foot” at the bottom of odd-numbered pages. The running foot is the same as the text of the heading.

:h2. :h3. and :h4.

These headings appear in your document with different types of formatting (like underlining, bold printing, or capital letters) to tell them apart. For example, the following headings:

```
:h2.Example of a Head 2  
:h3.Example of a Head 3  
:h4.Example of a Head 4
```

might appear in the printed document like this:

EXAMPLE OF A HEAD 2

Example of a Head 3

Example of a Head 4

Heading levels 0 through 4 are included in the document's table of contents (if it has one). You should type them in upper and lowercase, because the way you enter them is the way they are put in the table of contents. (For information on getting an automatic table of contents, see the *Document Composition Facility: Generalized Markup Language Starter Set User's Guide*.)

:h5. and :h6.

These headings are "run-in," as opposed to all the other levels, which are "stand-alone."

For example,

```
:h5.Example of a Head 5  
:p.Level 5 is a run-in heading.
```

produces

Example of a Head 5: Level 5 is a run-in heading.

Exercise: Paragraphs and Headings

The tags you'll be using most often are paragraph and heading tags. Let's create a SCRIPT file and practice these tags.

This file will be a memo that you are planning to send to all your sales personnel. Your company is the Ezgadget Company, so the file will be called **ezgadget script a**.

To create the file, enter

```
x ezgadget script
```

Now enter the **INPUT** command and type these lines:

```
:h1.Company Newsletter
:h2.Promotions
:p.Harry Welldone, former accounting clerk, is promoted to the
copy edit department.
You'll recognize his office by the mound of eraser crumbs
on the desk, table, and floor.
Ecologically-minded Harry has suggested that he might save
the crumbs to stuff a pillow.
Good thinking, Harry!
:p.Harry is now working on the department's
Let's Spell It Rite series.
Although Harry wasn't sure that the transfer would further his
career aims, his manager assured him that erasing was one of the
finer editorial skills.
:h2.New Products
:p.As you all know, the new selling season is here.
We had a super fantastic year last year, and this year will
be even better!
We have developed some exciting new gadgets.
So let's get out there and sell, sell, sell!
```

Now enter a **FILE** command.

Displaying Your Document

You use the **SCRIPT** command to format your document according to the formatting instructions - the **GML** tags - you gave it. The **SCRIPT** command format that displays the results at your terminal is:

```
SCRIPT filename (PROFile(DSMProf3) COntinue
```



The keywords following the left parenthesis are **SCRIPT** command options. "PROFile" means a special file will handle your **GML** tags. The name shown in our examples is **DSMPROF3**, but you should ask a friend whether that is the correct file to use for your computer system. If it is not, substitute the name used on your computer system for **DSMPROF3**. The "COntinue" option tells Script to continue processing your file unless it finds a severe error.

Now enter this command on the **CMS** command line:

```
script ezgadget (prof(dsmprof3) co
```

Script displays a message similar to the following: `SCRIPT/VS
RELEASE n, LEVEL (n) - mm/dd/yy`. Then you'll see your formatted file.

Printing Your Document

If you want to print the file on the system printer instead of displaying it at your terminal, you need to add the **PRINT** option to the **SCRIPT** command, like this:

```
SCRIPT filename (PROF(DSMProf3) CO PRINT
```

Remember that you may need to substitute the name of another profile for DSMPROF3.

Correcting Mistakes in GML Tags

If you made a mistake in typing a GML tag in the file, Script will display the line number in the file and the incorrect tag. What kind of mistake might you make? Well, if you type ":g2." instead of ":h2.", Script won't recognize the tag.

To correct an error, use the XEDIT command to get a copy of the file on your screen. Then, to find the error quickly, you can use the LOCATE command followed by the incorrect tag (for example, locate/:g2./). When using the LOCATE command, be sure to type the incorrect control word exactly as it is shown in the error message. Then you can use a CHANGE command, or you can correct the mistake simply by typing over it.

There's another way to locate the incorrect control word. In its error message, Script gives you the number of the line containing the incorrect control word. You can make that line current by typing a colon (:) followed by the line number in the large arrow command line, like this:

```
====> :7
```

(Don't type locate/7 because the editor would then look for the number 7 throughout the file.) When you press the ENTER key, line 7 becomes the new current line. You could then type over the error to correct it. Or, you could use the CHANGE command.

Lists

After paragraphs and headings, you'll mostly be doing lists. Making lists - and maintaining them - is easy with GML. GML has tags for four kinds of lists: simple, unordered, ordered, and definition.

Simple List

A simple list is just a list of items in no special order, like a grocery list. Here's an example of a simple list:

- 8 ears of corn
- 8 tablespoons butter
- salt and pepper

You can also have a "compact" simple list, with no spaces between the list items:

- cherries
- cognac
- sugar
- butter
- eggs
- oranges
- whipping cream

Unordered List

An unordered list is similar to a simple list, but it's used when the items in the list are fairly long. Dots, called "bullets," appear before each item. Here's an example:

- This is an item in an unordered list. It has a bullet beside it.

This item consists of two paragraphs. This paragraph does not get a bullet because it is not a separate list item.

- This is the next list item in our unordered list.

Ordered List

An ordered list is numbered. For example,

1. Slice the kernels from the corn.
2. Melt butter in a 10-inch skillet.
3. Add the corn and saute gently about 5 minutes, stirring and tossing with a fork.
4. Season with salt and pepper.

Unordered and ordered lists can also be compact.

List Tags

Each of these list types starts with a tag that tells what kind of list it is and ends with a matching end tag. End tags are easy. You just insert an "e" after the colon (:), and the rest of the tag remains the same. Here are the tags:

Tags	List Type
:sl. :esl.	simple list
:ul. :eul.	unordered list
:ol. :eol.	ordered list

Within the list, each item must be tagged with the list item tag:

```
:li.
```

Here's how you would enter the simple list shown above:

```
:sl.  
:li.8 ears of corn  
:li.8 tablespoons butter  
:li.salt and pepper  
:esl.
```

To make a simple list compact, you'd add the word "compact" to the beginning list tag, like this:

```
:sl compact.  
:li.cherries  
:li.cognac  
:li.sugar  
:li.butter  
:li.eggs  
:li.oranges  
:li.whipping cream  
:esl.
```

Here's how we marked up our unordered list:

```
:ul.  
:li.This is an item in an unordered list.  
It has a bullet beside it.  
:p.This item consists of two paragraphs.  
This paragraph does not get a bullet because  
it is not a separate list item.  
:li.This is the next list item in our unordered list.  
:eul.
```

Notice that the second paragraph in the first list item is marked with a P tag. You can put paragraphs within lists, and still get the correct spacing and indenting, automatically.

To make an unordered list compact (no space between list items), you'd include the word "compact" on the beginning tag, like this:

```
:ul compact.  
.  
.  
.  
:eul.
```

And here's how we marked up the ordered list:

```
:ol.  
:li.Slice the kernels from the corn.  
:li.Melt butter in a 10-inch skillet.  
:li.Add the corn and saute gently about 5 minutes, stirring  
and tossing with a fork.  
:li.Season with salt and pepper.  
:eol.
```

Again, we could have used the word "compact," just as we did with the other lists.

By now, you can see how easy it is to create lists with GML tags. It's even easier to change them. For example, with an ordered list, Script numbers the list items automatically. And if you want to add a list item, just put one in. Script automatically rennumbers the list! If

you want to remove a list item, just delete that line. The list will be renumbered, automatically.

Exercise: Lists

Let's add a list of self-improvement courses to our Ezgadget Company newsletter.

Get a copy of the file on your screen:

```
x ezgadget script
```

We'll add this list to the end of the file, so enter a **BOTTOM** command. Then enter an **INPUT** command.

Now type these lines:

```
:h2.Company Courses
:ol.
:li.Cutting Out Paper Dolls.
Bring a pair of blunt scissors to the first class.
Advanced students will be allowed to color.
:li.Ways to Use Leftover Hamburger to Make
Fabulous Gourmet Dishes.
:li.Making Plans to Swim the English Channel.
A fun way to get in shape.
:eol.
```

Press the **ENTER** key twice. Then enter a **FILE** command. To see the results at your terminal, enter:

```
script ezgadget (prof(dsmprof3) co
```

Remember that you may need to substitute the name of another profile for DSMPROF3. If you want to print the file, add "PR" to the **SCRIPT** command.

Definition Lists

There's one more kind of list that you'll find useful. A definition list is the kind of list where you pair a term or phrase with its description.

Here's an example of a definition list:

gopher A burrowing rodent that feeds on roots of plants.

lawn A gopher highway.

tiger lily Lovely flowering plant, the roots of which are the preferred food of gophers.

To mark up a list like this, you must first decide how many character spaces the term will take up (on the left hand side). If the term is no more than 10 spaces, you don't have to worry about it. Here, we'd need to reserve 12 spaces to give us enough room for "tiger lily" and the two extra spaces that follow it.

Here are the tags for a definition list:

```
:dl.      starting tag
:dt.      tag for each term
:dd.      tag for each definition
:edl.     end tag
```

In addition, you specify the number of characters in the term (if it's greater than 10) on the starting tag, like this:

```
:dl tsize = n.
```

where "n" is the number of characters in the term. "TSIZE" stands for "term size." Notice that you put the period **after** the term size.

Here's how you'd mark up the definition list shown above:

```
:dl tsize=12.  
:dt.gopher  
:dd.A burrowing rodent that feeds on roots of plants.  
:dt.lawn  
:dd.A gopher highway.  
:dt.tiger lily  
:dd.Lovely flowering plant, the roots of which are the preferred  
food of gophers.  
:edl.
```

In short, a definition list starts with the DL tag. Within the list, you tag the terms with the DT (definition term) tag and their descriptions with the DD (definition description) tag. The end tag is EDL.

Exercise: Definition List

Let's add the list of new products to our newsletter. Enter:

```
x ezgadget script
```

Make current the line that begins, "So let's get out there and sell...."
Enter an INPUT command, then type these lines:

```
:dl tsize=13.  
:dt.gizmo  
:dd.A gizmo can fix anything, from broken windows to  
broken toys.  
And when you fix broken toys, you fix a broken heart at the  
same time.  
:dt.thingamabob  
:dd.Shreds lettuce, carrots, incriminating letters, tin cans,  
or anything you can think of.  
:dt.eeezzee opener  
:dd.Nothing is impossible to open with this  
wonderful new household tool.  
Opens jars, bottles, cans, safes, sticky doors, and  
hard hearts.  
:edl.
```

Enter a **FILE** command. To see the results at your terminal, enter:

```
script ezgadget (prof(dsmprof3) co
```

Remember that you may need to substitute the name of another profile for DSMPROF3. If you want to print the file, add "PR" to the SCRIPT command.

Highlighting

Highlighting means giving special emphasis to a word or phrase, so that it can be contrasted with the rest of the text. You can highlight something in various ways, like underlining or using bold type or italic type.

GML has four tags for highlighting. They all consist of HP (to start the highlighted phrase) plus the number for the type of highlighting you want. They all have a matching end tag. The type of highlighting you get with each of the HP tags depends on the type of terminal or printer you're using. (For example, you can't get italics on a display terminal.)

The beginning and ending tags are shown below, along with the type of highlighting you get if you display the file at your terminal.

:hp1.	:ehp1.	Underlining.
:hp2.	:ehp2.	Capital letters.
:hp3.	:ehp3.	All capitals and underlined.
:hp0.	:ehp0.	No emphasis.

On a printer that has italics and bold type available, HP1, HP2, and HP3 mean *italics*, **bold type**, and **bold italics**, respectively.

So if you wanted to do something like this:

Attention! This is *very* important! ***Don't go out in the rain without your galoshes!***

you would enter:

```
:p.:hp2.Attention!:ehp2.  
This is :hp1.very:ehp1. important!  
:hp3.Don't go out in the  
rain without your galoshes!:ehp3.
```

Remember, the type of emphasis you get depends on the type of printer or terminal you're using. However, regardless of the output device, the different HP tags will give you different kinds of emphasis.

Exercise: Highlighting

Let's highlight a phrase in our Ezgadget Company newsletter.

First get a copy of the file on your screen:

```
x ezgadget script
```

Locate the line that contains "Let's Spell It Rite." Using the insert mode, insert the tags to highlight this phrase, like this:

```
:hp3.Let's Spell It Rite:ehp3. series.
```

Now enter a `FILE` command and then a `SCRIPT` command to display the results at your terminal.

Checking Your Spelling

If you'd like to check your document for spelling errors, you can add the SPELLCHK option to the SCRIPT command. For example,

```
SCRIPT filename (PROF(DSMPROF3) CO SPELLCHK
```

Because checking your spelling requires additional processing time (each word is checked), you should use this option when your document is otherwise complete.

Script will display the words it doesn't find in its dictionary. Some of the words may be spelled correctly but are listed because they are not in the dictionary. For each line that contains words that aren't in the dictionary, you'll get the message WORDS NOT FOUND followed by the words that Script couldn't find. Below each message will be the filename in quotation marks, the line number, and the full text of the line containing the misspelled word (or words).

After Script shows you all the lines that contain words not found in its dictionary, it shows you the formatted file. Then you can use the XEDIT command to get a copy of the file on your screen and correct any misspelled words.

The Ezgadget Newsletter

If you have done the exercises in this section, your printed file may look like this. Your file may look slightly different, depending on the type of printer you have.

COMPANY NEWSLETTER

PROMOTIONS

Harry Welldone, former accounting clerk, is promoted to the copy edit department. You'll recognize his office by the mound of eraser crumbs on the desk, table, and floor. Ecologically-minded Harry has suggested that he might save the crumbs to stuff a pillow. Good thinking, Harry!

Harry is now working on the department's *Let's Spell it Rite* series. Although Harry wasn't sure that the transfer would further his career aims, his manager assured him that erasing was one of the finer editorial skills.

NEW PRODUCTS

As you all know, the new selling season is here. We had a super fantastic year last year, and this year will be even better! We have developed some exciting new gadgets. So let's get out there and sell, sell, sell!

- gizmo** A gizmo can fix anything, from broken windows to broken toys. And when you fix broken toys, you fix a broken heart at the same time.
- thingamabob** Shreds lettuce, carrots, incriminating letters, tin cans, or anything you can think of.
- eezzee opener** Nothing is impossible to open with this wonderful new household tool. Opens jars, bottles, cans, safes, sticky doors, and hard hearts.

COMPANY COURSES

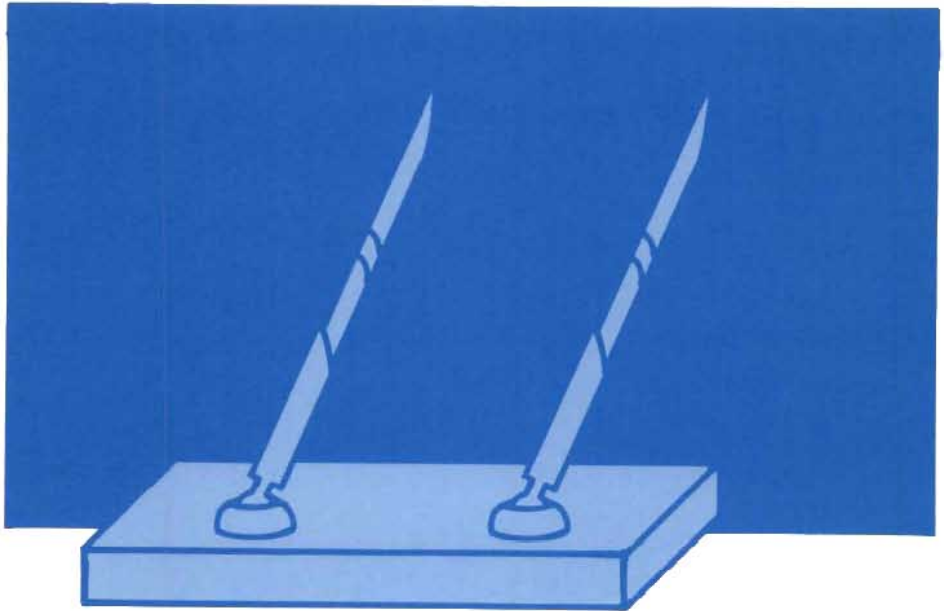
1. Cutting Out Paper Dolls. Bring a pair of blunt scissors to the first class. Advanced students will be allowed to color.
2. Ways to Use Leftover Hamburger to Make Fabulous Gourmet Dishes.
3. Making Plans to Swim the English Channel. A fun way to get in shape.

Company Newsletter 1

Summary of GML Tags

Text Element	GML Tag	End Tag
paragraph	:p.	
simple list	:sl.	:esl.
unordered list	:ul.	:eul.
ordered list	:ol.	:eol.
list item	:li.	
definition list	:dl.	:edl.
definition term	:dt.	
definition descr.	:dd.	
headings (7 levels)	:h0. - :h6.	
highlighting (4 levels)	:hp0. - :hp3.	:ehp0. - :ehp3.

Chapter 8. Creating and Using 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 chapter, you’ll learn how to create simple EXECs. You’ll learn some of the features of the Restructured Extended Executor (REXX) language, which has special keywords called *instructions* that are used to tailor your EXECs to meet your needs.

More information on EXECs can be found in the *VM/SP System Product Interpreter User’s Guide* and the *VM/SP System Product Interpreter Reference*.

An EXEC to Print a File

In Chapter 4, you learned to 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:

```
x myprint exec
```

In our examples, we will show command keywords and REXX keywords in capital letters, and everything else in lowercase letters. If you want to use “mixed case” in your EXEC files, enter the command SET CASE MIXED or put that command in a PROFILE XEDIT file (see “Creating an XEDIT Profile” on page 101). (You will also notice that in EXECs, we use quotation marks around command keywords. We’ll explain more about this later.)



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.

```
/* This exec prints 5 copies of a file. */  
"SPOOL PRINT COPY" 5  
"PRINT" crime theft  
"SPOOL PRINT COPY" 1
```

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

```
/* This exec prints 5 copies of a file. */  
The first line in an EXEC must be a comment. It must start with /* and end with */. The comment itself can be anything you like that describes your EXEC. The comment line calls on the System Product Interpreter, which is the part of the VM system that handles EXECs written in the REXX language.
```

```
"SPOOL PRINT COPY" 5  
is the command you use to get more than one copy printed. In this case, you are asking that five copies be printed.
```

```
"PRINT" crime theft  
This command gives the filename and filetype of the file you want printed.
```

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

Enter a **FILE** command. The EXEC is now on your disk. How do you use it? Just type the filename **MYPRINT** on the CMS 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 additions and changes:

```
/* This prints any number of copies of any file. */
SAY "Enter number of copies."
PULL num
SAY "Enter filename and filetype."
PULL fn ft
"SPOOL PRINT COPY" num
"PRINT" fn ft
"SPOOL PRINT COPY" 1
```

SAY is a REXX instruction that displays a message on the screen. The message can ask you to enter information (that is, it can be a "prompt" message). In this example, the first SAY instruction asks the user to enter the number of copies. The second SAY instruction asks the user to enter the filename and filetype. Notice that the prompt should be enclosed in quotation marks.

PULL is a REXX instruction. The words after the PULL instructions (num, fn, and ft) are names of **variables**. You can use any names you'd like for variables. They are called variables because their value varies, depending on what you type in response to the prompt.

A PULL instruction gets values from the terminal and assigns them to these variables. The first PULL instruction assigns the value you type to **num**. The second PULL instruction assigns the first word you type to **fn** and the second word you type to **ft**.

Let's look at the other changes you made. In the SPOOL command, you changed the number of copies to **num**. In the PRINT command,



you changed the filename and filetype (crime theft) to **fn** and **ft**. The values that you type in response to the prompts replace these variable names in the commands.

Now when you enter MYPRINT, you will be prompted for the number of copies you want printed, the filename, and the filetype of the file you want printed. The values you type will replace **num**, **fn**, and **ft** in the EXEC.

Let's see how this works. Enter a **FILE** command. Then enter this command to start the EXEC:

```
myprint
```

Respond to the first prompt with the number of copies you want, and respond to the second prompt with the filename and filetype of one of your files. For example:

```
System asks: Enter number of copies.
```

```
You type: 3
```

```
System asks: Enter filename and filetype.
```

```
You type: crime theft
```

The PULL instruction takes the values you enter and assigns them to **num**, **fn**, and **ft**.

This command:

```
"SPOOL PRINT COPY" num  
"PRINT" fn ft
```

**Becomes this command
for this execution:**

```
SPOOL PRINT COPY 3  
PRINT CRIME THEFT
```

You can use a different number and a different filename and filetype each time you enter MYPRINT.

An EXEC to Print Script Files

In Chapter 7, you learned how to create Script files containing GML tags. You use the SCRIPT command to format and print these files. Let's create an EXEC to print SCRIPT files.

To format and print five copies of a Script file whose filename is MEETING, you'd normally enter these two commands:

```
SPOOL PRINT COPY 5
SCRIPT MEETING (PROF(DSMPROF3) CO PR
```

For DSMPROF3 you would substitute the name of the profile used on your computer system.

Our EXEC will prompt you for the number of copies and the filename (the file to be printed must have a filetype of SCRIPT). Let's call your EXEC file MYSCRIPT EXEC A. Enter:

```
x myscript exec
```

Enter an INPUT command and type:

```
/* This exec formats and prints Script files. */
SAY "Enter number of copies to be printed."
PULL numcopy
"SPOOL PRINT COPY" numcopy
SAY "Enter the filename of the Script file."
PULL fname
"SCRIPT" fname "(PROF(DSMPROF3) CO PR"
"SPOOL PRINT COPY" 1
```

Press the ENTER key twice to get out of input mode. Let's take a closer look at these lines:

```
SAY "Enter number of copies to be printed."
```

The SAY instruction tells the EXEC to display the message.



`PULL numcopy`

The `PULL` instruction waits for you to type a number and press the `ENTER` key. It assigns the value you type to the variable ***numcopy***.

`"SPOOL PRINT COPY" numcopy`

Now you see how the variable is used. If you entered "5" in response to the prompt, ***numcopy*** would have the value 5, and you'd get five copies of the Script file printed.

`SAY "Enter the filename of the Script file."`

This tells the `EXEC` to display the message.

`PULL fname`

The `PULL` instruction waits for you to type a filename and press the `ENTER` key. It takes the filename you enter and assigns it to the variable ***fname***. This variable is used in the `SCRIPT` command.

`"SCRIPT" fname "(PROF(DSM PROF3) CO PR"`

Notice that quotation marks enclose the command keywords. The variable name (`fname`) is not enclosed in quotation marks.

Now enter a **FILE** command. To try out your `EXEC`, just enter **MYSCRIPT**. When the `EXEC` prompts you for information, type it and press the `ENTER` key. Make sure you use the filename of a Script file. The ready message (Ready;) will indicate when the `EXEC` is finished.

When Should I Use Quotation Marks?

You should use quotation marks around CMS and CP command keywords in EXECs to set them off from variable names. Because you can use any name for a variable name, this prevents a potential mix-up between keywords and variables that have the same name. It also makes your EXEC easier to read.

In addition, when a command has special characters they must be enclosed in quotation marks. (See, for example, the portion of the above SCRIPT command that contains parentheses.)

Do **not** use quotation marks in the REXX instructions, except where specified (for example, around the prompt message).

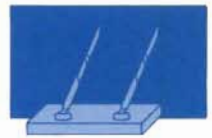
An EXEC to Link to Someone Else's Disk

In Chapter 3, you learn that to share someone else's files, you need a LINK command and an ACCESS command. Let's say you want to share someone's files often. His userid is HAROLD. His files are on his disk number 191. Harold told you his read password is "123456." You could enter these two commands:

```
link to harold 191 196 RR 123456
access 196 b
```

To create an EXEC that links to Harold's disk, you could enter XEDIT HAROLD EXEC and put these two commands in the file. By entering HAROLD, the EXEC filename, you execute those commands.

But let's create an EXEC that links to any user's disk, provided he's given us his password and disk number that contains the files we want to see. We'll call it SHARE EXEC.



Enter

```
x share exec
```

Enter an **INPUT** command and type these lines:

```
/* This exec will link to someone else's files. */  
SAY "Enter the userid."  
PULL userid  
SAY "Enter the disknumber."  
PULL disknumber  
SAY "Enter the password."  
PULL password  
"LINK TO" userid disknumber "196 RR" password  
"ACCESS 196 B"
```

Press the **ENTER** key twice to return to edit mode. Then enter a **FILE** command.

To use this EXEC, you simply enter **SHARE**. For example, to use this EXEC to link to Harold's disk, you enter:

```
share
```

After being prompted for the userid, you enter:

```
harold
```

After being prompted for the disk number, you enter:

```
191
```

After the last prompt, for the password, you enter:

```
123456
```

In the EXEC, HAROLD will be assigned to **userid**, 191 to **disknumber**, and 123456 to **password**.

PROFILE EXEC

The EXECs you have created are executed when you enter the filename of the EXEC. One exception is an EXEC with the file label PROFILE EXEC. Not only is the filetype special, but also the filename is special. Do not use the filename PROFILE when you're creating regular EXECs.

A PROFILE EXEC is executed automatically when you enter the IPL CMS command (or if you have automatic IPL). Usually, your VM administrator will supply you with a PROFILE EXEC when you receive your userid. This EXEC contains the necessary "housekeeping" commands that set up your VM system, like commands to access necessary disks, commands that describe your terminal and printer, and so forth.

Chapter 9. Creating and Running Programs



A program is a set of instructions written in a language the computer can understand. Some of the programming languages used with the VM system are BASIC, COBOL, FORTRAN, and PL/I.

In this chapter, you will learn how to create and run programs, regardless of the programming language you use. You don't even have to know a programming language to complete this chapter. Just follow the examples.

You should know how to edit files using XEDIT (Chapter 2) before reading this chapter.

For more information on the procedures and commands discussed in this chapter, see the *VM/SP Application Development Guide* and the *VM/SP CMS Command Reference*. For information on how to write programs in the various languages, ask your supervisor what IBM manuals to use.

Creating a Program

Note: Follow the instructions in this part of the chapter for all programming languages except IBM BASIC. For information on creating and running BASIC programs, see "Using IBM BASIC" on page 227.

A program is a file that contains instructions for the computer. You create a program file with an XEDIT command, the same way you create any other file. Let's review its format:

```
Xedit filename filetype
```

A program file can have any filename that is no longer than eight characters. But its filetype indicates the name of the programming language you are using. Some of the special programming language filetypes used in VM are:

COBOL
FORTRAN
PLI

Based on the filetype, the editor gives values to various settings, like tabs. For example, suppose a programming language requires you to enter data in certain columns. The tabs are set according to that



language, which makes it easy to line up the data using PF4 (the tab key). For a complete list of these settings, see the *VM/SP System Product Editor Command and Macro Reference*.

Because many students, scientists, engineers, and other professionals use FORTRAN for problem solving, we have chosen a simple FORTRAN program to use for our example. Your system must have VS FORTRAN for the example to work. To find out if you have VS FORTRAN, either ask your system administrator, or enter

```
filel fortvs2 * *
```

If the file list contains FORTVS2 MODULE, you can do the example. If not, you can still use the procedures explained in this chapter, but you'll have to use a programming language that is in your system. If you are in the FILELIST environment, **press PF3** to leave it.

Our program converts a Fahrenheit temperature to Celsius. The program asks you to enter a Fahrenheit temperature with a decimal point (for example, 32.0). It then displays that temperature in Celsius.

Now enter:

```
x degrees fortran
```

Enter the following command to enter input mode:

```
====> i
```

Now type these lines, exactly as shown.

```
00010 WRITE (6,15)
00015 FORMAT (' ENTER FAHRENHEIT VALUE WITH DECIMAL POINT')
00020 READ (5,25)VALUE
00025 FORMAT (F4.1)
00030 OUTPUT = (VALUE - 32 ) / 1.8
00040 WRITE (6,45)OUTPUT
00045 FORMAT (F6.1,' DEGREES CELSIUS')
00050 END
```

You are finished entering your program instructions. Press the ENTER key twice to enter edit mode. Then enter:

```
====> file
```

Now you have a **source program**.

Compiling a Program

If you know FORTRAN, your source program is meaningful to you, but it means nothing to the computer. The computer only understands machine language. You must now **compile** your program. That means you ask a compiler to translate the program into machine language.

Before you compile your program, you must make sure you have enough virtual storage for the compiler. You probably have enough storage for all compilers except VS FORTRAN, which is used in our example. To find out how much virtual storage you have, enter:

```
query storage
```

You will see a message like one of the following:

```
STORAGE = 512K or STORAGE = 1024K or STORAGE = 2048K
```

If the number is less than 3060 and you are using VS FORTRAN, you need to get more storage. Here's how:

1. Enter this command:

```
define storage 3m
```

If you see the message, STORAGE EXCEEDS ALLOWED MAXIMUM, ask your system administrator to increase your

Before we try out the program, let's see a list of files that have the filename DEGREES. Enter:

```
filel degrees
```

Your list should contain these files:

```
DEGREES FORTRAN  
DEGREES LISTING  
DEGREES TEXT
```

DEGREES FORTRAN is the file you created. But where did the other two files with the filename DEGREES come from? When your program is compiled, two new files are created. Both have the filename of your program. What are these files? The file DEGREES LISTING contains your source program and a list of any errors that were found. You can look at this file by using an XEDIT command. You can also get a printed copy by using the PRINT command, which is explained in Chapter 4.

The file DEGREES TEXT contains the machine language version of your program. This file is loaded into storage when you run the program.

Both the LISTING and TEXT files are replaced if you change the program and compile it again.

Now you can press PF3 to leave the FILELIST environment.



Running a Program

Before you can run your program, you must tell VM the names of certain “libraries” that the system needs to run your program. You do that with the GLOBAL command. It has the following formats:

```
GLobal TXTLIB library-name...
GLobal MACLIB library-name...
GLobal LOADLIB library-name...
```

The exact GLOBAL command you need depends on the language you are using and your installation. Ask your VM administrator to tell you which GLOBAL command(s) to use.

For example, the necessary GLOBAL commands for VS FORTRAN programs might look like this:

```
global txtlib vsf2fort cmslib
global loadlib vsf2load
```

When you find out which GLOBAL commands to use, it’s a good idea to include them in your PROFILE EXEC, which is discussed in “PROFILE EXEC” on page 218. That way, you won’t have to enter GLOBAL commands in each terminal session.

To run your program, you use a command that loads your program into storage and runs (executes) it. It has this format:

```
LOAD filename (START
```

Now, to run your program, enter

```
load degrees (start
```

You see

```
EXECUTION BEGINS...  
ENTER FAHRENHEIT VALUE WITH DECIMAL POINT
```

Now enter a value with a decimal point. For example, enter:

```
32.0
```

You see

```
    0.0 DEGREES CELSIUS  
Ready; T=0.1/0.1 11:12:09
```

The ready message (Ready;) means that your program has completed.

Suppose you want to run the DEGREES program again. Do you have to compile it again? No, all you have to do is enter the LOAD command, because the machine language version of your program (the DEGREES TEXT file) is on your disk.

Note: Another way to run a program is to create a "module" using the GENMOD (generate module) command. You might choose this way when your program is free of errors, and you want to run it often or send it to other people to use. For information on creating a module, see the *VM/SP Application Development Guide*.



Using IBM BASIC

BASIC is one of the most popular programming languages. It is easy to learn and use and is suitable for a wide variety of applications. IBM BASIC is designed to be used with your VM/SP system.

Creating a BASIC Program

You can use XEDIT to create a BASIC source program. However, an easier way is to use BASIC's "interactive mode." In that mode, BASIC prompts you to enter your program instructions, a line at a time. For information on all the things you can do in interactive mode, see the IBM BASIC manuals. For now, just follow the instructions to create a BASIC program.

Our program converts a Celsius temperature to Fahrenheit. The program asks you to enter a Celsius temperature. It then displays that temperature in Fahrenheit.

To enter BASIC interactive mode, simply enter:

```
basic
```

If you receive the message "Loader table overflow," enter this command:

```
SET LDRTBLS 4
```

and enter the BASIC command again. IBM BASIC lets you know it's ready to start work by answering:

IBM BASIC/VM VERSION v RELEASE r yyyy/mm/dd hh:mm

*

↑

cursor

The * shows you where to begin typing your program. Each time you complete a line, press the ENTER key. Now enter:

```
* 100 PRINT 'ENTER CELSIUS VALUE TO BE CONVERTED TO FAHRENHEIT'  
* 110 INPUT VALUE  
* 120 OUTPUTV = VALUE * 1.8 + 32  
* 130 PRINT OUTPUTV; ' DEGREES FAHRENHEIT'  
* 140 PRINT 'PRESS ENTER TO EXIT PROGRAM'  
* 150 INPUT X$  
* 160 END
```

Correcting Mistakes

You should check your typing before you do anything more with your program. If you made mistakes, it's easy to correct them in BASIC interactive mode. For example, suppose you typed this incorrect line:

```
110 INPUD VALUE
```

You can correct it simply by entering a new line 110, as follows:

```
* 110 INPUT VALUE
```

When you press the ENTER key, the new, corrected line 110 replaces the old line 110.

A message from the compiler tells you whether or not you made any errors while typing your program. To correct any errors, you:

- Use XEDIT to edit the source program. For our example, you would enter
`x temp basic`
- Correct the errors and enter a FILE command.
- Compile the program again. For our example, you would enter
`basic temp`

Running a BASIC Program

To run your program, use the BASICRUN command followed by the filename. Now enter:

```
basicrun temp
```

command that runs your BASIC program

filename

You see:

```
ENTER CELSIUS VALUE TO BE CONVERTED TO FAHRENHEIT  
? _
```

The program “prompts” you for the data it needs (the Celsius value). To show you it’s waiting for you to type something, it displays a question mark (shown above). It also positions the cursor where it wants you to enter the data.

Now enter a Celsius value. For example:

```
? 0
```



When you press ENTER, you see

```
32 DEGREES FAHRENHEIT
PRESS ENTER TO EXIT PROGRAM
? _
```

Your program is complete, so press the ENTER key. You will see the ready message (Ready;) near the top of the screen.

Summary

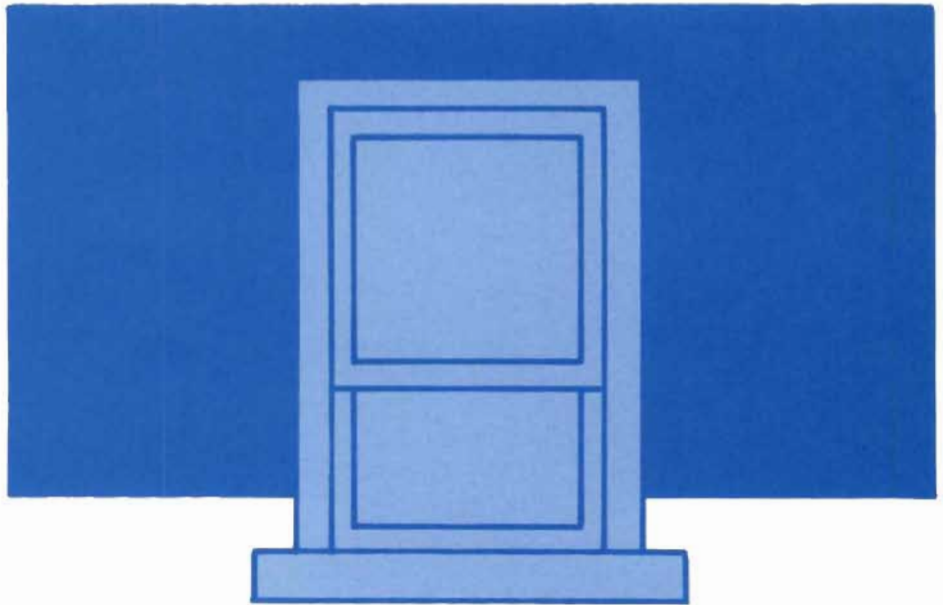
Follow these steps to create, compile, and run a program (other than IBM BASIC):

Command	Function
Xedit filename filetype	Create a program. Filetype identifies the programming language.
compiler filename	Compile the program. Use a compiler command followed by filename of the program (like FORTVS DEGREES).
GLobal TXTLIB library... GLobal MACLIB library... GLobal LOADLIB library...	Identify libraries.
LOAD filename (START	Run the program.

Follow these steps to create, compile, and run an IBM BASIC program:

Command	Function
BASIC	Enter BASIC interactive mode and create program.
* SAVE filename	Save program on disk.
* QUIT	Leave BASIC interactive mode.
BASIC filename	Compile the program. Use XEDIT to correct any errors (XEDIT filename BASIC).
BASICRUN filename	Run the program.

Chapter 10. Looking at VM/SP Through Windows

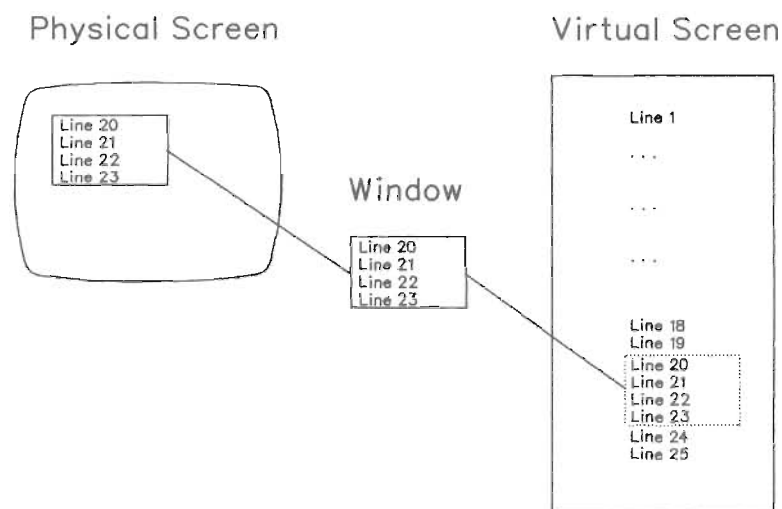


So far in our use of CMS, we've had to enter commands like XEDIT, FILELIST, and RDRLIST on a line near the bottom of the screen. Otherwise, the keyboard would lock.

CMS has another way of working called **full-screen CMS**. When using full-screen CMS, you can type commands almost anywhere on the screen. You can also manage several pieces of information on the screen at the same time, in much the same way that several pieces of paper can be arranged on a desk top. To make this

“electronic desk top” possible, CMS uses *windows* and *virtual screens* (or *vscreens*).

A *window* is an area of your terminal screen in which data is displayed. One window might show messages while another might show the results of commands you’ve entered. More than one window can appear on your screen at the same time. Like the pieces of paper on your desk top, windows are various sizes, often overlap each other, and can be easily moved about.



What you see in a window is actually data that CMS places in *virtual screens*, as shown above. A virtual screen can be thought of as a “presentation space.” Virtual screens can be various sizes, and are often larger than the window used to view them.

CMS keeps track of several virtual screens for you and places a different kind of data in each. For example, CMS puts your messages in one virtual screen and the results of your commands in another. To see the data in these virtual screens, you use windows.



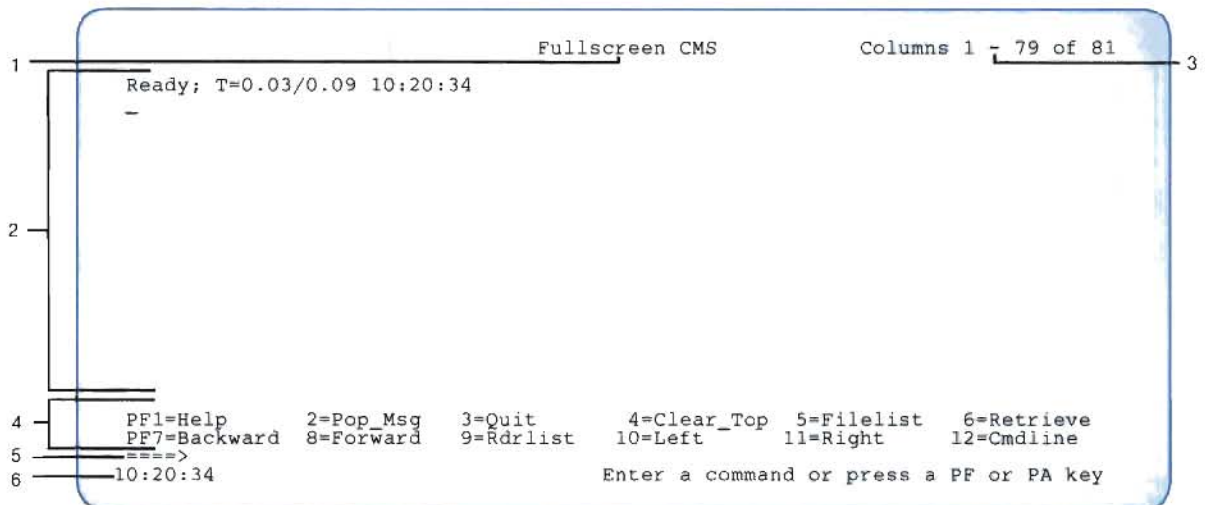
This chapter describes how to use some of the features of full-screen CMS. For more information about using full-screen CMS, refer to the *VM/SP CMS User's Guide* and the *VM/SP CMS Command Reference*.

Starting Full-Screen CMS

Let's start full-screen CMS. Enter the following command:

```
set fullscreen on
```

CMS changes your screen to look like this:



Displayed on your screen is the **CMS window**. You can think of the CMS window as the “home base” from which you’ll start all your full-screen CMS work and to which you’ll always return. One special feature of the CMS window is that it allows you to enter commands almost anywhere on the screen.

Before we continue, let's look at the parts of the screen:

1 Title Line

The title line tells you that you are in full-screen CMS.

2 Command Input and Output Area

You can type commands on any line in this area. This is also where CMS output is displayed. You can scroll forward and backward through the data displayed in this area.

3 Scroll Location Information Area

In this area CMS displays window scroll location information. Two kinds of information may be displayed: column location and line location.

Column location information looks like this:

```
Columns 1 - 79 of 81
```

This means there are 81 columns in the virtual screen and you are looking at columns 1 to 79. To see data in other columns, you must scroll the window to the right and then back again to the left. In the CMS window, the PF10 key scrolls left and the PF11 key scrolls right.

Column location information is displayed when there is any possibility that virtual screen data is to the left or right of your window.

Line location information looks like this:

```
Lines 20 - 28 of 28
```



This means there are 28 lines of data and you are looking at lines 20 through 28.

Line location information is displayed only when you are not seeing all the lines in the virtual screen. To see the other lines, you must scroll the window forward or backward. In the CMS window, the PF8 key scrolls forward and the PF7 key scrolls backward.

4 CMS PF Key Settings

These two lines give you a one-word description of what the CMS PF keys will do when you press them. CMS PF keys are explained later in this chapter.

5 CMS Command Line

You can enter commands on this line.

6 Status Line

On this line, VM tells you three things: the time of day, whether there are any messages for you, and what VM is doing or waiting for you to do.

The leftmost part of the status line is a digital clock display. It shows hours, minutes, and seconds.

To the right of the digital clock, CMS displays notices that tell whether there are any messages for you. When CMS receives a message, it displays one of the following notices to tell you what kind of message it has received:

Message

This notice means the message is from another computer user like yourself.

Warning

This notice means the message is important. Usually someone who works in your computer center sends warning messages.

Network

This notice means the message is from another computer system. You may get network messages if you send files to or receive files from users who work on a different computer system.

If there are no messages of any kind for you, no notice is displayed. We'll be talking more about messages later.

The rightmost part of the status line tells you what is happening in the system or what the system expects you to do next. One of the following notices may appear:

Enter a command or press a PF or PA key

VM is waiting for you to enter a command or press a PF or PA key.

Executing a command

VM is processing your command.

Enter your response in vscreen 'vname'

VM is waiting for you to reply to a request.

Note: In this message, 'vname' will be replaced with the name of a virtual screen.



Scroll forward for more information in vscreen '*vname*'

VM has more information to show you, but you must *scroll* the window to look at it. To scroll the CMS window forward, press the PF8 key. To scroll the window backward, press PF7.

There are windows other than the CMS window. We'll discuss how to scroll them later.

Note: In this message, '*vname*' will be replaced with the name of a virtual screen.

Usually you will receive only the above notices. When you are running programs, however, it is also possible to receive the status notices described in Chapter 1.

Working with Windows

We mentioned that CMS has windows other than the CMS window. In this section, we'll work with some of those windows.

The following is a list of all windows that CMS automatically maintains for you. Each window allows you to see data that CMS has written on a virtual screen.

CMS

this window looks at the CMS virtual screen. It's our "home base." The CMS virtual screen holds up to 120 lines of commands and command results.

MESSAGE

this window looks at the MESSAGE virtual screen. As one might expect, CMS writes messages from other users on

the MESSAGE virtual screen. The MESSAGE virtual screen holds up to 20 lines of messages.

WARNING

this window looks at the WARNING virtual screen, in which warnings are recorded. A warning, you'll recall, is an important message usually sent by someone in your computer center. The WARNING virtual screen holds up to four lines of warning messages.

NETWORK

this window looks at the NETWORK virtual screen, in which messages from other computer systems are recorded. The NETWORK virtual screen holds up to 16 lines of network messages.

CMSOUT

this is a small version of the CMS window that's used while you're editing files or using other functions that take the entire screen. It looks at information in the CMS virtual screen. We'll be using this window later.

STATUS

The status line really isn't part of the CMS window (we stretched the truth earlier). The status line is a one-line window that looks at the STATUS virtual screen. We never work with the STATUS window directly. CMS handles that for us.

WM

this window is a kind of super window. We'll be using it later to manage the other windows. It looks at the WM virtual screen.

We'll be working mostly with the MESSAGE window in this chapter, but the same principles apply to other windows as well.

When you send a message to yourself, the MESSAGE window appears immediately. When the message is from someone else, the "Message" status notice appears, but not the window. It is displayed the next time you press a key, like ENTER or a PF key, that gets the computer's attention.

Most windows are similar to the one shown above. That is, they may partially cover whatever you were doing, and will be surrounded by a border.

Now send another message to yourself. But, instead of typing the whole message over, let's use another feature of full-screen CMS. Move the cursor to the line that shows the TELL command you just typed. Now type "wrench!" *right over* the word "hammer!" and press the ENTER key. CMS executes the command. Your screen should look like this:

```

                                     Fullscreen CMS                               Columns 1 - 79 of 81
Ready: T=0.03/0.07 10:24:57
tell * Don't force it, use a bigger hammer!
Ready: T=0.04/0.08 10:25:13
tell * Don't force it, use a bigger wrench!
Ready: T=0.05/0.10 10:25:54
-
+-----+
|                                     Messages                                     |
| 10:25:13 MSG FROM SNOWHITE: Don't force it, use a bigger hammer!           |
| 10:25:54 MSG FROM SNOWHITE: Don't force it, use a bigger wrench!           |
+-----+

PF1=Help      2=Pop_Msg    3=Quit      4=Clear_Top  5=Filelist  6=Retrieve
PF7=Backward  8=Forward   9=Rdrlist 10=Left     11=Right   12=Cmndline
====>
10:25:54 Message                                     Enter a command or press a PF or PA key
```

You can execute any command in the CMS window by typing over it and pressing ENTER. To re-execute the same command, type over any letter before pressing ENTER. Try it. Move the cursor to the first



TELL command you entered and retype the “t” in TELL. Then press ENTER. The message will appear in the MESSAGE window.

Let’s send a few more messages so we can see how scrolling works. In the first TELL command, replace “Don’t force it” with “Don forced it.” In the second TELL command type “buy” over “use.” Then press the ENTER key. CMS executes both commands. You’ll see two more messages.

For the final three messages, let’s use the Retrieve function of full-screen CMS. The PF6 key is set to the Retrieve function. Now press it. The last command CMS executed will appear on the large arrow command line. Press the PF6 key again. The next-to-last command will appear. Keep pressing the PF6 key. You’ll see older and older commands. Eventually CMS begins re-displaying the same commands.

Send yourself another message by using PF6 to retrieve a TELL command you’d like to repeat. Then press ENTER. You don’t have to type over a letter of a command you’ve retrieved using PF6, but if you’d like to change the message, you can type over any part of it.

Use the Retrieve function to send yourself **two** more messages.



MESSAGE window. That's not exactly what we wanted. Press PF7 to scroll the CMS window back.

Using the PA2 Key to Scroll

In the CMS window, the PA2 key causes the top window to scroll forward. In this case, the MESSAGE window is the window that's on top.

Press the PA2 key once. Your screen should look something like this:

```
Fullcreen CMS                      Columns 1 - 79 of 81
Ready; T=0.03/0.07 10:26:50
tell * Don't force it, use a bigger hammer!
Ready; T=0.05/0.09 10:27:10
tell * Don't force it, use a bigger wrench!
Ready; T=0.04/0.06 10:27:20
tell * Don't force it, use a bigger hammer!
Ready; T=0.04/0.06 10:27:30
+-----+
|                Messages                | Lines 6 - 8 of 8 |
+-----+
| 10:32:33 MSG FROM SNOWHITE: Don't force it, call a bigger mechanic! |
| 10:32:35 MSG FROM SNOWHITE: Don't force it, Paul will probably panic! |
| 10:32:36 MSG FROM SNOWHITE: Don't force it, we're all a bit manic! |
+-----+
Ready; T=0.04/0.06 10:32:35
tell * Don't force it, Paul will probably panic!
Ready; T=0.04/0.06 10:32:36
PF1=Help      2=Pop_Msg   3=Quit       4=Clear_Top  5=Filelist   6=Retrieve
PF7=Backward  8=Forward   9=Rdrlist    10=Left     11=Right    12=Cmdline
====>
10:32:39 Message                               Enter a command or press a PF or PA key
```

Using the PA2 key is an easy way to scroll forward when you are working in the CMS window. By default, the PA2 key has different meanings when you are editing a file, or looking at a file list or reader list, so it cannot be used to scroll a window in those environments.

Keep pressing PA2 until you see only the last message. Press PA2 again. The window will disappear. When you scroll forward beyond the last line in the virtual screen, the window disappears.

It is important to remember that you can make some windows disappear by scrolling beyond the data. Which windows? Well, for our purposes, the MESSAGE, NETWORK, WARNING, and CMSOUT windows.

Popping a Window

When you are not using full-screen CMS, messages are gone forever once you clear them from your screen. This is not true in full-screen CMS.

Because CMS has recorded the messages in a virtual screen, you can still look at them through the MESSAGE window. All you need to do is make the MESSAGE window reappear. That is, you need to "pop" the window, which can be done by using the POP WINDOW command. Its format is:

```
POP WINDOW wname
```

For **wname**, you would type the name of the window you want to see. You can see data on any virtual screen by popping the window associated with it. In our case, we would type:

```
pop window message (Don't enter this command now.)
```

In the CMS window, the PF2 key is set to POP WINDOW MESSAGE. Now press the PF2 key. You'll see the MESSAGE window again:



```
Fullscreen CMS                               Lines 1 - 17 of 18
                                           Columns 1 - 79 of 81

Ready; T=0.03/0.07 10:26:50
tell * Don't force it, use a bigger hammer!
Ready; T=0.05/0.09 10:27:10
tell * Don't force it, use a bigger wrench!
Ready; T=0.04/0.06 10:27:20
tell * Don't force it, use a bigger hammer!
Ready; T=0.04/0.06 10:27:30

+-----+
+           Messages                       Lines 8 - 8 of 8
+-----+
+ 10:32:36 MSG FROM SNOWHITE: Don't force it, we're all a bit manic!
+-----+

Ready; T=0.05/0.07 10:32:33
tell * Don't force it, call a bigger mechanic!
Ready; T=0.04/0.06 10:32:35
tell * Don't force it, Paul will probably panic!
Ready; T=0.04/0.06 10:32:36

PF1=Help      2=Pop_Msg   3=Quit      4=Clear_Top  5=Filelist  6=Retrieve
PF7=Backward  8=Forward   9=Rdrlist  10=Left     11=Right   12=Cmdline
====>
10:33:38                                           Enter a command or press a PF or PA key
```

If you try to pop the MESSAGE, WARNING, NETWORK, or CMSOUT window, and the associated virtual screen doesn't have anything written on it, the window will not appear.

Using Border Commands

Border commands are commands that you type in any of the four corners of the border surrounding a window. The corners are represented by a plus sign (+). To use a border command, just type the command over one of the plus signs and press ENTER.

For example, to scroll our MESSAGE window backward, move the cursor over a + and type a "b" (for Backward):

```

                                Fullscreen CMS                                Lines 1 - 17 of 18
                                                                Columns 1 - 79 of 81
Ready; T=0.03/0.07 10:26:50
tell * Don't force it, use a bigger hammer!
Ready; T=0.05/0.09 10:27:10
tell * Don't force it, use a bigger wrench!
Ready; T=0.04/0.06 10:27:20
tell * Don't force it, use a bigger hammer!
Ready; T=0.04/0.06 10:27:30
b-----+
                                Messages                                Lines 8 - 8 of 8
+-----+
10:32:36 MSG FROM SNOWHITE: Don't force it, we're all a bit manic!
+-----+
Ready; T=0.05/0.07 10:32:33
tell * Don't force it, call a bigger mechanic!
Ready; T=0.04/0.06 10:32:35
tell * Don't force it, Paul will probably panic!
Ready; T=0.04/0.06 10:32:36

PF1=Help      2=Pop_Msg   3=Quit      4=Clear_Top  5=Filelist  6=Retrieve
PF7=Backward  8=Forward   9=Rdrlist  10=Left     11=Right   12=Cmdline
====>
10:33:38                                Enter a command or press a PF or PA key
```

Now press ENTER. The window will scroll backward. The border command you just used is called the B border command. There are other border commands, including:

- F** scrolls the window forward.
- R** scrolls the window to the right.
- L** scrolls the window to the left.
- D** drops the window.
- M** moves the window.
- P** pops the window.

Because the window scrolled right, you see only the rightmost characters of the line. Now type an "l" over one of the corners and press ENTER to scroll the window back to the left.

The D border command is another way to get rid of a window. Type a "d" in one of the corners of the window and press the ENTER key. The window will disappear just as it would have if you scrolled beyond the data.

What's the difference between the two techniques? When you use the D border command the window may not be positioned on the latest data the next time it is popped. When you scroll beyond the data, however, the window will be correctly positioned the next time it is popped. For this reason, it's preferable to get rid of a window by scrolling. The D border command is easier if you don't care which data the window shows the next time it appears. (You may have to scroll forward to see the latest data.)

Before using the M and P border commands, lets practice the border commands we've already learned.

First display a list of files on your screen. In the CMS window, the PF5 key is set to FILELIST. Press the PF5 key. You should see a list of files on your A-disk.

Now enter the following command on the large arrow command line:

```
q disk a
```

The result is displayed in a window. Your screen should look something like this:



Cursor

```

SNOWHITE FILELIST A0 V 108 Trunc=108 Size=39 Line=17 Col=1 Alt=0
Cmd  Filename Filetype Fm Format Lrecl  Records  Blocks  Date      Time
    100002 AUTOSAVE A1 V      27      22      1 6/27/86 11:31:34
    SF     SCREEN  A1 V      77      24      1 6/27/86  8:40:30
    TEST   EXEC    A1 V      11       6      1 6/25/86 14:45:43
    LONG3  LOG     A1 V      79      24      1 6/25/86 14:36:46
    LONG2  LOG     A1 V      79      24      1 6/19/86 11:59:37
+-----+-----+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+-----+-----+-----+
m |                                     Fullscreen CMS          Lines 23 - 24 of 24 |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     Messages                Lines 3 - 8 of 9 |BL
+-----+-----+-----+-----+-----+-----+-----+-----+
| 10:32:10 MSG FROM SNOWHITE: Don't force it, use a bigger hammer! |-- +
| 10:32:20 MSG FROM SNOWHITE: Don't force it, buy a bigger wrench! | 7:21
| 10:32:31 MSG FROM SNOWHITE: Don't force it, use a bigger wrench! | 2:26
| 10:32:33 MSG FROM SNOWHITE: Don't force it, call a bigger mechanic! | 3:20
| 10:32:35 MSG FROM SNOWHITE: Don't force it, Paul will probably panic | 1:07
| 10:32:36 MSG FROM SNOWHITE: Don't force it, we're all a bit manic! | 7:02
+-----+-----+-----+-----+-----+-----+-----+-----+
| 1= Help      2= Refresh  3= Quit    4= Sort(type)  5= Sort(date)  6= Sort(size)
| 7= Backward  8= Forward  9= FL /n 10=      11= XEDIT     12= Cursor
+-----+-----+-----+-----+-----+-----+-----+-----+
|=====>
|
|                                     X E D I T  1 File

```

Now press the ENTER key. The window moves to its new location:

```

S +-----+-----+-----+-----+-----+-----+-----+-----+
Cm |                                     Messages                Lines 3 - 8 of 9 | 34
+-----+-----+-----+-----+-----+-----+-----+-----+
| 10:32:10 MSG FROM SNOWHITE: Don't force it, use a bigger hammer! | 30
| 10:32:20 MSG FROM SNOWHITE: Don't force it, buy a bigger wrench! | 43
| 10:32:31 MSG FROM SNOWHITE: Don't force it, use a bigger wrench! | 46
| 10:32:33 MSG FROM SNOWHITE: Don't force it, call a bigger mechanic! | 37
| 10:32:35 MSG FROM SNOWHITE: Don't force it, Paul will probably panic | +
| 10:32:36 MSG FROM SNOWHITE: Don't force it, we're all a bit manic! | +
+-----+-----+-----+-----+-----+-----+-----+-----+
| LABEL  CUU M  STAT  CYL TYPE  BLKSIZE  FILES  BLKS USED-(%) BLKS LEFT  BL
| MB191  191 A  R/W   2 3380 2048    39    126-23    414
+-----+-----+-----+-----+-----+-----+-----+-----+
| VACATION CAMPING A1 F      80      1      1 4/18/86 9:07:21
| STOCK     INVEN  A1 F      80      7      1 4/18/86 9:02:26
| RADIO     WAVES  A1 F      80      3      1 4/18/86 8:53:20
| CRIME     THEFT  A1 F      80      4      1 4/18/86 8:51:07
| PICNIC    INGREDTS A1 F      80      3      1 4/18/86 8:47:02
| TEMP     SCRIPT  A1 V      4      21     1 3/06/86 16:04:38
+-----+-----+-----+-----+-----+-----+-----+-----+
| 1= Help      2= Refresh  3= Quit    4= Sort(type)  5= Sort(date)  6= Sort(size)
| 7= Backward  8= Forward  9= FL /n 10=      11= XEDIT     12= Cursor
+-----+-----+-----+-----+-----+-----+-----+-----+
|=====>
|
|                                     X E D I T  1 File

```

It's possible to move a window so that part of its border is either off the edge of the screen or covered by the other window. That's OK. You still have other corners in which you can enter border commands.

Try moving the windows a few more times. The only rule to follow is that the entire window must fit on the screen. If it doesn't, the window won't move and you'll see an error message in the CMSOUT window. Don't worry. Just move the window again.

When you are done practicing, move one window so it partially covers the other. Let's put the partially-covered window on top. To do this, we'll use the P border command to "pop" it.

Now type a "p" in any corner of the partially-covered window and press ENTER. Try a few moves to see the difference.

Practice using the M and P border commands. The best way to become comfortable with windows is by using them.

When you're done, get rid of both windows by using the F border command to scroll beyond the data. Then press the PF3 key to leave FILELIST.

WARNING and NETWORK Windows

Two windows we haven't yet worked with are the WARNING window and the NETWORK window. Handling those two windows is no different from handling the MESSAGE and CMSOUT windows. You can use the PA2 key in the CMS window to scroll the windows forward, and you can use all the border commands as well as the POP WINDOW command.

When you receive a warning, the terminal will "beep." If you're working in the CMS window, the "Warning" status notice will also



appear. The next time you press a key that gets the computer's attention, like a PF key or the ENTER key, the WARNING window automatically pops.

Unlike the other windows, the NETWORK window does not automatically pop. When you receive a network message, the terminal will "beep." If you're working in the CMS window, the "Network" status notice will also appear. The next time you press a key that gets the computer's attention, the notice disappears, but the window does *not* pop. To see the network messages, you must issue POP WINDOW NETWORK.

Using the WM Window

Earlier we said that the WM window was a kind of super window. That's because the WM window can be used to manage every other window that appears on your screen. You can make the WM window appear at any time by pressing the PA1 key. Let's see how to use the WM window.

From the CMS window, edit any file you have stored on your A-disk. Then issue this XEDIT command:

```
q pf
```

The CMSOUT window will appear. Because you moved the CMSOUT window in a previous exercise, it may not appear in the location shown below. The CMSOUT window will appear where it was when you last cleared it from the screen.

Now press the PA1 key. The WM window will appear at the bottom of your screen, like this:

```

STOCK      INVEN      A1  F 80  Trunc=80 Size=7 Line=0 Col=1 Alt=0

+-----+-----+
|                                     Fullscreen CMS          Lines 33 - 38 of 56
|                                     Columns 1 - 74 of 81
|
| PF1      BEFORE  HELP MENU
| PF2      BEFORE  SOS LINEADD
| PF3      BEFORE  QUIT
| PF4      BEFORE  TABKEY
| PF5      BEFORE  SCHCHANGE 6
| PF6      ONLY    ?
|-----+-----+
|==== 03469 GASKET          AA0306  9432
|==== 32554 RATCHET         AA0205  3762
|==== 32868 DOOHICKEY       BB0206  5248
|==== 64400 GIZMO           CC0101  9942
|==== 09263 WIDGET          AA0403  8645
|-----+-----+
| PF1=Help      2=Top      3=Quit      4=Clear      5=Copy      6=Retrieve
| PF7=Backward  8=Forward  9=Maximize 10=Left     11=Right   12=Restore
| Enter a windowing command or press PF key
|====>

```

WM Window

The WM window provides a set of PF keys that you can use to manipulate the top window, which is the CMSOUT window in our example.

Some of the PF keys are set to familiar functions. The PF7 and PF8 keys scroll the top window backward and forward. Try pressing the PF8 key. The CMSOUT window scrolls forward. Now press PF7 to scroll it back.

PF10 and PF11 scroll the top window left or right. Now press the PF11 key followed by the PF10 key. CMSOUT scrolls right, then left.

Another useful key is PF4, which clears the top window. When you clear the MESSAGE, CMSOUT, WARNING, or NETWORK window, it disappears from your screen. Press PF4 now. The CMSOUT window will disappear.



When you're done using the WM window, press the PF3 key to get rid of it. You should be looking at the file that you edited. Enter a QQ command to end the editing session and return to the CMS window.

Note that in some situations CMS automatically displays the WM window to help you manage other windows that are covering your screen.

Logging Your Messages and Warnings

Full-screen CMS automatically *logs* (or saves) all your messages and warnings in files called *log files*. You can edit or print these files just as you can edit or print any other file. The messages are logged in a file named MESSAGE LOGFILE, while the warnings are logged in WARNING LOGFILE. If you already have files with those names, CMS adds your messages and warnings to the end of them. Otherwise, CMS creates the files for you.

If you don't want your messages or warnings to be logged, enter the SET LOGFILE OFF command, which has this format:

```
SET LOGFILE vscreen OFF
```

For *vscreen*, type either MESSAGE or WARNING. To stop the logging of warnings, for example, you would enter:

```
set logfile warning off
```

To resume logging, use the SET LOGFILE ON command, which has this format:

```
SET LOGFILE vscreen ON
```

For *vscreen*, type MESSAGE or WARNING.

You should occasionally edit your log files to delete messages and warnings that you no longer need to keep. Otherwise, the size of the file will continue to grow and consume space on your A-disk.

The SET LOGFILE vscreen OFF command is good only until you log off. The next time you log on and start full-screen CMS, logging of both messages and warnings will resume automatically.

Network messages are not automatically logged.

Using the CMS PF Keys

In the CMS window, the PF keys are set to frequently-used commands. Instead of typing the command, you can press the PF key. The settings of the CMS PF keys are summarized below.

PF1 Help

When you press the PF1 key, the main help menu is displayed.

PF2 Pop_Msg

This key causes the MESSAGE window to be popped. The MESSAGE window will not appear if you haven't yet received any messages.

PF3 Quit

This key suspends the operation of full-screen CMS. It does not issue the LOGOFF command.

To resume using full-screen CMS, enter the command SET FULLSCREEN ON.

**PF4 Clear_Top**

This PF key clears the top window. Clearing the MESSAGE, NETWORK, WARNING, or CMSOUT window will make it disappear from your screen.

PF5 Filelist

When you press the PF5 key, a list of all the files on your A-disk is displayed. Pressing PF5 is the same as entering a FILELIST command.

PF6 Retrieve

When you press PF6, the last command you entered is displayed on the CMS command line. To execute the command, just press the ENTER key.

PF7 Backward

When you press the PF7 key, the CMS window scrolls backward one window.

PF8 Forward

When you press the PF8 key, the CMS window scrolls forward one window.

PF9 Rdrlist

Pressing the PF9 key is the same as entering the RDRLIST command. A list of all the files in your reader is displayed.

PF10 Left

This key causes the CMS window to scroll left.

PF11 Right

This key causes the CMS window to scroll right.

PF12 Cmdline

Pressing this key causes the cursor to move from its position on the CMS window to the large arrow command line.

If you have a terminal with 24 PF keys, PF keys 13 through 24 have the same settings as 1 through 12.

Setting a CMS PF Key to TELL

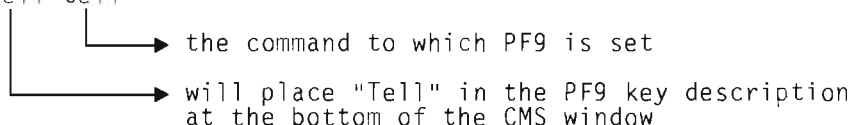
You can change the setting of any PF key in the CMS window by using the SET CMSPF command. The SET CMSPF command has this format:

```
SET CMSPF nn synonym command
```

For *nn*, type the number of the PF key you want to set. Substitute a description of the command for *synonym*. The description must be nine letters or fewer. (VM updates the PF key description at the bottom of the window with whatever you type for *synonym*.) For *command*, type the command you want assigned to that PF key.

Let's set the PF9 key to the TELL command. Then, whenever you want to send a message, you can press PF9. Type the command:

```
set cmspf 9 Tell tell
```



the command to which PF9 is set

will place "Tell" in the PF9 key description at the bottom of the CMS window

and press the ENTER key.

The PF key descriptions at the bottom of the CMS window should show PF9 set to "Tell." Now press the PF9 key. TELL will appear in



the command line. Type a nickname and a message. Then press ENTER to send the message.

This command lasts only until you log off. The next time you log on and use full-screen CMS, you'll have to enter the command again.

Leaving Full-Screen CMS

To leave full-screen CMS, press the PF3 key while viewing the CMS window. To resume using full-screen CMS, enter the command SET FULLSCREEN ON.

Summary

You learned the following commands in this chapter:

Command	Function
SET FULLSCREEN ON	Starts full-screen CMS.
PA1 key	Pops the WM window.
PA2 key in the CMS window	Scrolls a window forward.
F border command	Scrolls a window forward.
B border command	Scrolls a window backward.
R border command	Scrolls a window to the right.
L border command	Scrolls a window to the left.

Command	Function
D border command	Drops a window.
M border command	Moves a window.
P border command	Pops a window.
POP WINDOW wname	Pops a window.
SET LOGFILE vscreen OFF	Stops logging for a virtual screen.
SET LOGFILE vscreen ON	Starts logging for a virtual screen.
SET CMSPF nn synonym command	Sets a PF key in the CMS window.

Summary of Changes

To obtain editions of this publication that pertain to earlier releases of VM/SP, you must order using the pseudo-number assigned to the respective edition. For:

Release 4, order ST00-1591

Release 3, order SQ24-5236

Release 2, order ST24-5236

Summary of Changes for SC24-5236-3 VM/SP Release 5

- Chapter 1 is changed to show logging on from the VM/SP logo.
- A description of online HELP has been added to Chapter 6.
- Most messages from VM are now in mixed-case.
- Chapter 10, "Looking at VM/SP Through Windows" has been added.
- This edition also includes minor editorial and technical changes.

Summary of Changes for SC24-5236-2 VM/SP Release 4

- Chapter 9, "Creating and Running Programs," has been added.

- A description of online HELP has been added to the Summary of Commands (card).
- In Chapter 2, use of the XEDIT task menu, an addition to the XEDIT HELP menu, is explained.
- Additional copies of the summary card (inside back cover) can now be ordered separately as *VM/System Product CMS Primer Summary of Commands*, SX24-5151.
- This edition also includes minor editorial and technical changes.

**Summary of Changes
for SC24-5236-1
VM/SP Release 3**

- **PF Key Changes**

Initial settings of various PF keys have changed for XEDIT and for screens created by the HELP, FILELIST, NAMES, NOTE, SENDFILE, PEEK, and RDRLIST commands. Most of the PF key changes affect the settings of PF keys 10, 11, and 12.

XEDIT PF Key Changes

In XEDIT, the functions of PF10 (SPLIT) and PF11 (JOIN) are now combined on a single key, the PF11 key. The PF12 key now moves the cursor to the command line or back to its previous location on the screen. (PF10 allows you to see data to the right of the screen; however, this function is not discussed in this book.)

PF Key Changes On Other Screens

Various PF key settings for the following screens have been changed: HELP, FILELIST, NAMES, NOTE, SENDFILE, PEEK, and

RDRLIST. The new settings are displayed at the bottom of the screens.

On the FILELIST and RDRLIST screens, commands typed in the **Cmd** space on the screens are now executed by pressing the ENTER key instead of the PF10 (EXECUTE) key. The PF10 key has no function. Similarly, on the SENDFILE screen, the ENTER key replaces the function previously assigned to PF10 (EXECUTE).

- **Screen Layout Changes**

XEDIT Screen

The command line arrow has an additional equal sign (====>).

An alteration count is displayed on the file identification line as ALT = n.

The COPY/MOVE PENDING and BLOCK INCOMPLETE notices no longer appear in the status area when a prefix command is pending or incomplete. Instead, the notice '**command pending...**' is displayed.

HELP, FILELIST, NAMES, NOTE, SENDFILE, PEEK, and RDRLIST Screens

The command line arrow has an additional equal sign (====>).

On the FILELIST, NOTE, PEEK, and RDRLIST screens, an alteration count is displayed on the file identification line as ALT = n.

The message line appears at the bottom of the screen, above the command line, instead of at the top of the screen.

- Chapter 8, "Creating and Using Execs," now deals with the **System Product Interpreter** instead of EXEC 2.
- This edition also includes minor editorial and technical changes.

Bibliography

This primer presents only some of the functions and commands available with the VM/SP system. At the beginning of each chapter, you will find the names of the IBM publications that contain additional information on the material presented in that chapter. These publications and their order numbers are summarized below:

Virtual Machine/System Product: CMS Command Reference, SC19-6209

Virtual Machine/System Product: CMS Macros and Functions Reference, SC24-5284

Virtual Machine/System Product: CMS User's Guide, SC19-6210

Virtual Machine/System Product: CP Command Reference, SC19-6211

Virtual Machine/System Product: System Product Editor User's Guide, SC24-5220

Virtual Machine/System Product: System Product Editor Command and Macro Reference, SC24-5221

Virtual Machine/System Product: System Product Interpreter User's Guide, SC24-5238

Virtual Machine/System Product: System Product Interpreter Reference, SC24-5239

*Document Composition Facility: Generalized Markup Language
Starter Set User's Guide, SH20-9186*

*Virtual Machine/System Product: Application Development
Guide, SC24-5247.*

Index

Special Characters

- / prefix command 47
- = command
 - using in XEDIT 72
 - using on FILELIST screen 116
- " prefix command 54

A

- A prefix command 42
- A-disk 106, 122, 125
- A-disk space, displaying 182
- abbreviation of commands 19
- ACCESS command 125
- acknowledgement 164
- adding data from another file 75
- adding lines to a file 42
- address of disk, displaying 182
- ALL NOTEBOOK file 144, 159, 175
- ALT = n, meaning of 22
- alteration count 22
- automatically saving your file 91
- autosave 91

B

- B border command 248
- back tab key 5
- backing-up a file 119
- backward locate 70
- backward scrolling in XEDIT 44
- BASIC 227
- BASICRUN command 230
- BEGIN command 14
- beginning to use XEDIT 17
- bold type 203
- border commands
 - B (backward) 248
 - D (drop) 248, 250
 - F (forward) 248
 - L (left) 248, 250
 - M (move) 248, 252
 - P (pop) 248, 254
 - R (right) 248, 249
- borrowing another user's file 122
- BOTTOM command 44

C

- C prefix command 55
- capital letters, typing 5
- CASE option in SET 28
- case, setting 28
- CC option of the NOTE command 158
- CC prefix command 56
- CHANGE command 61
- changes, getting rid of 78
- changing a file 40

- changing the name of a file 117
- checking your spelling 205
- clearing a window 256, 259
- clearing the prefix area 58
- clock, on the CMS window 237
- CMS (Conversational Monitor System) 12
- CMS command line 27, 237
- CMS PF keys 237, 258
- CMS PF keys, setting 260
- CMS window 235, 239
- CMSOUT window 240
- COBOL 220
- column location information 236
- command abbreviation 19
- command formats, interpreting 19
- command input area of the CMS window 236
- command line
 - in the CMS window 237
 - large arrow 22
- command line in CMS 27
- commands
 - how to enter 19
 - rerunning 242
 - retrieving 243
 - rules for entering 22
- communicating 137
- compact simple lists 196
- COMPARE command 120
- comparing two files 120
- compiling a program 222, 229
- CONTINUE option of SCRIPT command 194
- Control Program (CP) 14
- Conversational Monitor System (CMS) 12
- COPYFILE command 119
- copying a file 119
- copying lines 55
- correcting typing mistakes 6
- CP (Control Program) 14
- CP READ status notice 14
- creating a file
 - from a FILELIST screen 114
 - using XEDIT command 20
- creating a program 220, 227
- creating an XEDIT profile 101

- current line
 - in file identification line 22
 - position on screen 23
- cursor
 - explanation of 3
 - movement keys for 3

D

- D border command 248, 250
- D prefix command 51
- date displayed 181
- day of week displayed 181
- DCF (Document Composition Facility) 187
- DD prefix command 53
- default tabs, displaying 32
- DEFINE STORAGE command 222
- definition list 201
- delete key 7
- deleted lines, recovering 54
- deleting a block of lines 53
- deleting lines from a file 51
- deleting the rest of the line 7
- destroying a file 115
- destroying a reader file 175
- digital clock on the CMS window 237
- directing messages to a file 257
- directing warnings to a file 257
- DISCARD command 115, 175
- discarding a file 115
- discarding a reader file 175
- disk address, displaying 182
- disk label, displaying 182
- disk numbers 122
- disk space, displaying amount used 182
- disk, storing files on 27
- displaying a document 194
- displaying a file on your screen 40
- displaying a reader file 176
- displaying list of files 106
- displaying PF key settings 35

- displaying tab settings 32
- Document Composition Facility (DCF) 187
- DOWN command 46
- dropping a window 248, 250
- dropping the WM window 257
- duplicating a file 119
- duplicating lines 54

E

- editing files 17
- editing multiple files 98
- editing session 20
- editor defined 17
- eliminating a reader file 175
- emphasizing a word or phrase 203
- ending editing session
 - using FILE 27
 - using QQUIT 78
 - using QUIT 78
- Enter a command or press a PF or PA key (status notice) 238
- ENTER key 22, 27
- Enter your response (status notice) 238
- entering commands, rules for 19, 22
- entering data in XEDIT 23
- equal (=) command
 - using in XEDIT 72
 - using on FILELIST screen 116
- ERASE EOF key 7
- erased lines, recovering 54
- erasing a file 115
- erasing a reader file 175
- erasing the rest of the line 7
- EXECs
 - definition of 209
 - example of 211, 214, 217
 - PROFILE 218
 - quotation marks within 216
 - running 211

- Executing a command (status notice) 238
- executing commands again 242

F

- F border command 248
- F prefix command 55
- fields for sorting 82
- file area in XEDIT 23
- FILE command 27
- FILE command (in FILELIST) 166
- file identification line in XEDIT 21
- file label 18
- FILELIST command 40, 106
- FILELIST command used to save a list of files 166
- FILELIST filetype 166
- filemode 18, 182
- filename 18
- filename, changing 117
- filenames of AUTOSAVE files 92
- files
 - adding lines to 42
 - changing 40
 - comparing 120
 - copying 119
 - creating 20
 - definition of 18
 - deleting lines from 51
 - displaying 40, 106
 - erasing 115
 - labels for 18
 - listing 40, 106
 - message log file 257
 - protecting 90
 - receiving 171, 173
 - renaming 117, 174
 - replacing 174
 - saving 90
 - saving a list of 166

- sending 162
- sending more than one 165
- sharing 122
- sorting 81
- storing of 27
- warning log file 257

filetype 18

finding text in a file 61

fixing typing mistakes 6

formats of commands, interpreting 19

FORTTRAN 221

FORTVTS command 223

forward scrolling in XEDIT 44

full-screen CMS

- starting 235
- suspending 261
- using 233

G

Generalized Markup Language (GML) 187

GENMOD command 226

GET command 75, 77

getting data from another file 75

getting help 87, 183

getting rid of the WM window 257

getting rid of your changes 78

GLOBAL command 225

GML (Generalized Markup Language) 187

H

heading tags 190

HELP command 87, 183

highlight phrase tags 203

highlighting 203

HOLDING (terminal status) 15

how to enter commands 19

I

IDENTIFY command 181

initial tabs, displaying 32

INPUT command 23

input inhibited light 6

input mode

- entering using INPUT 23
- getting out of 25, 26
- screen layout in 24

input zone 24

insert mode key in XEDIT 36

insert mode, leaving 36

inserting data from another file 75

inserting words in a line 36

interactive mode of BASIC 227

interpreting command formats 19

introduction to XEDIT 17

invoking a compiler 222, 229

invoking a program 225, 230

invoking an EXEC 211

IPL CMS command 12

italic type 203

J

join a line at cursor 30

K

keeping messages in a file 257

keeping warnings in a file 257

keyboard layout 3

keys

- back tab 5

- cursor movement 3
- ERASE EOF 7
- lock 5
- new-line 5
- shift 5
- tab 5
- to delete characters 7

L

- L border command 248, 250
- label of AUTOSAVE files 92
- learning to use XEDIT 17
- leaving full-screen CMS 261
- leaving insert mode 36
- libraries needed for programs 225
- line location information 236
- line numbers, setting on 80
- lines
 - adding 42
 - copying 55
 - deleting 51, 53
 - duplicating 54
 - inserting words into 36
 - moving 57
 - recovering 54
- LINK command 124
- linking 124
- list
 - definition 201
 - ordered 197
 - simple 196
 - tags for 198, 201
 - unordered 197
- list of files, saving 166
- LISTING filetype 224
- listing your files 40, 106
- LOAD command 225
- LOCATE command
 - using in FILELIST 110

- using in XEDIT 61, 70
- location information 236
- lock key 5
- log files 257
- logging of messages 257
- logging of warnings 257
- logging off 13
- logging on 9
- logo 2, 9
- LOGOFF command 13
- LOGON command 11
- logon logo 2, 9
- LONG option of the NOTE command 155
- looking at a reader file 176
- looking at another user's files 122
- looking at messages and warnings again 257
- looking at VM/SP through windows 233

M

- M border command 248, 252
- M prefix command 57
- markup 187
- menu for HELP 184
- message line 22
- MESSAGE LOGFILE 257
- Message status notice 237
- MESSAGE virtual screen 239
- MESSAGE window 239
- messages
 - definition of 150
 - log file for 257
 - network 240, 255
 - reviewing 257
 - sending 150
 - status 14
 - turning off 151
 - warning 240, 254
- messages, status 238
- mistakes, correcting 6

- mixed case 28
- MIXED option in SET CASE 28
- MM prefix command 57
- modules, generating 226
- MORE status notice 15
- moving lines 57
- moving though a file 44
- moving windows 252
- multiple files, editing 98
- multiple views of files 98, 100

N

- name of file, changing 117
- NAMES command 139
- names file
 - changing an entry within 146
 - creating 139
 - definition of 139
 - deleting an entry from 147
 - finding information in 147
- names, displaying 181
- NETLOG filetype 164
- network messages 240, 255
- Network status notice 238
- NETWORK virtual screen 240
- NETWORK window 240, 255
- new-line key 5
- nicknames 141
- nicknames used in SENDFILE 163
- nicknames used on the NOTE command 153
- node displayed 181
- NOT ACCEPTED status notice 15
- NOTE command 153
- NOTEBOOK file 144
- notes
 - continuing 159
 - creating 153
 - definition of 152
 - keeping copies of 159
 - receiving 171, 175

- sending 156
- sending complimentary copies of 158
- to a list of people 156
- typing 155
- notices, status 14, 238
- numbering of disks 122
- numbers in prefix area, setting on 80

O

- obliterating a file 115
- online information 183
- ordered list 197

P

- P border command 248, 254
- P prefix command 55
- paragraph tag 190
- PA1 key 255
- PA2 key in the CMS window 245
- PA2 key in XEDIT 36
- peeking at a reader file 176
- pending notice 58
- people using computer, displaying 181
- permanent storage 27
- PF keys
 - displaying XEDIT settings of 35
 - in the CMS window 237, 258
 - on FILELIST screen 170
 - on the NOTE screen 160
 - on the PEEK screen 176
- PF1 key
 - in NAMES 148
 - in NOTES 160
 - in the CMS window 258
 - in XEDIT 87
- PF10 key

- in NAMES 149
- in NOTES 161
- in the CMS window 259
- in the WM window 256
- PF11 key
 - in FILELIST 41, 114
 - in NAMES 149
 - in NOTES 161
 - in RDRLIST 176
 - in the CMS window 259
 - in the WM window 256
 - in XEDIT 30, 31
- PF12 key
 - in NAMES 149
 - in NOTES 161
 - in the CMS window 260
 - in XEDIT 45
- PF2 key
 - in FILELIST 114
 - in NAMES 142, 148
 - in NOTES 160
 - in the CMS window 246, 258
- PF3 key
 - in NAMES 142, 148
 - in NOTES 160
 - in the CMS window 258
 - in the WM window 257
 - in XEDIT 78
- PF4 key
 - in FILELIST 109
 - in HELP 185
 - in NAMES 142, 148
 - in NOTES 160
 - in the CMS window 259
 - in the WM window 256
 - in XEDIT 32
- PF5 key
 - in FILELIST 109
 - in NAMES 148
 - in NOTES 156, 160
 - in the CMS window 250, 259
- PF6 key
 - in FILELIST 109
 - in NAMES 147, 148
 - in NOTES 160
 - in the CMS window 243, 259
- PF7 key
 - in NAMES 147, 148
 - in NOTES 160
 - in the CMS window 259
 - in the WM window 256
 - in XEDIT 44
- PF8 key
 - in NAMES 147, 148
 - in NOTES 156, 160
 - in the CMS window 259
 - in the WM window 256
 - in XEDIT 44
- PF9 key
 - in NAMES 149
 - in NOTES 160
 - in RDRLIST 173
 - in the CMS window 259
 - in XEDIT 72
- phrase highlighting tags 203
- PL/I 220
- POP WINDOW command 246
- popping a window 246, 254
- positioning windows 252
- POWER command 95
- power typing 95
- prefix area in XEDIT 22
- prefix area numbers, setting on 80
- prefix commands
 - / command 47
 - " command 54
 - A command 42
 - C command 55
 - CC command 56
 - D command 51
 - DD command 53
 - F command 55
 - M command 57
 - MM command 57

- P command 55
 - resetting 58
 - where to type 42
- premature EOF defined 121
- PRINT command 130
- PRINT option of SCRIPT 194
- printing a file 130
- PROFILE EXEC file 218
- profile for XEDIT, creating 101
- PROFILE option of SCRIPT command 194
- programs
 - compiling 222, 229
 - creating 220, 227
 - definition of 219
 - libraries needed for 225
 - running 225, 230
- protecting your files 90
- PULL (REXX instruction) 212
- putting messages in a file 257
- putting warnings in a file 257

Q

- QUIT command 78
- QUERY AUTOSAVE command 92
- QUERY command
 - DISK option 182
 - NAMES option 181
 - SEARCH option 123
 - STORAGE option 222
 - USERS option 181
- QUERY PF command 35
- QUERY TABS command 32
- QUIT command 78
- quotation marks in EXECs 216

R

- R border command 248, 249
- R/O disks 182
- R/W disks 182
- RDRLIST command 172
- read-only disks, linking to 124
- reader
 - description of 171
 - discarding a file from 175
 - displaying information about 172
 - peeking at 176
 - receiving a file from 173
 - receiving a note from 175
- ready message 12, 27
- receiving files 171
- receiving notes 171, 175
- recording messages in a file 257
- recording warnings in a file 257
- RECOVER command 54
- recovering deleted lines 54
- refreshing a FILELIST screen 114
- RELEASE command 126
- releasing another person's disk 126
- RENAME command 117
- renaming a file 117, 174
- repeating XEDIT commands 72
- replacing a file 174
- rerunning commands 242
- RESET command 58
- RESET key 6
- resetting the prefix area 58
- Restructured Extended Executor (REXX)
 - language 210
- resuming full-screen CMS 261
- Retrieve function 243
- retrieving commands 243
- reverse locate 70
- reverse tab key 5
- reviewing messages and warnings 257
- REXX (Restructured Extended Executor)
 - language 210

running a compiler 222, 229
running a program 225, 230
running an EXEC 211
running commands again 242

S

SAVE command 90
SAVE command in BASIC 229
saving a BASIC program 229
saving a list of files 166
saving messages in a file 257
saving warnings in a file 257
SAY (REXX instruction) 212
scale 23
screen in XEDIT, splitting 98
screen layout
 in FILELIST 107
 in NAMES 141
 in NOTES 154
 in XEDIT edit mode 21
 in XEDIT input mode 24
 of RDRLIST 172
 of the CMS window 236
Script 187
SCRIPT command 133, 194
SCRIPT command options
 CONTINUE 133, 194
 PRINT 133, 194
 PROFILE 133, 194
 SPELLCHK 205
SCRIPT/VS 187
Scroll forward for more information (status notice) 239
scroll location information 236
scrolling
 in FILELIST 109
 through a file using XEDIT 44
scrolling a window 245, 248, 256
SENDFILE command 162
sending file(s) 162
sending messages 150
sending more than one file 165
SET AUTOSAVE command 91
SET CASE command 28
SET CMSPF command 260
SET FULLSCREEN ON command 235
SET LOGFILE command 257
SET MSG command 151
SET NUMBER ON command 80
SET PF command 102
SET SCREEN command 98
SET TABS command 32
setting CMS PF keys 260
setting tabs 32
settings of PF keys, displaying 35
sharing files 122
shift keys 5
signing off 13
signing on 9
simple list 196
SORT command 83
sort fields 82
sorting a file 81
space bar 6
space remaining, displaying 182
SPELLCHK option of SCRIPT command 205
spelling check 205
split a line at cursor 30
splitting the screen 98
SPOOL command 130, 133
starter set of GML 188
starting a program 225, 230
starting an EXEC 211
starting full-screen CMS 235
starting to use XEDIT 17
status area in XEDIT 22
status line 237
status notices 14, 238
STATUS virtual screen 240
STATUS window 240
stopping full-screen CMS 261

storage, permanent 27
storage, virtual 26
summary of changes 263
suspending full-screen CMS 261
syntax of commands, interpreting 19

T

tab key 5
tabs, setting 32
tags for lists 198, 201
tags, GML (Generalized Markup Language) 188
TELL command 150
terminal, using 2
TEXT filetype 224
throwing away a file 115
time displayed 181
title line of the CMS window 236
TOP command 45
TRUNC=n, meaning of 21
truncation column 21
turning off messages 151
turning on full-screen CMS 235
typamatic, definition of 4
typing capital letters 5
typing mistakes, correcting 6
typing over commands 242

U

underlining 203
undoing your changes 78
unlocking your keyboard 6
unordered list 197
UP command 46
uppercase letters, typing 5
uppercase, data entered in 25
userid displayed 181

users, displaying 181
using a compiler 222, 229
using an EXEC 211
using full-screen CMS 233
using IBM BASIC 227
using other people's files 122

V

variables used in an EXEC 212
video display terminal 1, 2
Virtual Machine/System Product (VM/SP) 1
virtual screens
 CMS virtual screen 239
 definition of 234
 MESSAGE virtual screen 239
 NETWORK virtual screen 240
 STATUS virtual screen 240
 WARNING virtual screen 240
 WM virtual screen 240
virtual storage 26, 222
VM READ status notice 14
VM/SP (Virtual Machine/System Product) 1
VS FORTRAN 221
vscreen (virtual screen) 234

W

WARNING LOGFILE 257
warning messages 240, 254
Warning status notice 238
WARNING virtual screen 240
WARNING window 240, 254
warnings
 log file for 257
 reviewing 257
windows
 clearing 256, 259

CMS window 235, 239
CMSOUT window 240
definition of 234
dropping 248, 250
making them appear 246, 254
MESSAGE window 239
moving 248, 252
NETWORK window 240, 255
popping 246, 248, 254
positioning 252
scrolling 244, 245, 248, 256
STATUS window 240
WARNING window 240, 254
WM window 240, 255
WM virtual screen 240
WM window 240, 255
writing a file on disk 27

X

XEDIT command 20
XEDIT commands, repeating 72
XEDIT file identification line 21
XEDIT profile, creating 101
XEDIT, introduction to 17

Z

zone of input 24

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

File No. S370/4300-39
Printed in U.S.A.

SC24-5236-3

IBM
®

VM/SP CMS Primer
Order No. SC24-5236-3

**READER'S
COMMENT
FORM**

This manual is part of a library that serves as a reference source for systems analysts, programmers, and operators of IBM systems. You may use this form to communicate your comments about this publication, its organization, or subject matter, with the understanding that IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you.

Your comments will be sent to the author's department for whatever review and action, if any, are deemed appropriate. Comments may be written in your own language; English is not required.

***Note:** Copies of IBM publications are not stocked at the location to which this form is addressed. Please direct any requests for copies of publications, or for assistance in using your IBM system, to your IBM representative or to the IBM branch office serving your locality.*

Your comments:

<p>Note: Staples can cause problems with automated mail sorting equipment. Please use pressure sensitive or other gummed tape to seal this form.</p>
--

If you would like a reply, please supply your name and address on the reverse side of this form.

Thank you for your cooperation. No postage stamp necessary if mailed in the U.S.A. (Elsewhere, an IBM office or representative will be happy to forward your comments or you may mail directly to the address in the Edition Notice on the back of the title page.)

SC24-5236-3

If you would like a reply, please print:

Your Name _____
Company Name _____ Department _____
Street Address _____
City _____
State _____ Zip Code _____
IBM Branch Office serving you _____

Fold and tape

Please Do Not Staple

Fold and tape

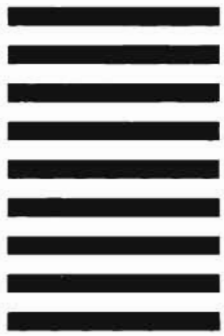


BUSINESS REPLY MAIL

FIRST-CLASS MAIL PERMIT NO. 40 ARMONK, NY

POSTAGE WILL BE PAID BY ADDRESSEE:

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES



INTERNATIONAL BUSINESS MACHINES CORPORATION
DEPARTMENT G60
PO BOX 6
ENDICOTT NY 13760-9987

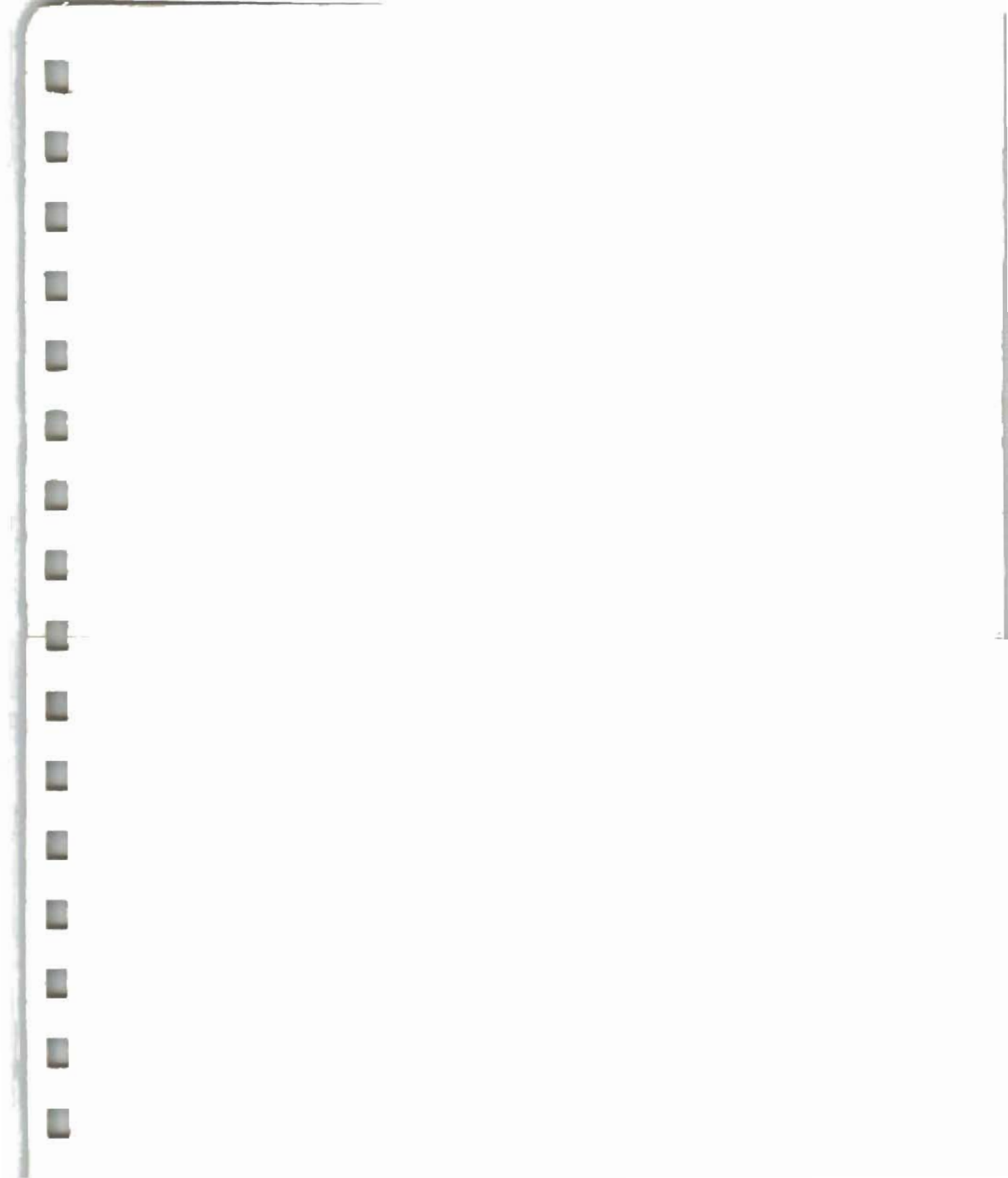


Fold and tape

Please Do Not Staple

Fold and tape





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

File No. S370/4300-39
Printed in U.S.A.

SC24-5236-3

SC24-5236-03

IBM
®

