

The Operating System Handbook

or, Fake Your Way Through Minis and Mainframes

by Bob DuCharme

VM/CMS

Table of Contents

Chapter 17 VM/CMS: An Introduction.....	
17.1 History.....	1
17.1.1 CP: The Control Program.....	2
Chapter 18 Getting Started with VM/CMS.....	
18.1 Starting Up.....	4
18.1.1 The Logon Screen.....	4
18.1.2 Entering CMS.....	6
18.1.3 Entering Commands.....	7
18.1.3.1 MORE	9
18.1.3.2 Aborting Screen Output.....	10
18.1.3.3 Case Sensitivity.....	12
18.1.4 Finishing Your CMS Session.....	12
18.2 File Names.....	12
18.2.1 Wildcards.....	13
18.2.1.1 The Asterisk.....	14
18.2.1.2 The Percent Sign.....	15
18.3 How Files Are Organized.....	15
18.3.1 Free Space on Your Disk.....	16
18.4 Available On-line Help.....	16
18.4.1 Help Function Keys.....	18
18.4.2 Help Menus.....	19
18.4.3 Command-Line Help.....	23
18.4.3.1 Command-Line Help and Error Messages.....	28
Chapter 19 Using Files in VM/CMS.....	
19.1 The Five Most Important Commands.....	29
19.1.1 Command Options.....	29
19.1.2 Common Error Messages.....	29
19.1.3 Listing File Names.....	31
19.1.3.1 Listing More than File Names.....	33
19.1.4 Displaying a Text File's Contents.....	35
19.1.5 Copying Files.....	35
19.1.5.1 Changing a File's Format.....	37
19.1.6 Renaming Files.....	38
19.1.7 Deleting Files.....	39
19.2 Sharing Files between Users.....	39
19.2.1 Unlinking the Disk.....	42
19.2.2 Other Ways to Link.....	42
Chapter 20 The VM/CMS XEDIT Text Editor.....	
20.1 Entering XEDIT.....	44
20.2 Customizing Your XEDIT Environment.....	45
20.3 Prefix Commands.....	47
20.3.1 Adding New Lines.....	47

20.3.2 Moving Your Cursor Around.....	48
20.3.3 Inserting, Deleting, and Typing over Words and Characters.....	49
20.3.4 Duplicating Lines.....	49
20.3.5 Deleting Lines.....	51
20.3.6 Copying Lines.....	52
20.3.7 Moving Lines.....	54
20.4 Searching for Text.....	54
20.5 Saving Your Changes.....	56
20.6 Quitting XEDIT.....	56
20.7 Other Useful XEDIT Features.....	56
20.7.1 XEDIT On-line Help.....	56
20.7.2 The Automatic Startup Macro.....	57
20.7.3 The Split/Join Key.....	58
Chapter 21 Using a VM/CMS System.....	
21.1 Printing Text Files.....	60
21.1.1 Checking the Print Queue.....	60
21.1.2 Cancelling Your Print Job.....	60
21.2 Command Files.....	61
21.2.1 The Automatic Logon Command File.....	62
21.3 Communicating with Other Users.....	62
21.3.1 Sending Files.....	64
21.3.2 Receiving Mail and Files.....	65
21.3.2.1 Looking at a File in the Reader.....	66
21.3.2.2 Saving a File from the Reader.....	67
21.3.2.3 Discard a File from the Reader.....	69
21.4 The FILELIST Program.....	69
21.4.1 Copying Files.....	70
21.4.2 Renaming Files.....	71
21.4.3 Deleting Files.....	71
21.4.4 Displaying A Text File's Contents.....	71
21.4.5 Editing a File.....	72
21.4.6 Printing a File.....	72
21.5 A Sample VM/CMS Session.....	72

This Section, the Rest of the Book

This is one part of the book "Operating Systems Handbook (or, Fake Your Way Through Minis and Mainframes)," which was originally published by McGraw-Hill as a \$49.60 hardcover. Once they reverted the rights to me after it went out of print, I converted the original XyWrite files to DocBook XML and then used Norm Walsh's stylesheets (see www.nwalsh.com) and the Apache FOP program (see xml.apache.org) to convert that to Acrobat files. The six parts of the book are all available at <http://www.snee.com/bob/opsys.html>:

- Part 1: Introduction. Note that this part includes a new section explaining why I didn't update the book. I strongly suggest that, no matter how much or how little of the book you can use, you glance through the whole Introduction as well. The "Comments and Suggestions" part is now obsolete; my home page is now <http://www.snee.com/bob> and my e-mail address is bob@snee.com.
- Part 2: UNIX. Everything described here should apply to Linux and its relatives.
- Part 3: OpenVMS. Basically, VMS. DEC was calling it "OpenVMS" at the time.
- Part 4: OS/400. The operating system for IBM's AS/400.
- Part 5: VM/CMS. An IBM mainframe operating system.
- Part 6: MVS. Another IBM mainframe operating system.

See www.snee.com/bob for information on books I've written since. In reverse chronological order, they are:

- *XSLT Quickly* is a tutorial and users guide to XSLT designed to get you writing stylesheets as quickly as possible.
- *XML: The Annotated Specification* is a copy of the official W3C XML specification with examples, terminology, and explanations of any SGML and computer science concepts necessary for a complete understanding of the XML spec.
- *SGML CD* is a users guide to free SGML software, most of which can be used with XML as well. The chapter on Emacs and PSGML, which requires no previous knowledge of Emacs, is available on the web site in English, Russian, and Polish.

One more thing: either I couldn't figure out the XSL `keep-together` property or version 0.18.1 of FOP doesn't implement it yet. Either way I apologize that some screen shots get split across page breaks.

Entire book copyright 2001 Bob DuCharme all rights reserved

Chapter 17 VM/CMS: An Introduction

Before we begin with VM/CMS, you must understand one crucial concept: the operating system you are using. This may seem obvious; after all, the name of this part of the book is "VM/CMS." But, as you often hear people refer to the operating system as VM and as CMS, learning the relationship between the two is the first step in understanding it.

17.1 History

IBM developed VM ("Virtual Machine") in 1964. Like any operating system, VM controlled the computer's resources. It also added a feature that had never existed before: the illusion, for each of its users, that they had a whole computer to themselves. (Because IBM developed VM well before the invention of the PC, having even a simulation of your own computer was a Big Deal.) If twenty people use a VM system at once, it gives them the illusion that they are using twenty different computers.

The most interesting part is that each of these simulated "computers" can run its own operating system. IBM has developed versions of MVS, UNIX, DOS/VSE, and even PC/DOS to run under VM (we call an operating system that runs one of VM's simulated computers a "guest" operating system running "under" VM), but the most popular is the one that was designed from the ground up to take the fullest possible advantage of the VM environment: CMS. (VM actually evolved from an earlier version of CMS called CTSS, but this all happened at IBM's Cambridge Scientific Center in Massachusetts, before either was released as a commercial product. The M.I.T. engineers who designed CTSS went on to design MULTICS, an ancestor of UNIX, which makes CMS a cousin of UNIX!)

Because the vast majority of people who use VM run CMS as the guest operating system, most VM user IDs are set up to automatically run CMS when someone logs on to their VM ID. Many users doesn't realize that this extra step is taking place and assume that it's all part of the normal process of logging on to the "VM/CMS" operating system. Think of it this way: once this logon process is complete, you interact with the CMS operating system, but behind the scenes, the VM operating system runs the show.

CMS originally stood for "Cambridge Monitor System," but IBM decided that its hundreds of TLAs (three-letter acronyms) weren't quite confusing enough and changed the official meaning of this one to "Conversational Monitor System." The idea of a "conversation" with the computer, where you type something, it immediately responds, and you continue this interchange, was also a Big Deal in the days when "input" often meant dropping off a pile of punch cards with the guy who fed them into the card reader and "output" meant picking up a printout and maybe another pile of cards from the same guy—not necessarily on the same day. In the days before Apple convinced everyone that moving a mouse pointer to a little picture of a garbage can was a more intuitive way to erase a file than typing the word "erase" followed by a file's name, CMS was actually considered downright user-friendly.

BUZZWORD *Virtual* Unlike most computer buzzwords, this one usually holds true to its dictionary definition when it refers to hardware or software. The Random House College Dictionary defines "virtual" as "being such in force or effect, though not actually or expressly such." On a computer, it usually applies to software or hardware that pretends to be something it isn't. You can designate a portion of RAM on a PC to be treated as a fast, efficient hard disk; we often call this a "virtual disk," because you use it as a disk, when it's not really a disk. Sometimes you can do the opposite—on some systems, software utilities exist that let you designate a portion of your hard disk to act like RAM, giving your computer the effect of having more memory than it really has. We call this "virtual memory," because you use it as memory, without it actually being memory in the modern sense.

The virtual machines created by a VM system aren't real machines, but simulations of separate machines created by the VM operating system. VM also gives its users virtual card punchers and virtual card readers, when actual computer punch cards and the machinery to deal with them are virtually (that word again!) obsolete. (For more on these, see the sidebar "This Is the Nineties. Why Do I Need a Card Puncher or a Card Reader?")

Because it does not manage multiple users, CMS is not a multi-user operating system. If twenty people want to use CMS simultaneously, twenty copies of CMS get started up. VM handles the multiuser part. If ten people try to print a file at once, their individual copies of CMS pass these requests through CP to VM, which deals with print job scheduling.

17.1.1 CP: The Control Program

But what is CP? More IBM initials to worry about? It stands for "Control Program." Different parts of VM deal with the various aspects of the computer's activities in the computer—the disk drives, memory, and so on. CP is the part that deals with CMS. It starts a CMS session and func-

tions as a mediator between the CMS sessions and the hardware.

CP has its own set of commands, but you only need to worry about the one that instructs it to start up a CMS session. (Section 18.1, "Starting Up," covers this in greater detail.) You may hear that some of the commands you use in CMS are actually CP commands, but don't worry; you don't need to treat these commands differently. You don't even need to remember which are CMS commands and which are CP commands. When you type a CP command at the CMS prompt, CMS automatically passes it along to CP.

VM/370? VM/IS? VM/SP? VM/XA? VM/ESA?

You might see other initials besides "CMS" after "VM." Usually, these show the version of VM being used. Various versions developed over the years take advantage of the hardware in different ways; these differences aren't anything for you to worry about. They all run CMS, and the commands that you type are the same for all of them. You might get a memo saying "On Friday the 12th from midnight to 6AM the VM system will be shut down so that we can upgrade from VM/SP to VM/XA. The increased functionality of VM/XA will allow us to utilize the blah blah blah..." Don't worry. The logon screen might look a little different, but you shouldn't need to do anything differently.

Chapter 18 Getting Started with VM/CMS

18.1 Starting Up

Logging on to a VM/CMS system consists of two or three basic steps:

- Entering your user ID.
- Entering your password.
- Telling the CP part of VM to start up CMS. This may be unnecessary.

If CP does not automatically start up CMS, ask your VM administrator (the VM term for a system administrator) to set up your CP directory to take care of this. The CP directory keeps track of who can do what on the system. Only the system administrator has access to the CP directory.

After you turn on your terminal or hook up to a VM system from a PC running terminal emulation software, and before you log on, you will probably see the logon screen with the logo of either the company that owns the system or the version of VM being used.

18.1.1 The Logon Screen

The logon screen will look similar to Figure 18.1.

```

VM/XA SP ONLINE

      VV      VV
     VV      VV
    VV      VV
   VVVVV  MMM  MMM
    V      MMM  MMM
           MM M M MM
           MM M  MM
           MM   MM

```

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

```

USERID  ===> _
PASSWORD ===> _

COMMAND ===>

```

Figure 18.1 Sample VM logon screen.

Your cursor should be on the `USERID` line, waiting for you to type your ID. Type it in, and press your Tab key to move your cursor to the `PASSWORD` line. Type in your password. As the screen tells you, the characters of your password will not appear as you type them, but the cursor moves along to show how many characters you have typed.

After you type your password, you could press Tab again to move your cursor to the `COMMAND` prompt and enter a command for the system to execute as soon as your logon is complete, or you could press Enter immediately to show that you are done with the logon screen. If you press Enter at any time before you enter your user ID and password, the system prompts you to enter the next piece of information it needs.

If you press Enter as soon as you see the logon screen, before you even enter your user ID, the screen clears and you see a prompt like this:

```
Enter one of the following commands:
LOGON userid          (Example: LOGON VMUSER1)
LOGOFF
```

(You may see additional commands suggested, but these two are the important ones.) Enter `LOGOFF` if you want to give up on logging on. To continue the logon procedure, enter `LOGON` followed by your user ID.

If you've entered your user ID but the system still needs your password, it prompts you for it:

```
VMXACI104R Enter logon password:
```

If you entered your user ID and password correctly, the system starts up a session for you. If you enter a user ID that the system doesn't recognize—in this example, `JEOUSER`—it displays something similar to this:

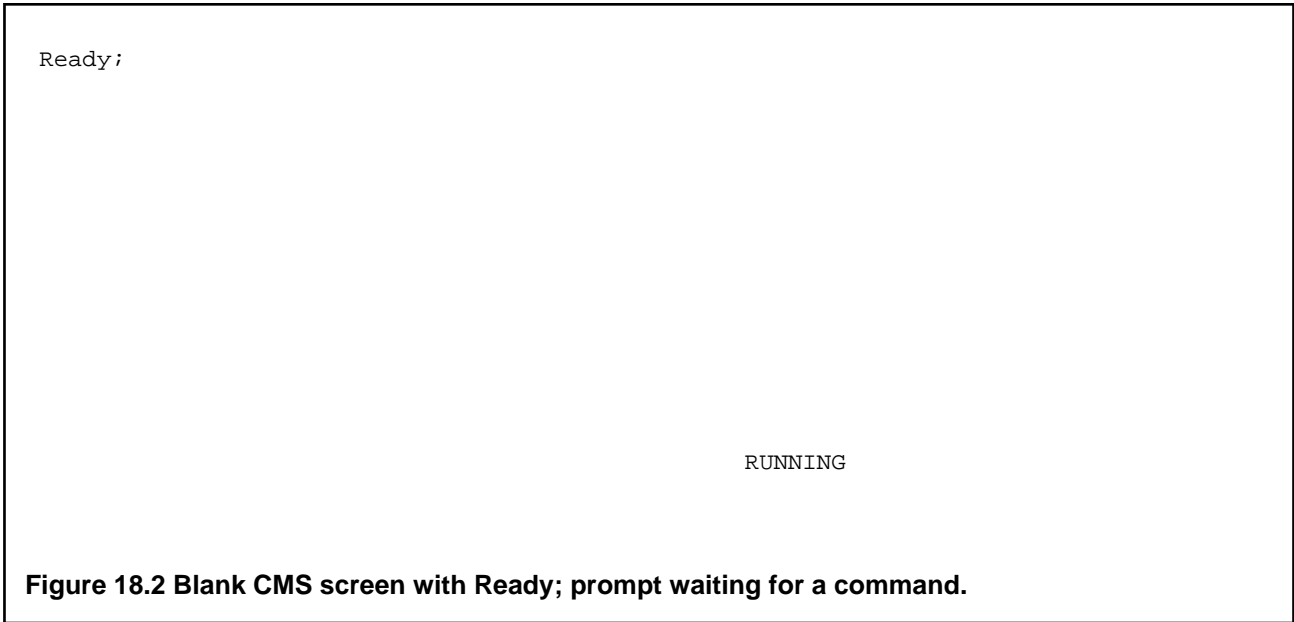
```
HCPLG0053E JEOUSER not in CP directory
Enter one of the following commands:
LOGON userid          (Example: LOGON VMUSER1)
LOGOFF
```

If you entered the password incorrectly, the system tells you

```
HCPLG0050E LOGON unsuccessful--incorrect password
```

followed by the prompt that tells you the syntax for logging on or off. Don't panic right away if your first attempt at logging on doesn't work; the VM system will let you make several more attempts. However, you don't get an unlimited number of chances, because a security feature of the system closes the connection if someone makes too many successive attempts to log on to a particular ID without succeeding. It could be an unauthorized person trying to guess a password, and you don't want to give anyone an unlimited number of guesses.

Once the logon procedure finishes, if you automatically enter `CMS`, you see the `CMS Ready;` prompt (as shown in Figure 18.2) which tells you that the system is ready to accept your commands.



Instead of the word `Ready;` you might just see `R;` as the prompt. The prompt may also include the amount of time that the computer's processor spent on the last task you requested and the current time of day. For example, at 6:43 PM it might look like this:

```
Ready; T=0.01/0.01 18:43:47
```

18.1.2 Entering CMS

If you don't see a CMS prompt, you will see a screen waiting for a CP command like the one in Figure 18.3.

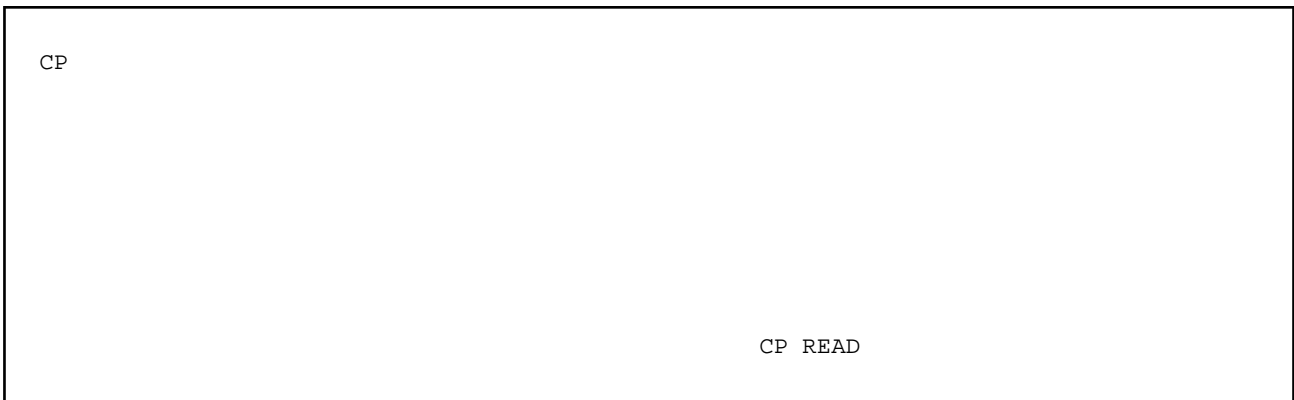


Figure 18.3 Blank CP screen waiting for a CP command.

The only CP command you need to worry about is the one that starts up CMS. Enter

```
IPL CMS
```

IPL stands for "Initial Program Load." After you type in this command and press Enter you will see some variation of the CMS *Ready;* prompt as described above. (You may need to press Enter one or two more times.)

The IPL CMS command may come in handy later in your session. If something goes drastically wrong with your CMS session, you may find yourself back in CP, with a prompt screen just like Figure 18.3. If this happens, type

```
BEGIN
```

which will hopefully resume the CMS session that you just lost. If this doesn't work, type

```
IPL CMS
```

as if you were starting up a brand new session. You really are starting up a brand new session, because you have permanently lost your original session.

18.1.3 Entering Commands

If you press Enter without typing a command, the system displays a message that tells you which environment you are currently in (either CMS or CP.) This is a handy trick if you are unsure of your present situation.

When you type a command it appears, as you type it, at the bottom of your screen, like the `list all *` being entered in Figure 18.4.

```
Ready;
```

```
list all *_  
  
RUNNING
```

Figure 18.4 Command being entered on CMS screen.

Although you might be entering your command 22 lines below the line where `Ready;` is displayed, we still speak of entering a command "at" the `Ready;` prompt. When you press Enter, the typed command jumps to the line below the most recently displayed line on your screen and the system's response appears under it, as shown in Figure 18.5.

```
Ready; T=0.01/0.01 18:48:56  
list all *  
ALL      BACKUP    A1  
ALL      NOTEBOOK A0  
ALL      NOTEINDX A0  
  
RUNNING
```

Figure 18.5 CMS response to command entered in Figure 18.4.
--

If you enter a command like `COPY` or `RENAME` and no problems occur when you execute it, CMS won't display any message telling you that the command executed successfully. If you don't get an error message, you can assume that the command worked as you had hoped.

18.1.3.1 MORE . . .

Unlike other command-driven operating systems you may have used, CMS doesn't scroll your typed commands and the system's responses off the top of the screen when the screen fills up. Instead, the message "MORE..." appears in the screen's lower-right hand corner, to indicate that there is more to see in addition to the currently displayed text. If you wait 60 seconds CMS automatically clears the current screen and displays the next screen.

If you don't feel like waiting, press the Clear key. On an actual mainframe terminal, this key has the word "Clear" written on it. When using a terminal emulation program, check its documentation to see which key on your PC serves as the Clear key with the particular terminal emulation setting being used. (For example, using Procomm on a DOS PC, the gray plus key on your keyboard's numeric keypad is the Clear key when you set the emulation to "3270/950," but F2 is the Clear key when Procomm emulates a VT220 terminal.)

If you want the information on the screen to remain for longer than a minute, press your Enter key. The message "HOLDING" replaces the "MORE..." message, and the screen remains until you press the Clear key.

There are four other messages that may appear in that part of your screen:

RUNNING	CMS is waiting for you to type a command. This is the most common message.
VM READ	CMS (or some other VM program) is waiting for you to enter something. Unlike RUNNING, this appears when the system has asked you for something and is waiting for your answer.
READ	CP is waiting for a CP command (as when you first log on, if you need to start up CMS).
NOT ACCEPTED	You entered a command and pressed Enter before the system was ready for a command, so it couldn't accept it. Press the Reset key (on a PC emulating a 3270 terminal, this is usually the Escape key) and try

again.

18.1.3.2 Aborting Screen Output

If a program has so much output that you press the Clear key over and over and still see the MORE . . . message, you may get tired of pressing the Clear key. For example, if you list file names alphabetically and only start to see file names that begin with the letter "B" after you press Clear eight times, you may regret issuing the command to list file names. (Actually, CMS refers to the names of files as "file IDs," not "file names." See section 18.2, "File Names," for more on this.)

There is help: the Halt Type command, abbreviated as HT. In Figure 18.6, Joe User saw the MORE . . . message in the lower-right and decided that he didn't want to see any more file IDs, so he entered HT to Halt the Typing.

```

BSHFFUTL EXEC      F1 F      80      119      3  9/11/93  1:33:42 65Y370
BSHFINST EXEC      F1 F      80      68       2  9/11/93  1:33:52 65Y370
BSHFPNTR EXEC      F1 F      80     167      4  9/11/93  1:34:02 65Y370
BSHFPROF DATA     F1 F      80      20       1  2/04/94 14:13:15 65Y370
BSHFQCKG EXEC      F1 F      80      39       1  9/11/93  1:34:12 65Y370
BSHFSHLL EXEC      F1 F      80     526     11  2/04/94 14:17:01 65Y370
BSHFSHOW EXEC      F1 F      80      30       1  9/11/93  1:34:34 65Y370
BSHFSMNU EXEC      F1 F      80     184      4  9/11/93  1:34:44 65Y370
BSHFSXST EXEC      F1 F      80     376      8  9/11/93  1:34:56 65Y370
BSHFTOOL EXEC      F1 F      80     119      3  9/11/93  1:35:06 65Y370
BSHFTUTL EXEC      F1 F      80     159      4  9/11/93  1:35:17 65Y370
BSHFVIEW EXEC      F1 F      80     314      7  9/11/93  1:35:28 65Y370
BSHFXFER EXEC      F1 F      80     455      9  9/11/93  1:35:39 65Y370
BSHFFUT1 ERRORS    F1 F      80     149      3  9/11/93  1:36:02 65Y370
BSHFFUT2 ERRORS    F1 F      80     119      3  9/11/93  1:36:12 65Y370
BSHHMFD ERRORS     F1 F      80     167      4  9/11/93  1:36:22 65Y370
BSHSHLL ERRORS     F1 F      80     151      3  9/11/93  1:36:32 65Y370
BSHSHL2 ERRORS     F1 F      80     134      3  9/11/93  1:36:43 65Y370
BSHTOL2 ERRORS     F1 F      80      76       2  9/11/93  1:36:53 65Y370
BSHTOOL ERRORS     F1 F      80     166      4  9/11/93  1:37:03 65Y370
BSHTTUTL ERRORS    F1 F      80     115      3  9/11/93  1:37:13 65Y370
BSHMSACC EXEC      F1 F      80      1       1  9/11/93  1:39:29 65Y37
ht

```

MORE . . .

Figure 18.6 Halt Typing (ht) command entered to stop CMS output.

After he presses Enter, CMS displays only one more line of output and goes back to the Ready ; prompt, as shown in Figure 18.7.

```
BSHMSACC HLP      F1 F          80          11          1  9/11/93  1:38:23 65Y370
ht
Ready; T=0.07/0.36 18:09:37
```

RUNNING

Figure 18.7 Result of Halt Typing command.

Remember, this not only works for the command that lists file IDs, but for any command that has you pressing the Clear key too often when you look at command output.

This is the Nineties. Why Do I Need a Card Puncher or a Punch Card Reader?

The "virtual computer" created for you as a part of the VM system includes a virtual card reader and a virtual card puncher. Why would anyone want even a software simulation of such archaic pieces of hardware? Before sending files from one user (or computer) to another was as simple as it is today, you sent someone a file by issuing a command to the computer to create a punch card version of the file in the recipient's card reader. In other words, you "punched" it to him. He then "read" the stack of cards by entering the command that told the card reader to read each card in that stack and turn it into a line of a disk file.

One of VM's original advantages was its ability to let people emulate these processes in software, so that physical card punchers and readers became much less necessary. Now that these machines are virtually obsolete, the terminology is still with us. In fact, you may still see references to "cards" now and then on your screen. Instead of referring to those rectangular pieces of paper, your computer is referring to an 80-character unit of information—the amount that a single card used to store.

18.1.3.3 Case Sensitivity

CMS is not case-sensitive. Your commands have the same effect whether you enter them in upper or lower case. The same applies to file IDs: when you create or refer to a file, whether you write out its name in upper or lower case, CMS translates it to upper case.

18.1.4 Finishing Your CMS Session

Logging off is simple. Just type

```
LOGOFF
```

at the Ready prompt. This one command ends both your CMS session and the CP session underlying it. You also get a short report on the statistics of your finished session:

```
CONNECT= 00:02:30 VIRTCPU= 000:01.22 TOTCPU= 000:01.52  
LOGOFF AT 09:20:42 EST MONDAY 011/06/94
```

CONNECT shows how long you were connected to the system, and the other figures show various aspects of the Central Processing Unit time that you used.

18.2 File Names

In CMS, we don't refer to a file's name as a "filename." It has three parts, and the first by itself is known as a "filename." Some IBM literature refers to the combination of the three parts as a "file label"; other IBM literature refers to a file's full name as the "fileid," with the last two letters pronounced out loud, as if it had been written "file ID." The three parts have the following purposes:

filename	If you compare the filename and filetype to a person's first and last name, then the filename is the first name. While a series of files with the same filetype are all in the same family, different filenames distinguish each from the other. The filename can have from one to eight characters and can include letters of the alphabet, numeric digits, and the characters @, #, \$, +, -, :, and _. As suggested in section 1.5, "General Advice," stick with letters, numbers, and the underscore character (_) to stay out of trouble.
filetype	This shows what family the file belongs to. For example, a filetype of EXEC shows that the file is a CMS command file (in section 21.2, "Command Files," we'll see how to write these); a filetype of XEDIT shows that the file is a series of commands used by the CMS text editor XEDIT as a macro; a filetype of COBOL shows that the file contains source code written in the COBOL programming language. The filetype can have from one to eight characters, and can contain the same characters that a filename can.
filemode	This identifies the minidisk where the file is stored. As you'll see in section 18.3, "How Files Are Organized," CMS stores files on disks named after letters of the alphabet, with your own personal disk named A. (Disks A, B, and C may actually be on the same physical disk. To make things easier to organize, disks are divided into sections with each named as if it were a separate disk.) If you omit the filemode when you refer to a file, CMS usually assumes that you're talking about a file with a filemode of A.

When you list file labels, you may see a numeric digit from 0 to 6 after the filemode letter. Different numbers represent different purposes and access rights that users have over the file. The most common digit is 1, the default for files that you create. Files created on your minidisk by the system usually have a 0, which prevents other users from looking at the file, even when they have access to your minidisk. Files that several users may want to share often have a 2 as their filemode digit. (Section 19.2, "Sharing Files between Users," explains how one user can gain access to another's minidisk.)

Here are some typical file labels:

MAY_BUDG	SCRIPT	A1
NOTEBOOK	ALL	A0
MAINMENU	EXEC	B2
120794CR	MEMO	A1

18.2.1 Wildcards

The use of wildcards gives you a fair amount of flexibility with `listfile`, the command used

to list file IDs, but very little with other commands. The two wildcard characters are the asterisk, which represents multiple characters, and the percent sign, which represents individual characters.

In the examples used to explain these characters, assume that your A minidisk has the these files on it:

```
111494JJ MEMO      A1
112894BD MEMO      A1
120794CR MEMO      A1
121794BD MEMO      A1
121794CR MEMO      A1
122094JJ MEMO      A1
LINKUTIL EXEC      A1
MAY_BUD  SCRIPT    A1
JUN_BUD  SCRIPT    A1
NOTEBOOK ALL      A0
PHONLIST TXT       A1
SUM_RPT  EXEC      A1
SUM_RPT  HLP       A1
```

18.2.1.1 The Asterisk

In many commands that operate on files, you can use the asterisk to substitute for the filename, filetype, or filemode. To use the LISTFILE command to list all files with a filename of SUM_RPT, the command

```
LISTFILE SUM_RPT *
```

produces this output:

```
SUM_RPT  EXEC      A1
SUM_RPT  HLP       A1
```

To list files with a filetype of EXEC, the command

```
LISTFILE * EXEC
```

produces this output:

```
LINKUTIL EXEC      A1
SUM_RPT  EXEC      A1
```

To list all the EXEC files on your minidisk and all the ones that you have access to, you could enter

```
LISTFILE * EXEC *
```

but be prepared to use the ht command described in section 18.1.3.2, "Aborting Screen Output." An EXEC is a CMS command file, and you will probably find a large number of them on other disks.

The ERASE command, which erases files, can also use the asterisk. Typing

```
ERASE * EXEC
```

would erase all of your EXEC files. Typing

```
ERASE * EXEC *
```

wouldn't erase all the EXEC files on all of the disks to which you have access; you shouldn't have privileges to alter or erase files on any minidisk besides your own A disk.

Used with the LISTFILE or COPY commands, you can use the asterisk in combination with other letters. Typing

```
LISTFILE 11* MEMO
```

produces the following output:

```
111494JJ MEMO      A1
112894BD MEMO      A1
```

The asterisk doesn't need to go at the end of the filename or filetype in the command. For example, you could type

```
LISTFILE *BD MEMO
```

and get this output:

```
112894BD MEMO      A1
121794BD MEMO      A1
```

18.2.1.2 The Percent Sign

You can use the percent sign with some commands to represent a single character. Use multiple percent signs to represent several letters. For example, using three percent signs to represent three letters in this command

```
LISTFILE SUM_RPT %%%
```

produces this output:

```
SUM_RPT HLP      A1
```

while doing this with 4 percent signs,

```
LISTFILE SUM_RPT %%%%
```

has the following output:

```
SUM_RPT EXEC      A1
```

18.3 How Files Are Organized

In keeping with the VM philosophy of giving you a simulation of your own computer all to your-

self, VM also gives you a simulation of your own disk on which to store your files. Wow! Big stuff, right? Well, it was in 1964.

The mainframe you use may have many hard disks, but in practice it's easier to think of them as one big one divided into multiple minidisks. The minidisk is the basic division of hard disk space; it is analogous to the concept of a volume, directory, or folder on other systems. (Unlike directories or folders, however, you cannot divide up a minidisk into smaller minidisks.)

CMS assigns a letter to each minidisk to make it possible to distinguish a file on one minidisk from a file with the same filename and filetype on another minidisk. The filemode letter represents the minidisk on which a file is stored. (See section 18.2, "File Names," for more on naming files.) Everyone knows their personal minidisk as disk A. The files that comprise the CMS system programs are stored on the S and Y disks.

In section 19.2, "Sharing Files between Users," you'll see that commands are available to make your minidisk accessible to other users and to gain access to other users' minidisks. When you execute ACCESS, the second of the two commands necessary to use someone else's minidisk, part of the command's syntax specifies the letter that you plan to use as a filemode for files on that minidisk. If the system lets you pick the letter yourself, the letters used to designate minidisks are clearly not hard and fast names. If you gain access to Mary Jones' personal disk and call it disk M, she still treats it as her A disk. The next day, you might use the same commands to access her minidisk but choose the letter G to represent it; you'll still be using the same files.

18.3.1 Free Space on Your Disk

When you execute the command QUERY DISK, CMS tells you information about the various minidisks to which you currently have access. In addition to your own A disk and the S and Y disks, your system may have other letters assigned to the minidisks that store application software. The output from the QUERY DISK might look like this:

LABEL	VDEV	M	STAT	CYL	TYPE	BLKSIZE	FILES	BLKS	USED-(%)	BLKS	LEFT	BLK	TOTAL
DSK191	191	A	R/W	2	3390	4096	110		202-56		158		360
65Y-XA	19F	F	R/O	32	3380	4096	727		3915-82		885		4800
MNT190	190	S	R/O	70	3380	4096	279		6547-62		3953		10500
CMSLIB	19E	Y/S	R/O	45	3380	4096	670		6163-91		587		6750

The most important columns of information right now are the M column, which shows the letters assigned to the available minidisks, and the BLKS USED-(%) column. Actually, only the first line of the BLKS USED-(%) column is important, because it applies to your own A minidisk. The second number tells how full it is. In the above illustration, A is 56% full. If this figure exceeds 90%, it's time to either make room on your A disk by erasing some files or to lobby with your system administrator for more space—that is, for a bigger chunk of hard disk space to be allocated as your A disk. (To help your bargaining position, sometimes they're willing to grant you extra space for only a limited time—for example, if a big project means that you'll only need more space until you can get rid of the files needed for the project.)

18.4 Available On-line Help

CMS has extensive on-line help, accessible from either menus or the command line. With either, you will find two types of help: brief and detailed.

Brief help rarely exceeds half a screen in length. It shows the syntax of a command and one or two examples. For example, Figure 18.8 shows the brief help for the LISTFILE command. It's quite brief—the upper-right, where it says "line 1 of 10," shows that the 10 lines that you see (counting the four blank lines) are all that it has. It shows the format, or syntax of the command, and one example of its use. Although the format shows where to put the options to the command (after a left parenthesis following the command name), it doesn't tell you what the options are.

```

COMMANDS LISTFILE      Brief Help Information      line 1 of 10

The LISTFILE command lists the names of files on any disk or SFS
directory you access.

FORMAT: Listfile (options

EXAMPLE: If you need to see a listing, then enter:

                listf

PF1= All      2= Top      3= Quit      4= Return      5= Clocate      6= ?
PF7= Backward 8= Forward  9= PFkeys  10=           11= Related    12= Cursor
DMSHEL241I Press PF11 to get related information.
====>

Macro-read 1 File

```

Figure 18.8 Brief help for LISTFILE command.

Note the use of upper and lower case to show the essential part of the command. All you really need to type is the upper case part. For the LISTFILE command, only the "L" is really necessary (making some CMS commands even terser than UNIX commands!). If you just type the letter "L" at the command line,

l

this has the same effect as typing

```
listfile
```

by itself. Of course, you can put any options after the `l` that you can put after the fully spelled-out `LISTFILE` command.

Some commands require more than the first letter; for example, the format part of the help screen for the `COPYFILE` command shows it as `COPYfile`; this means that you must use at least the first four letters of the word.

While viewing a brief help screen, press `PF1` for "All" to display the complete, detailed help about this command. Figure 18.9 shows how the beginning of this help might look for the `LISTFILE` command.

```

COMMANDS LISTFILE          All Help Information          line 1 of 610
(c) Copyright IBM Corporation 1990

LISTFILE

Use the LISTFILE command to obtain specified information about:

*   CMS files residing on accessed minidisks

*   files and subdirectories in Shared File System (SFS)
    directories.

Format

+-----+
| Listfile | | fn | | ft | | fm | | | | | ( (options...()) ) |
|          | | * | | * | | * | | | | |          |
|          | | + | | + | | + | | | | |          |
PF1= Brief  2= Top    3= Quit  4= Return  5= Clocate  6= ?
PF7= Backward 8= Forward 9= PFkeys 10=      11= Related 12= Cursor

====>

```

Figure 18.9 Beginning of detailed help for the LISTFILE command.

The "line 1 of 610" shows that this is only the tip of the iceberg. If you wish, you can page down and page down and read more help about the `LISTFILE` command than you'll ever need.

18.4.1 Help Function Keys

On either the brief or detailed help screens, the following are the important function keys:

- PF1 *Help or All or Brief:* On a help menu screen, PF1 is the `HELP` key, which has the same effect as the Enter key: when you move your cursor to a choice on the menu and press it, CMS displays help information about the chosen menu item.
- On a screen that displays help about a particular command, PF1 gives you the `All` (that is, detailed) help screen instead of the `Brief` help screen. When viewing a Detailed help screen, PF1 gives you the `Brief` help.
- PF2 *Top:* If you used PF8 to page through a few screens of help information, PF2 returns you to the top.
- PF3 *Quit:* Quit the current help screen. When using help menus, you may go from one screen to another as you narrow your search for the information that you need. The Quit key returns you to the previous screen that you were viewing. Repeatedly pressing it brings you back to the `CMS Ready;` prompt.
- PF4 *Return:* Return directly back to the `Ready;` prompt, no matter how many levels of help you went through to get to the current screen.
- PF7 *Backward:* Page up, or view help text above the currently displayed text.
- PF8 *Forward:* Page down, or view help text below the currently displayed text.
- PF11 *Related:* Display a menu of related help topics to choose from.
- PF12 *Cursor:* If the cursor is on the menu part of a help menu, put it at the command line; if it's at the command line, put it at its previous position on the screen.

18.4.2 Help Menus

Type the word `HELP` by itself at the `CMS Ready;` prompt to start up the Help program. The first screen you see, shown in Figure 18.10, is its main menu.

```
HELP TASKS            Task Help Information            line 1 of 25
(c) Copyright IBM Corporation 1990
```


Chapter 18 Getting Started with VM/CMS

Move the cursor to the task that you want, then press the ENTER key or the PF1 key.

```
TASKS      - Help if you don't know VM/ESA commands.
            Good choice for beginners
MENUS      - List the HELP component MENUS
HELP       - Explain some ways for using HELP
COMMANDS   - List VM/ESA commands that you can use
CMS        - Show only CMS commands
CP         - Show only CP commands
OPTIONS    - Show options for the QUERY and SET
            commands of both CMS and CP
SUBCMDS    - List VM/ESA subcommands that you can
            use, such as XEDIT
STATEMTS   - Show statements for REXX, EXEC2, and EXEC
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
```

Figure 18.10 First page of main CMS help menu.

STATEMTS is not the last choice on this menu; press PF8 to page forward, and you see more of the menu, as shown in Figure 18.11.

```
HELP TASKS      Task Help Information      line 9 of 25
MENUS          - List the HELP component MENUS
HELP          - Explain some ways for using HELP
COMMANDS      - List VM/ESA commands that you can use
CMS           - Show only CMS commands
CP            - Show only CP commands
OPTIONS       - Show options for the QUERY and SET
              commands of both CMS and CP
SUBCMDS       - List VM/ESA subcommands that you can
              use, such as XEDIT
STATEMTS     - Show statements for REXX, EXEC2, and EXEC
ROUTINES     - Show callable routines
MACROS        - Show Assembler Language Macros for
              CMS, APPC/VM, and IUCV
MESSAGES     - Explain how to get help for messages
OTHER        - Show commands for other products,
              PRODUCTS such as RSCS and SQLDS
LIBRARY      - Describe the VM/ESA library
* * * End of File * * *
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

Figure 18.11 Second page of main CMS help menu.

That's everything for this particular menu. The first clue about its total number of choices was the first screen's upper-right hand corner, which said line 1 of 25. This meant that the top of your screen had the first of a total of 25 lines in this help file. (The second clue is the line at the bottom of the second screen that says * * * End of File * * *.) Other help files are longer; the PF8 key lets you go through them a page at a time.

Press PF7 to go back to the first screen. Note that part of the description for the TASKS menu choice says "Good choice for beginners." That sounds good; move your cursor to any position on that line and press either Enter or PF1. CMS displays the TASK HELP menu, shown in Figure 18.12.

```

TASK TASK          Task Help Information          line 1 of 23
(c) Copyright IBM Corporation 1990

Each VM/ESA task below leads to a more detailed
list of tasks for you to choose from.

Move the cursor to the task that you want, then 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
Work with virtual screens in CMS
Scroll data in a window
Communicate with other users
Use the CMS Shared File System (SFS)
Debug programs and EXECs
Develop programs and EXECs
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

```

Figure 18.12 First screen of "Task Help" menu.

This is great, because on most computers, if you want help with a command, you must remember the command's name to ask for help with it. In CMS, this help screen helps you find the information you need even when you don't know the command's name. (Forgetting the command names is difficult—one of the nice things about CMS is the sensible choice of command names: ERASE to erase a file, COPY to copy a file, RENAME to rename a file, LIST to list their names, and so forth.)

Of the choices on this screen, `Manage files` covers the most basic ways to manipulate files; if you move your cursor there and press Enter or PF1, CMS displays the screen shown in Figure 18.13.

```

MANAGE TASK          Task Help Information          line 1 of 18
(c) Copyright IBM Corporation 1990

PURPOSE: To help you manage your files.

Move the cursor to the task that you want, then press the ENTER key
or the PF1 key.

List files and associated information
List the files on a minidisk (names only)
Print files
Make copies of files
Rename files
Compare files
Discard or Erase files
Send files to other users
Work with files sent by other users
Specify patterns in file identifiers
PF1= Help      2= Top      3= Quit      4= Return      5= Clocate      6= ?
PF7= Backward 8= Forward  9= PFkeys 10=          11=             12= Cursor

====>
                                         Macro-read 3 Files

```

Figure 18.13 Beginning of "Manage Files" help menu.

If you pick `List files on a minidisk`, the help system displays Brief help explaining the command that lists file IDs, as shown in Figure 18.14.

```

CMS LISTFILE        Brief Help Information          line 1 of 10

The LISTFILE command lists the names of files on any disk or SFS
directory you access.

```

```

FORMAT: Listfile (options
EXAMPLE: If you need to see a listing, then enter:
           listf

PF1= All      2= Top      3= Quit      4= Return      5= Clocate    6= ?
PF7= Backward 8= Forward  9= PFkeys  10=           11= Related   12= Cursor
DMSHEL241I Press PF11 to get related information.
====>
                                           Macro-read 4 Files

```

Figure 18.14 Brief help for "List files on a minidisk."

Section 18.4.3, "Command-Line Help," shows a quicker way to find help about a particular command.

18.4.3 Command-Line Help

We saw earlier that typing `HELP` by itself at the command line brings up the help program's main menu. If you add the name of a command to the `HELP` command, it skips the menus and displays help information about that command. For example, if you type

```
HELP COPY
```

at the `Ready;` prompt, the help program jumps right to the screen shown in Figure 18.15.

```

COMMANDS COPY      Brief Help Information      line 1 of 9
The COPYFILE command lets you copy and/or modify files located on
CMS minidisks or in an SFS directory.

FORMAT: COPYfile fname1 ftype1 fmode1 fname2 ftype2 fmode2
(options

EXAMPLE: To copy HISTORY SCRIPT H to HISTORY SCRIPT A, enter:
           copy history script h history script a

```

```

PF1= All      2= Top      3= Quit      4= Return      5= Clocate      6= ?
PF7= Backward 8= Forward  9= PFkeys   10=           11= Related    12= Cursor
DMSHEL241I Press PF11 to get related information.
====>
Macro-read 1 File

```

Figure 18.15 Brief help for COPYFILE command.

This shows the brief version of help for the COPYFILE command. If you want to go directly from the Ready; prompt to the detailed help about a command, enter

```
HELP command (DETAIL
```

where command is the name of the command with which you need help.

If we can abbreviate LISTFILE to one letter and COPYFILE to four, maybe we don't need to completely spell out DETAIL. Let's use the HELP program to find out. Since we want to know about the syntax of the HELP command, enter HELP followed by the name of the command you're inquiring about:

```
HELP HELP
```

The HELP program displays the screen shown in Figure 18.16.

```

COMMANDS HELP      Brief Help Information      line 1 of 10

The HELP command tells you how to use VM/ESA HELP. For
introductory information on the VM/ESA HELP Facility itself, enter
HELP SELF.

FORMAT:  Help component-name command-name (options

EXAMPLE: You forgot how to use the ACCESS command.  If you want to
get help for that command, then enter:
          help cms access

```

```
PF1= All      2= Top      3= Quit      4= Return    5= Clocate   6= ?
PF7= Backward 8= Forward  9= PFkeys   10=          11=          12= Cursor

====>

Macro-read 1 File
```

Figure 18.16 Brief help for HELP command.

It looks like the brief help is a little too brief to answer our question. Press PF1 to display the detailed help shown in Figure 18.17.

```
COMMANDS HELP      All Help Information      line 1 of 678
(c) Copyright IBM Corporation 1990

HELP

Use the HELP command to display online information from the VM/ESA
HELP Facility. Online HELP provides information for both new and
experienced users.

*   Tasks

    -   End-user, application programmer, and administrator

*   Commands

    -   For VM/ESA (ESA Feature): CMS, CP, AVS, TSAF, VMSES and
        dump viewing facility

    -   For VM/ESA (370 Feature): CMS, CP370, AVS, TSAF, VMSES and
PF1= Brief      2= Top      3= Quit      4= Return    5= Clocate   6= ?
PF7= Backward  8= Forward  9= PFkeys   10=          11=          12= Cursor

====>

Macro-read 1 File
```

Figure 18.17 Beginning of detailed help for the HELP command.

the syntax of the HELP command seems to begin just below this screen, so the information about the use of the word "detail" can't be far off.

If you continue to press PF5 after finding a search target, it continues to search for more occurrences of the target. Press it a second time after you get the screen in Figure 18.18 and you'll see a screen similar to the one shown in Figure 18.19.

```

COMMANDS HELP          All Help Information          line 80 of 678
|                   |  DETail  |
|                   |  RELated  |
|                   |  +      |
| OptionB: (ALL)  (DESCript)  (FORMat)  (PARMs)
|           (OPTions)  (NOTEs)  (ERRors)
|                   |  +      |
| OptionC:  |SCREen|  |TYPE|  (EXTend)
|           |NOScreen|  |NOType|
|           |  +      |  +      |
|           |  +      |  +      |
+-----+
Operands

Help, specified without any parameters, displays a task menu if
you are using the VM/ESA HELP files.  If you are using files
other than the VM/ESA HELP files and if the HELP HELPMENU file
has been created, then you will get the HELP HELPMENU file.
PF1= Brief    2= Top    3= Quit    4= Return    5= Clocate    6= ?
PF7= Backward 8= Forward 9= PFkeys 10=         11=         12= Cursor

====>
Macro-read 1 File

```

Figure 18.19 Result of second search for the word "detail."

There it is: the word "DETail" as part of the detailed help on the HELP command's format. It looks like you only need the first three letters. In the future, to go directly to the detailed help about any command, you only need to type

```
HELP cmdname (DET
```

where cmdname is the name of the command you want to learn about. You'll find that many CMS programs, like the XEDIT text editor, have their own built-in help. While using a program, there are two easy ways to check whether it has its own help facility:

- If the program has its own command line, type HELP there and press Enter.

- Press the PF1 key. CMS programs are pretty consistent about always using this as the Help key.

18.4.3.1 Command-Line Help and Error Messages

When you get an error message in CMS, you usually get a cryptic code along with the brief description of what went wrong. Command-line help can use this code to get much more detailed help about the error. See section 19.1.2, "Common Error Messages," for more information.

Chapter 19 Using Files in VM/CMS

19.1 The Five Most Important Commands

The five most important commands in VM/CMS are:

LISTFILE	lists file IDs.
TYPE	displays the contents of files.
COPY	copies files.
RENAME	renames files.
ERASE	deletes files.

In addition to these commands, CMS has a program that many people use to perform all these functions and more. Section 21.4, "The FILELIST Program," introduces this program.

19.1.1 Command Options

CMS uses a left parenthesis to indicate the beginning of special instructions about how a command should operate. A right parenthesis is optional, and has no effect. For example, the LISTFILE command lists file IDs by themselves with no other options, but adding the LABEL option indicates that the system should also list other information about the files:

```
LISTFILE (LABEL
```

When you use the on-line help in CMS, the brief version of help shows you where to put the options, but you need to view the detailed help about a command to learn the available options and the syntax for using them. See section 18.4, "Available On-line Help," for more about brief versus detailed help.

19.1.2 Common Error Messages

When you type anything at the Ready; prompt, CMS looks for a program with that name and executes it. If you make a typing mistake, for example

```
HLEP COPY
```

when you meant to type HELP COPY, CMS tells you:

```
Unknown CP/CMS command
```

In other words, there is no CMS or CP command called HLEP.

Many commands expect some information after the command. For example, when you copy a file, you must indicate the file to copy and the name you want to give to the new copy. If you type the COPY command by itself without any filenames, the system responds with the message

```
DMSCPY042E No fileid(s) specified
```

The second part of this tells you the problem: you didn't specify any file ID with the command, which needed at least one. And remember: a complete file ID has three parts, and forgetting the third part is another common cause of error messages. For example, the command

```
copy profile exec
```

causes the system to respond with this message:

```
DMSCPY054E Incomplete fileid specified
```

The letters and numbers at the beginning of these error messages look pretty ugly, but they actually provide the key to some of the best error messages you will find in any operating system. Using the HELP program with one of these codes as a parameter tells you much more about your error. For example, entering

```
HELP DMSCPY054E
```

at the Ready; prompt displays the screen shown in Figure 19.1.

```

MSG DMSCPY054E          All Help Information          line 1 of 14
(c) Copyright IBM Corporation 1990

054E          Incomplete <fileid|execid> specified

Explanation:  You must specify the file name and file type or execname
and exectype in order for the command to be executed.  In addition,
for some commands you must specify the file mode.

System Action: RC=24.
Execution of the command is terminated.  The system status remains the
same.

User Response: Check the description of the command, correct the
command line, and reissue the command.
* * * End of File * * *

PF1=          2= Top          3= Quit          4= Return          5= Clocate          6= ?
PF7= Backward 8= Forward    9= PFkeys   10=          11=          12= Cursor

====>
Macro-read 1 File

```

Figure 19.1 Help description of the DMSCP054E error code.

The crucial parts of this are the Explanation, which explains the problem in complete sentences, and the User Response, which tells how to correct the problem.

This is a normal help screen. All of its function keys perform identically to their descriptions in section 18.4.1, "Help Function Keys."

Another popular cause of error messages is instructions that tell the system to perform an action on a nonexistent file. For example, let's say you want to copy a file called `TEMPLATE TXT` and call the copy `MAY_BUD TXT`, but you make a typo and enter the following command:

```
COPY TEMPALTE TXT A MAY_BUD TXT A
```

The CMS system responds with

```
DMSCP002E INPUT file TEMPALTE TXT A not found
```

which means "The COPY program can't find the input file that you named." Again, you can use its error code to learn more if you wish.

Remember, the COPY command is just used as an example here. Similar mistakes with many different commands elicit similar error messages. For a more detailed explanation of the use of the COPY command, see section 19.1.5, "Copying Files."

19.1.3 Listing File Names

The full name of the command to list file IDs is `LISTFILE`. Used by itself without any parameters, it gives you the names of all the files on your own minidisk (the A disk).

Many users abbreviate the `LISTFILE` command as `LISTF` or `LIST`. Its help screen shows that you can abbreviate it down to one character: the "L."

In Figure 19.2, the user types `LISTFILE` and presses Enter; CMS then responds with the first screenful of names.

```
listfile
ALL      BACKUP   A1
ALL      NOTEBOOK A0
ALL      NOTEINDX A0
ANSICODE TXT     A1
BOB      SCRIPT   A1
BOB      TEST     A1
```

```

BOB      TEST1   A1
BUZZ     EXEC    A2
C_ATTACH EXEC    A1
CLOOP    C       A1
CLOOP    TEXT    A1
CMDLIST  TXT     A0
CMS      EXEC    A1
CMS      TXT     A1
COMBINE  XEDIT   A1
DIR      EXEC    A1
DOWNLOAD EXEC    A1
EMPLOYEE DATA   A1
EMPLOYEE INDEX   A1
GRAPH    DATA   A1
GRAPH    EXEC    A1

```

MORE . . .

Figure 19.2 Using the listfile command.

As you can see, there could be more file IDs than will fit on your screen, so you would need to press the Clear key to continue viewing them (or wait 60 seconds between each screenful). Don't forget the HT command described in section 18.1.3.2, "Aborting Screen Output"; it can be handy here.

As explained in section 18.2.1, "Wildcards," the LISTFILE (and FILELIST) commands give you the most flexibility of any of the commands that allow the use of wildcards. (For more information on FILELIST, see section 21.4, "The FILELIST Program.") To list only the files with a filetype of EXEC, use the asterisk to mean "any filename." For example, when you enter

```
L * EXEC
```

CMS displays output similar to this:

```

BUZZ     EXEC    A2
C_ATTACH EXEC    A1
CC       EXEC    A1
CMS      EXEC    A1
CRITBETA EXEC    A1
DIR      EXEC    A1
DOWNLOAD EXEC    A1
EMPATTCH EXEC    A1
FIND     EXEC    A1
GRAPH    EXEC    A1
HTRACK   EXEC    A1
INFONT   EXEC    A1
IOBETA   EXEC    A1
MARY     EXEC    A1
P        EXEC    A1

```

```

PROFILE EXEC A1
SERVER EXEC A1
STAFF EXEC A1
TEST1 EXEC A1
TEST4 EXEC A1
TUTOR EXEC A1

```

Although the default filemode of listed files is A, using the asterisk to denote a filemode with the LISTFILE command tells the system to list file IDs on any minidisk that match the filename and filetype given. For example, to list all the SCRIPT files (text files with IBM formatting codes) on any minidisk that are accessible to your user ID, enter the following:

```
L * SCRIPT *
```

This command means "list the IDs of files with any filename, a filetype of SCRIPT, and any filemode." The output looks like this:

```

DOCCHAP1 SCRIPT A1
DOCCHAP2 SCRIPT A1
DOCCHAP3 SCRIPT A1
DOCCHAP4 SCRIPT A1
DOCCHAP5 SCRIPT A1
DOCCHAP6 SCRIPT A1
DOCCHAP7 SCRIPT A1
GCC024 SCRIPT Y2
GCC025 SCRIPT Y2
GCC029 SCRIPT Y2
GCC311 SCRIPT Y2
GCC312 SCRIPT Y2

```

If someone tells you that your system has a report menu program that you start up by typing RPTMENU, you could check out the program's filetype and minidisk location by typing this:

```
L RPTMENU * *
```

If the output was something along the lines of

```

RPTMENU EXEC Y2
RPTMENU HELPCMS Y2

```

then you would know that the RPTMENU program was an EXEC file (a CMS command file) that you can probably look at with the TYPE command or the XEDIT text editor now that you know the file's full name. (You'll learn how to write your own EXECs in section 21.2, "Command Files.") The other file listed probably stores HELP information that RPTMENU EXEC uses if someone presses PF1 while using the RPTMENU program.

19.1.3.1 Listing More than File Names

The LABEL option specifies that you want to see much more than the files' names. Enter

```
LISTFILE * EXEC (LABEL
```

and CMS displays a screen similar to this:

Chapter 19 Using Files in VM/CMS

FILENAME	FILETYPE	FM	FORMAT	LRECL	RECS	BLOCKS	DATE	TIME	LABEL
BUZZ	EXEC	A2	V	70	36	1	8/17/92	17:54:44	DSK191
C_ATTACH	EXEC	A1	F	80	2	1	1/30/92	16:48:41	DSK191
CRITBETA	EXEC	A1	V	19	4	1	1/21/94	11:23:18	DSK191
DIR	EXEC	A1	V	24	1	1	1/29/92	16:27:20	DSK191
DOWNLOAD	EXEC	A1	V	73	31	1	2/02/92	11:01:21	DSK191
HTRACK	EXEC	A1	V	24	2	1	7/27/92	13:17:05	DSK191
INFONT	EXEC	A1	V	19	2	1	12/10/92	13:30:28	DSK191
IOBETA	EXEC	A1	V	19	4	1	1/07/94	12:34:47	DSK191
MBAILEY	EXEC	A1	V	22	1	1	1/30/92	16:49:02	DSK191
PROFILE	EXEC	A1	V	29	8	1	1/18/94	12:47:00	DSK191
TEST1	EXEC	A1	V	27	9	1	2/02/92	10:53:40	DSK191
TEST4	EXEC	A1	V	18	8	1	8/21/92	16:20:19	DSK191
TUTOR	EXEC	A1	V	7	1	1	9/23/93	16:16:57	DSK191

Ready; T=0.01/0.01 10:18:00

The screen shows the following information about the files:

FILENAME	The first part of a file ID's three parts. See section 18.2, "File Names," for more on this.
FILETYPE	The second of a file ID's three parts. The filetype shows the type of file on that line.
FM	The filemode, or the third of a file ID's three parts. The filemode is the letter that represents the minidisk where the file is located. Section 18.3, "How Files Are Organized," describes its role more fully.
FORMAT	A format (or "record format") of V means that a file's records have variable length. "Records" here means "lines." The terms were once fairly synonymous, but as the science of dealing with databases advanced, the term "record" took on a more specific meaning.) In other words, the lines aren't all the same length. A format of F means that they have fixed length. If any lines seem shorter than others, spaces are inserted to make their lengths equal.
LRECL	The Logical RECORD Length (pronounced "ell-rekkul," with the accent on the first syllable). For a Fixed format file, this shows each line's length; otherwise, this shows the longest line's length.
RECS	The number of records, or lines.
BLOCKS	How many blocks, or units of hard disk space, the file takes up.
DATE	The date the file was last modified.
TIME	The time of day the file was last modified.
LABEL	A name used to identify the actual physical disk where the file is lo-

cated (as opposed to the virtual disk, which is represented by the file-mode).

The most important information that you ever need to know about a file (besides its name) is its size and when it was last modified. So RECS, DATE, and TIME are the most important columns here. You may need to worry about FORMAT and LRECL if an application program expects a file to have one format but it actually has another. We'll see in section 19.1.5, "Copying Files," that when you copy a file, you can specify the copy's FORMAT and LRECL and make a copy that doesn't cause you trouble if the original did not cooperate with the application that tried to use it.

Forget about BLOCKS and LABEL. Part of the point of RECS and filemodes are to keep you from worrying about these.

19.1.4 Displaying a Text File's Contents

The command to type out a file's contents on the screen is simple: TYPE. For example, if you want to see what's in a file called PHONLIST TXT, type the following:

```
TYPE PHONLIST TXT
```

CMS displays the file's contents on your screen:

```
Frank           x5388
Kate            x3325
Jim             x0543
Kevin           x5453
Tastee Deli     874-5342
Messenger Service 878-0775
Mary            878-3431
Mary @ work     874-5363
Cinema 123      874-6543
```

If the file's contents take up more than one screen, you may end up pressing the Clear key repeatedly, so don't forget about the HT command described in section 18.1.3.2, "Aborting Screen Output." If you don't know how long a file is, you might want to check it out before displaying it with the TYPE command by using the LISTFILE command's LABEL option. For example, if you entered the command

```
LISTFILE PHONLIST TXT (LABEL
```

and CMS told you something like

```
FILENAME FILETYPE FM FORMAT LRECL      RECS    BLOCKS  DATE      TIME      LABEL
PHONLIST TXT      A2 V          70      1046      1  8/17/92  17:54:44 DSK191
```

you would know that, with 1,046 lines, using the TYPE command with this file could keep you pressing that Clear key for a while. It's better to use the XEDIT text editor to look at such a file.

19.1.5 Copying Files

The command to copy a file is `COPY`. (The full name is actually `COPYFILE`, but you only need the first four letters.) You simply tell CMS the name of the file you want to copy and the name that you want to give to the copy. The only tough part to remember is that unlike many other CMS commands, you must include the filemode in both file IDs. It's easy to make this mistake, because so many CMS commands don't require you to specify the filemode for files that have a filemode of `A`.

Let's say you want to make a copy of the file `PHONLIST TXT` and call it `PHONLIST BACKUP`. This is the correct syntax:

```
COPY PHONLIST TXT A PHONLIST BACKUP A
```

If you entered

```
COPY PHONLIST TXT PHONLIST BACKUP
```

CMS would tell you:

```
DMSCP054E Incomplete fileid specified
```

as if to tell you, "If you want to copy this file `PHONLIST TXT P` (it ignores any character after the first in a filemode, unless it's a number designating the file's access rights), you need to give the copy a full three-part name, not just `BACKUP`." If you try to make a copy with the same name as an existing file, CMS displays the following error message:

```
DMSCP024E File PHONLIST BACKUP A1 already exists; specify REPLACE option
```

If you look at the detailed help for `COPY`, you will see that `REPLACE` is one of its many possible options. (Don't forget the left parenthesis before you add an option.) If included, it means "Make a copy of file1 and call it file2, even if a file named 'file2' already exists." Section 19.1.5.1, "Changing a File's Format," describes other options for the `COPY` command.

Using the equal sign can reduce the amount of typing necessary for the `COPY` command if any part of the copy's file ID is the same as that of the source file. For example, instead of typing

```
COPY SCHEDULE TXT A SCHEDULE BACKUP A
```

you could type

```
COPY SCHEDULE TXT A = BACKUP =
```

and the copy would have the same filename and filemode as the original.

You can use the asterisk as a wildcard in the `COPY` command, but not in combination with other letters. It must be used to represent an entire filename, filetype, or filemode. For example, to make copies of all your files that have a filetype of `TXT` so that all the copies have a filetype of

BACKUP, you could type this:

```
COPY * TXT A = BACKUP =
```

19.1.5.1 Changing a File's Format

Sometimes a file might not have the right record format for your needs. For example, if you use a telecommunications program like Kermit to copy a file from a PC to a VM/CMS system, the VM copy will probably have a variable record format (a V shows up as the FORMAT for that file when you list its name with the LABEL option). If this is a data file for use by some CMS application, the application might not read it, claiming that it's in the wrong format.

Instead of actually changing its format, it's better to make a copy of it with a different format. This is safer in case it takes more than one try, because you leave your original intact. To specify the copy's format, use the RECFM and LRECL options:

```
COPY fn1 ft1 fm1 fn2 ft2 fm2 (RECFM F LRECL xx
```

where *fn1*, *ft1*, and *fm1* are the filename, filetype, and filemode of the file you're copying and *fn2*, *ft2*, and *fm2* are the copy's filename, filetype, and filemode. *xx* is the length of the lines in the fixed format copy.

How do you determine what this number should be? 80 is always popular, being the number of characters that fit on one line of a character-based display screen (as well as the number of characters that fit on the ancestor of one of these lines, a punch card). The value listed under LRECL for your source file provides another clue. This shows the length of the longest line. If you pick a smaller value for the record length of your fixed format copy, you will cut off the longer lines and lose data.

If you enter the command

```
COPY PHONLIST TXT A PHONTEST TXT A (RECFM F LRECL 80
```

and then type

```
LISTFILE PHON* TXT (LABEL
```

to check on the results of your copy command, CMS displays something like this:

FILENAME	FILETYPE	FM	FORMAT	LRECL	RECS	BLOCKS	DATE	TIME	LABEL
PHONLIST	TXT	A1	V	78	1046	1	10/11/92	15:36:15	DSK191
PHONTEST	TXT	A1	F	80	1046	1	1/24/94	9:35:13	DSK191

The best clue for the LRECL value for your data file is found by looking at any other data files used by the application that rejected your file in the first place. They'll probably all have the same filetype, since they all serve the same purpose.

For example, say you've used the PC version of the UpRiteBase database program to create a data file called ADDRESS.UPR. You use Kermit to move it to the mainframe and rename it

ADDRESS UPRITE so that it has the same filetype as other data files used by the CMS version of UpRiteBase. The CMS version still doesn't accept it as a data file, so you enter the command

```
LIST * UPRITE (LABEL
```

and CMS shows you the following:

FILENAME	FILETYPE	FM	FORMAT	LRECL	RECS	BLOCKS	DATE	TIME	LABEL
ADDRESS	UPRITE	A1	V	78	342	1	10/11/94	15:36:15	DSK191
INVENTORY	UPRITE	A1	F	80	1046	1	10/10/94	9:35:13	DSK191
EMPLOYEE	UPRITE	A1	F	80	214	1	9/24/94	12:34:42	DSK191

It looks like the other CMS UpRiteBase data files have a fixed format and an LRECL of 80, so you need to change ADDRESS UPRITE to have these attributes. You copy it with the following command:

```
COPY ADDRESS UPRITE A ADDRESS1 = = (RECFM F LRECL 80
```

If the CMS version of UpRiteBase accepts ADDRESS1 UPRITE as a proper UpRiteBase data file, then you can erase ADDRESS UPRITE and rename ADDRESS1 UPRITE with that name.

Whatever you try, make sure that there are no problems with your copy being accepted by the program that will use it before you erase your original file and rename the copy to have the same name as the original. If your LRECL value shortens any lines, you should keep your original as a backup to check for the lost data.

Changing the format can work the other way, if you need to make a variable length copy of a fixed format file. Use the following syntax:

```
COPY fn1 ft1 fm1 fn2 ft2 fm2 (RECFM V
```

Note that the command includes no LRECL value. The system automatically figures it out. It's not hard to guess, if your original file had fixed format records: the length of those records will be the length of the lines in the copy.

19.1.6 Renaming Files

The syntax for renaming a file is nearly identical to the syntax for copying a file:

```
RENAME oldfn oldft oldfm newfn newft newfm
```

where *oldfn*, *oldft*, and *oldfm* are the filename, filetype, and filemode of the file to rename and *newfn*, *newft*, and *newfm* are the three parts of the new file ID. Don't forget the filemodes.

Since you rarely change all three parts of a file's ID when renaming it, the equal sign mentioned in section 19.1.5, ("Copying Files") is handy when renaming files. For example, to rename the file PROPOSED SCHEDULE as APRIL SCHEDULE,

```
RENAME PROPOSED SCHEDULE A APRIL = =
```

requires less typing than

```
RENAME PROPOSED SCHEDULE A APRIL SCHEDULE A
```

You can also use the asterisk the same way that you use it when copying files. To rename all the files with a filetype of SCHEDULE so that they have a filetype of OLDSCHED, type:

```
RENAME * SCHEDULE A = OLDSCHED =
```

19.1.7 Deleting Files

It's better to remember this as "erasing" files instead of "deleting" them, because it makes it easier to remember the command:

```
ERASE filename filetype
```

If you leave off the filemode, CMS assumes that you mean a filemode of A. You may never need to specify the filemode when you erase a file, because you may never have permission to erase any files other than those on your own A minidisk.

You can use the asterisk with the ERASE command the same way you use it with the COPY or RENAME commands, but be careful. If you decide that you want to erase all the files that you don't need with the command

```
ERASE * BACKUP
```

then you might want to use the same file ID specification with the LIST command first:

```
LIST * BACKUP
```

This shows you the names of the files that fit this description, so you can be sure that you don't erase more files than you had planned.

19.2 Sharing Files between Users

IBM mainframes have always made security a high priority. Poking around the hard disk to see what else is on the system is not nearly as easy as it is with UNIX or VMS. There is a way, however, to gain access to minidisks assigned to other user IDs. And remember, each user ID doesn't necessarily represent a user, because some programs get assigned their own user ID when they are installed onto the system. In order to run these programs, you need access to the disk where they are stored.

There are two steps to gaining access to another minidisk:

- Telling CP about the disk to access with the LINK command.

- Telling CMS what to call this new minidisk with the ACCESS command.

The QUERY DISK command (first mentioned in section 18.3, "How Files Are Organized") gives you some good background information that is useful to check before you tell the LINK and ACCESS commands what they need to know. This was the output we saw before from the QUERY DISK command:

LABEL	VDEV	M	STAT	CYL	TYPE	BLKSIZE	FILES	BLKS USED-(%)	BLKS LEFT	BLK TOTAL
DSK191	191	A	R/W	2	3390	4096	110	202-56	158	360
65Y-XA	19F	F	R/O	32	3380	4096	727	3915-82	885	4800
MNT190	190	S	R/O	70	3380	4096	279	6547-62	3953	10500
CMSLIB	19E	Y/S	R/O	45	3380	4096	670	6163-91	587	6750

In addition to the M column that shows the letters assigned to the minidisks, note the VDEV column that shows the three-digit hexadecimal number that CMS uses to keep track of each disk. (Hexadecimal numbers have sixteen possible digits going from 0 through 9 and then A through F, so you occasionally see the first six letters of the alphabet showing up as a digit. The buzzword version of the term is "hex numbers," a variation whose voodoo numerology overtones make it one of the more evocative terms in computer science.) If someone wants to access your A disk, they need to know that your copy of CMS treats it as device number 191. Almost everyone's A disk has a device number of 191.

When you gain access to someone else's minidisk, you add it to your list of accessible minidisks, so you must assign it a letter and a device number. You can use any letter of the alphabet that's not already assigned to a minidisk in your list, and you can use just about any number that's not currently assigned to a minidisk. Let's say we need access to the minidisk of Mary Jones' MJONES user ID, and we're going to call it disk B and give it the device number 192.

First issue the LINK command. Use the following syntax:

```
LINK userid devnumber1 devnumber2 mode password
```

where:

userid	Is the user ID of the person whose minidisk you're going to link to.
devnumber1	Is the virtual device number used by that ID for that disk. (In other words, the number that the ID uses to keep track of which piece of hardware it is.) A user's A disk is usually 191; if not, they have to tell you. They can't expect you to guess it.
devnumber2	Is the device number that your ID will use for the disk. You can pick just about any number you like, as long as it doesn't show up in the VDEV column when you enter the command QUERY DISK, because those numbers show the ones that are already taken. The upper limit on allowable numbers varies from system to system; keep it under 300

and you'll be safe.

mode	Shows whether you want read and/or write access to the disk, and whether it matters to you if someone else is currently using it. The safest value to put here is <code>r</code> , which means you want read access, and only if no one else is currently using the disk. A mode of <code>rr</code> means you want read access even if someone is currently using the disk. This means that you can read files currently in the process of being changed, which can lead to trouble.
password	Is the password required to link to the disk. This is not the same as the password that the other user logs on with; they must have a specific password known as the "read password" for someone to link to their disk to read their files. Another password known as the "write password" is necessary if other users are going to link to their disk to change or add files. CMS has no direct command for setting this password; each installation buys a special program like <code>DIRMAINT</code> or <code>VMSECURE</code> . You start either of these programs by typing its name, and then a series of menus leads you through the necessary steps.

If you link to a disk with software that everyone's allowed to use, there might not even be a password. If you leave it out when you should have included it, the system will ask you for it.

Instead of having people use read and write passwords, the system administrator may assign a set of rules determining who can link to where to do what. People in the same group might get permission to link to each other's disks for read permission, but nothing else. If you try to link to a disk using a mode that a rule prevents you from using, CMS displays an error message telling you that a rule prevents you from making the requested link.

Just because you lack write access to a particular disk doesn't mean you lack read access; because write access leads to trouble more easily, system administrators often withhold this permission when they grant read access.

Now it's time to link to Mary's disk. The `QUERY DISK` command showed that no disk is using the number 192, and Mary said that her read password is `READTHIS`. So we enter this:

```
LINK MJONES 191 192 R READTHIS
```

If the system displays no error messages, it worked.

CP now knows that you can use Mary's disk, but you must still tell CMS. We do this with the `ACCESS` command, which has the following syntax:

```
ACCESS devnumber2 diskletter
```

where `devnumber2` is the same number you used when you issued the `LINK` command—the

number you said you would use to refer to this hardware device. The `diskletter` is the letter of the alphabet that you will use to refer to it in CMS commands. In other words, it's the file-mode of any files that you will use from that disk. You can use any letter that didn't show up in the M column when you entered `QUERY DISK`, because those letters are taken. We now enter

```
ACCESS 192 B
```

and then another `QUERY DISK` to see what happened:

LABEL	VDEV	M	STAT	CYL	TYPE	BLKSIZE	FILES	BLKS	USED-(%)	BLKS	LEFT	BLK	TOTAL
DSK191	191	A	R/W	2	3390	4096	110		202-56		158		360
MKTJXS	192	B	R/O	45	3390	4096	355		883-70		377		1260
65Y-XA	19F	F	R/O	32	3380	4096	727		3915-82		885		4800
MNT190	190	S	R/O	70	3380	4096	279		6547-62		3953		10500
CMSLIB	19E	Y/S	R/O	45	3380	4096	670		6163-91		587		6750

And there it is, labeled "MKTJXS." The `STAT` column for the B disk says "R/O" to show that you have "Read Only" access.

19.2.1 Unlinking the Disk

The system automatically unlinks these disks when you log off, but you may need to undo these two steps without logging off (for example, if your link prevents someone else from linking). You can unlink in two steps, but it can also be done in one with the following syntax:

```
RELEASE devnumber2 (DETACH
```

These terms each play the following roles:

`RELEASE` Undoes the `ACCESS` command.

`devnumber2` Is the device number you've been using to refer to the disk since you accessed it.

`(DETACH` Shows that you want to detach as well as release the disk.

You'll probably want to do a `QUERY DISK` again just to make sure that it worked correctly.

19.2.2 Other Ways to Link

Because `LINK` and `ACCESS` are used together so often, many sites have an `EXEC` command file available that automates these two steps. It might be called something like `ATTACH`. (Don't be surprised if many users at that site aren't even aware that `ATTACH`—or whatever the `EXEC` is called—isn't even a real CMS command. This is the beauty of command files, that creating them is like adding a new command to an operating system.) Just ask if they use `LINK` and `ACCESS` to gain access to other disks, or if they use something else. If the terms `LINK` and `ACCESS` inspire a blank stare, you can be sure that a system administrator at that site has put together an

EXEC for them to use. This won't prevent you from using LINK and ACCESS; in fact, the EXEC that they call will be using it.

Chapter 20 The VM/CMS XEDIT Text Editor

XEDIT (pronounced "eks-edit") is a powerful text editor with plenty for the beginner and even more for the expert. It is popular enough that a DOS version called KEDIT has been developed for PCs. It features extensive on-line help, and you can easily create files called macros that are like command files for the text editor. When you tell XEDIT to execute a certain macro, it executes the series of commands in that file as if the file was a program.

20.1 Entering XEDIT

To invoke XEDIT, you only need its first letter. Enter X followed by the name of the file you want to edit. If it exists, XEDIT displays its first eighteen lines on the screen. If the file doesn't exist, XEDIT creates an empty file with that name. For example, if you don't have a file named KUBLA KHAN and you enter

```
X KUBLA KHAN
```

XEDIT displays a screen similar to the one shown in Figure 20.1.

```

KUBLA      KHAN      A1  F 80  Trunc=80 Size=0 Line=0 Col=1 Alt=0
===== * * * Top of File * * *
===== * * * End of File * * *

=====>

X E D I T  1 File

```

Figure 20.1 Opening XEDIT screen for a new file.

Across the top line, XEDIT displays various details about the file (for more on these details, see section 19.1.3.1, "Listing More than File Names.")

KUBLA	The file's filename.
KHAN	The file's filetype.
A1	The file's filemode.
F	The file's record format: either Fixed or Variable.
80	The length of these fixed-length records (lines of text).
Trunc=80	XEDIT truncates any characters after the 80th.
Size=0	The number of records in the file.
Line=0	The current line number.
Col=1	The column location of a string found using certain advanced commands.
Alt=0	The number of alterations since the last autosave. See section 20.2, "Customizing Your XEDIT Environment" for more on this.

The next two lines show the file's top and bottom. You enter data between these lines. To their left, the five equal signs show the prefix area, where you enter commands to insert, delete, copy, and move lines. At the bottom of the screen, the arrow (====>) shows the command-line, where you enter commands, and the message `X E D I T 1 F i l e` shows that XEDIT is currently editing 1 file. (It is capable of editing several at once.)

20.2 Customizing Your XEDIT Environment

Your screen may look slightly different from this. For example, your command line might be at the top of the screen, your prefix area may show up as sequential numbers (00000, 00001, 00002, etc.). Your prefix area might not be there at all. You can control these details with the XEDIT SET command. For now, type in the following SET commands manually at the XEDIT command line; you'll see in section 20.7.2, "The Automatic Startup Macro," how to make them execute automatically whenever you start up XEDIT.

To move your cursor to the command line, just press your Enter key.

If you don't see a prefix area, enter

```
SET PREFIX ON
```

at the command line and press Enter to display it. If the prefix area displays as numbers instead of equal signs, it won't affect the way you use it. `SET NUMBER ON` and `SET NUMBER OFF` control whether it displays as numbers.

Because old-fashioned terminals could only deal with upper case letters, sometimes XEDIT translates everything you type to upper case. This can be annoying when you type a paragraph, press Enter to move your cursor to the command line, and the whole paragraph gets translated to upper case. To tell XEDIT to preserve a mix of upper and lower case letters, enter

```
SET CASE MIXED
```

at the command line. When writing a program in a language like FORTRAN, which expects all upper case letters, entering the command

```
SET CASE UPPER
```

will translate every new line of text to all upper case letters each time you press Enter.

XEDIT has a great feature called "autosave." You can set it to periodically save your work in a backup file automatically. The backup frequency is determined by how often you press the Enter key. If you like, you can set it to back up your file every time you press Enter, but the extra second or two it takes to save can make this annoying. Setting it to back up your file every fifth time you press Enter is a good interval. Enter the command

```
SET AUTOSAVE 5
```

to do this. As you edit your file, the `Alt=` message at the top of your screen shows how many alterations since the last save. When it reaches 5, XEDIT displays a message similar to the following:

```
DMSXSU510I AUTOSAVED as 100005 AUTOSAVE A1
```

The number used for the filename may vary, but the filetype is always `AUTOSAVE` and the filemode is `A1`.

Note that it doesn't save your file with the same name that you originally gave it. This leaves you the option of aborting your changes if you like. If the system does crash or you lose your connection, you probably will have forgotten your autosaved file's filename, so you can enter

```
LISTFILE * AUTOSAVE (LABEL
```

to see what autosave files you have and when they were created. Don't worry about these autosave files cluttering up your A disk—if your editing session goes without incident and you save your file or abort the session without interruption, the system automatically gets rid of any autosave files.

XEDIT has many other `SET` commands to customize your session. Section 20.7.1, "XEDIT On-

line Help," tells how you can find out about more of them.

20.3 Prefix Commands

In addition to command-line commands, many XEDIT operations are performed with prefix commands. You can add, delete, copy, and move lines by entering one- or two-character commands in the prefix area and pressing the Enter key. (MVS and AS/400 users who find that the screen looks similar to the ISPF or SEU editor screens will be familiar with line commands; prefix commands are those editors' counterpart to this.)

Prefix commands are not limited to the prefix area of the contents of your file; you can also enter them in the prefix area of the `Top of File` and `End of File` lines. (If you couldn't do this, it would be pretty tough to add your first blank line.)

You can enter a prefix command anywhere on the prefix. For example, if you enter `2d`, the command to delete two lines, like this

```
2d===
```

or like this

```
===2d
```

it still works the same. Likewise, if you issue the `SET NUMBER ON` command so that your prefixes appear as numbers, it doesn't matter whether you enter the `2d` as

```
2d001
```

or

```
0002d
```

although it can obviously lead to confusion if you enter prefix commands that use numbers (like `2d`) on a prefix area that is displayed as numbers instead of as equal signs.

If you ever enter a prefix command and then realize, before you press Enter, that you didn't mean to enter that prefix command there, just type spaces over it. The next time you press Enter, XEDIT will put back whatever was there before you entered the prefix command—either equal signs or, if the prefix area took the form of numbers, the appropriate prefix numbers.

20.3.1 Adding New Lines

Before you can type any text into your file, you must add some blank lines. To add one blank line, move your cursor to the prefix of the line that says `* * * Top of File * * *` and type the letter `a` anywhere on the prefix. Figure 20.2 shows where to enter the `a`.

```

KUBLA      KHAN      A1  F 80  Trunc=80 Size=0 Line=0 Col=1 Alt=0
==a== * * * Top of File * * *
===== * * * End of File * * *

```

Figure 20.2 Entering the a prefix command to add a new line in XEDIT.

Press Enter, and XEDIT adds a new line below the one where you entered the prefix command, as shown in Figure 20.3.

```

KUBLA      KHAN      A1  F 80  Trunc=80 Size=0 Line=0 Col=1 Alt=0
===== * * * Top of File * * *
=====
===== * * * End of File * * *

```

Figure 20.3 Result of the a prefix command entered in Figure 20.2.

To add multiple lines, you don't need to do this over and over. Instead, most prefix commands let you put a number before or after them to say "perform this command this many times." If you enter

```
==a7=
```

or

```
==7a=
```

and then press Enter, XEDIT adds seven blank lines below the line where you entered the command.

20.3.2 Moving Your Cursor Around

Your up, down, left, and right cursor keys move the cursor in the direction in which they point.

The Tab key helps you move around more quickly. To move your cursor to the beginning of the previous line, use the Backtab key (if you're using a PC that is emulating a mainframe terminal, press the Shift key and the Tab key simultaneously). If your XEDIT prefix area is displayed and your cursor is on a line of text, Tab and Backtab will jump your cursor to the prefix area. If your cursor is on the prefix area, pressing either key jumps your cursor to the beginning of the appropriate line.

20.3.3 Inserting, Deleting, and Typing over Words and Characters

To add text to blank lines, just move your cursor to the line and type. When you need to move your cursor back to the command line, press Enter; if XEDIT turns all your text into upper case, you'll need the SET CASED MIXED command described in section 20.2, "Customizing Your XEDIT Environment."

To delete an individual character, move your cursor there and press your Delete key. On a 3270 terminal, the delete key has a lower case "a" with a proofreader's symbol for deletion: a line through it that forms a loop. When emulating a 3270 terminal, your emulation software probably has your PC's Delete key doing this job.

To type over existing text, just move your cursor where you want the new text and type.

To insert text, move your cursor to the place where you want to insert it and press the Insert key. On a 3270 terminal, this key has the letter "a" with a carat symbol (^) over it. When you press it, a carat symbol should appear at the bottom of your screen to indicate that you are in insert mode. (When emulating a 3270, your cursor may change shape.) Text that you type in moves any text currently to the right of the cursor further to the right. If the last character gets moved to the column indicated by the TRUNC= value at the top of your screen, XEDIT beeps at you to indicate that you can't insert more characters on that line.

To return to overstrike mode while using a 3270 terminal, press the key marked "Reset." The carat symbol should disappear, and newly typed text takes the place of the characters at the cursor. (When your keyboard "locks up," or refuses to accept input, the Reset key is also useful for freeing up the keyboard.) On most PCs emulating a 3270, the Insert key does the job of the 3270 keyboard's Insert key and the Escape key serves as the Reset key. Check your emulation program's documentation to make sure.

20.3.4 Duplicating Lines

Use the double quotation mark, sometimes called the "ditto" symbol (") to duplicate a line of text. Suppose you entered it next to the line "A stately pleasure-dome decree" as shown in Figure 20.4.

```

KUBLA    KHAN      A1  F 80  Trunc=80 Size=8 Line=0 Col=1 Alt=0

===== * * * Top of File * * *
===== In Xanadu did Kubla Khan
===== A stately pleasure-dome decree
===== Where Alph, the sacred river, ran
===== Through caverns measureless to man
===== Down to a sunless sea.
=====
=====
=====

```

Figure 20.4 Entering the " prefix command for line duplication in XEDIT.

After pressing Enter, the line is duplicated; Figure 20.5 shows the result.

```

KUBLA    KHAN      A1  F 80  Trunc=80 Size=8 Line=0 Col=1 Alt=0

===== * * * Top of File * * *
===== In Xanadu did Kubla Khan
===== A stately pleasure-dome decree
===== A stately pleasure-dome decree
===== Where Alph, the sacred river, ran
===== Through caverns measureless to man
===== Down to a sunless sea.
=====
=====

```

Figure 20.5 Result of the " prefix command entered in Figure 20.4.

If you put a number immediately before or after the ditto symbol, XEDIT duplicates the line that many times. Figure 20.6 shows what would have happened if you had entered 5" instead of just " in Figure 20.4.

```

KUBLA    KHAN      A1  F 80  Trunc=80 Size=8 Line=0 Col=1 Alt=0

===== * * * Top of File * * *
===== In Xanadu did Kubla Khan
===== A stately pleasure-dome decree
===== A stately pleasure-dome decree

```

```

===== A stately pleasure-dome decree
===== A stately pleasure-dome decree
===== A stately pleasure-dome decree
===== A stately pleasure-dome decree
===== Where Alph, the sacred river, ran
===== Through caverns measureless to man
===== Down to a sunless sea.
=====

```

Figure 20.6 Result of using the 5" prefix command to duplicate a line five times.

This command is handy when you need to type something repetitive. You just type it once, make as many copies as you need, and edit the copies.

20.3.5 Deleting Lines

The letter d in a line's prefix deletes that line the next time you press Enter. A number immediately preceding or following it tells XEDIT to delete that many lines, starting with the line where the command is entered. In Figure 20.7, two lines are about to be deleted.

```

KUBLA      KHAN      A1  F 80  Trunc=80 Size=8 Line=0 Col=1 Alt=0

===== * * * Top of File * * *
===== In Xanadu did Kubla Khan
=2d== A stately pleasure-dome decree
===== Where Alph, the sacred river, ran
===== Through caverns measureless to man
===== Down to a sunless sea.
=====
=====

```

Figure 20.7 Entering the 2d prefix command to delete two lines in XEDIT.

After you press Enter, they're gone. Figure 20.8 shows the result.


```

KUBLA    KHAN      A1  F 80  Trunc=80 Size=8 Line=0 Col=1 Alt=0

===== * * * Top of File * * *
===== In Xanadu did Kubla Khan
===== Through caverns measureless to man
===== Down to a sunless sea.
=====
=====
=====
=====
=====

```

Figure 20.8 Result of the `2d` prefix command entered in Figure 20.8.

Put the letter `d` twice in a prefix without any number to indicate that the line begins or ends a block that you want to delete. If you press `Enter` while only one line has the `dd`, XEDIT leaves it there until it has a partner. In Figure 20.9, XEDIT is ready to delete all but the first and last lines of text the next time you press `Enter`.

```

KUBLA    KHAN      A1  F 80  Trunc=80 Size=8 Line=0 Col=1 Alt=0

===== * * * Top of File * * *
===== In Xanadu did Kubla Khan
=dd== A stately pleasure-dome decree
===== Where Alph, the sacred river, ran
===dd Through caverns measureless to man
===== Down to a sunless sea.
=====
=====
=====

```

Figure 20.9 Entering the `dd` prefix command to delete a block of lines in XEDIT.

This is particularly useful when the beginning and end of the block that you want to delete are not on the same screen, because the alternative (counting the number of lines to delete so that you can put a number after a single `d`) is a lot of trouble.

20.3.6 Copying Lines


```
==== Down to a sunless sea.  
==== In Xanadu did Kubla Khan  
==== A stately pleasure-dome decree  
==== Where Alph, the sacred river, ran  
==== Through caverns measureless to man  
==== Down to a sunless sea.  
====  
====
```

Figure 20.11 Result of the `cc` and `p` prefix commands entered in Figure 20.10.

20.3.7 Moving Lines

Moving is similar to copying, except that after you press Enter, the original lines are no longer there—they're moved to their new location. As with copying, there are three ways to specify the block to move, but these use the letter `m`:

- Enter a single `m` in a line's prefix if you only need to move that one line.
- Enter a single `m` preceded or followed by a number to indicate how many lines to move.
- Enter `mm` at the first and last lines of the block to move.

To specify the destination of the block to move, use the letters `p` or `f` the same way you do to specify the destination of a block to copy.

20.4 Searching for Text

To search for a particular string of text, use the following command:

```
LOCATE /string
```

where `string` is the string you want to find. Actually, you don't really need the `LOCATE` part. If you enter the command

```
/ALPH
```

and press Enter, XEDIT scrolls forward to make the the line with "Alph" in it the first line on the screen, as shown in Figure 20.12.

```

KUBLA      KHAN      A1  F 80  Trunc=80 Size=8 Line=3 Col=1 Alt=1

===== Where Alph, the sacred river, ran
===== Through caverns measureless to man
===== Down to a sunless sea on 10/12/94.
=====
=====
=====
===== * * * End of File * * *

```

Figure 20.12 Result of searching for the string "ALPH" with the command /ALPH

Note that although we were searching for "ALPH" it found "Alph." The search is not case-sensitive.

When XEDIT searches, it searches forward. If you now try the command

```
/kubla
```

XEDIT skips to the bottom and tells you "Target not found." To jump to the top of the file before the next search, enter this command:

```
TOP
```

BOTTOM is an equally valid command.

You can use another character instead of the slash to show where your search target begins, but then you must include the LOCATE command. This is useful when searching for something that contains a slash, like a date. If you enter

```
LOCATE #10/12/94
```

at the XEDIT command line and press return, XEDIT jumps to the next occurrence of that string, as shown in Figure 20.13.

```

KUBLA      KHAN      A1  F 80  Trunc=80 Size=8 Line=3 Col=1 Alt=1

===== Down to a sunless sea on 10/12/94.
=====
=====
=====
===== * * * End of File * * *

```

Figure 20.13 Result of searching for the string "10/12/94" with the command #10/12/94

20.5 Saving Your Changes

In section 20.2, "Customizing Your XEDIT Environment," we saw how `SET AUTOSAVE 5` tells XEDIT to automatically back up your file with a special name every fifth time you press the Enter key. To save your file with its normal name, just enter `SAVE` at the command line.

To save a copy with a different file ID, enter the new file ID after the `SAVE` command. For example, entering

```
SAVE CHAKA KHAN
```

saves the file in its current state under the name `CHAKA KHAN`.

20.6 Quitting XEDIT

To quit out of XEDIT, type `QUIT` at the command line. If you made any edits to the file and have not saved it since you made the most recent edit, XEDIT tells you

```
DMSXSU577E File has been changed; type QQUIT to quit anyway
```

The extra "Q" means "Quit no matter what, I don't want to save the changes I've made."

20.7 Other Useful XEDIT Features

XEDIT is one of the most powerful, full-featured text editors that you'll ever find included as part of an operating system's basic tools. We can't leave it without mentioning a few additional features that, without being absolutely essential, can make your life much easier.

20.7.1 XEDIT On-line Help

XEDIT's built-in help works almost identically to the CMS help. Type `HELP` by itself at the XEDIT command line, and you'll see a menu of XEDIT commands. You can move your cursor to one and press Enter to find out more about that command. (You won't see all of them; press PF8 to page down for the complete list.)

As with help from the CMS command line, you can also enter the command `HELP` followed by a

command name if you want help about that command. For example,

```
HELP LOCATE
```

displays the on-line help for the LOCATE command. And, while viewing help information, the PF1 key toggles between brief and detailed help just like it does with CMS help.

20.7.2 The Automatic Startup Macro

You can store a series of XEDIT commands in a file called a macro and then instruct XEDIT to execute that macro. All XEDIT macros have XEDIT as a filetype, and one has a special filename: PROFILE. Whenever you start XEDIT, it looks for this PROFILE XEDIT file and if it exists, executes it automatically. We saw in section 20.2, "Customizing Your XEDIT Environment," that certain commands are useful enough to run every time you start XEDIT. Let's see how to create a profile macro with them.

Start up XEDIT with the command

```
X PROFILE XEDIT A
```

to create this file (or edit the existing one, if it's there). You need to include the A filemode because if you leave it off and XEDIT doesn't find the specified file ID on the current A disk, it looks for a file with that filename and filetype on other disks that your user ID has access to, and you'll be looking at some system default PROFILE XEDIT. You're not allowed to edit one from another disk, so make sure to include the filemode of A so that XEDIT knows to create a PROFILE XEDIT for you if no PROFILE XEDIT A exists.

Use the a3 prefix command to add three blank lines to your file, and type in the commands covered in the section 20.2 so that your screen ends up looking like Figure 20.14.

```
PROFILE XEDIT A1 V 80 Trunc=80 Size=3 Line=0 Col=1 Alt=0
===== * * * Top of File * * *
===== SET CASE M
===== SET PREFIX ON
===== SET AUTOSAVE 5
===== * * * End of File * * *
```

Figure 20.14 Sample PROFILE XEDIT file.

After you save this file, every time you start up XEDIT it automatically executes these three

commands.

If you already have a PROFILE XEDIT file, see if any of these commands are there before you add them. Now you know what they do.

20.7.3 The Split/Join Key

One annoying thing about an editor that deals with text almost exclusively in lines is the effect when you delete words from a line, making it too short, or add words to a line, making it too long. PF11, the Split/Join key, can fix this for a line or pair of lines.

If your cursor is at the end of a line, pressing PF11 joins the following line to the one with the cursor. In Figure 20.15, the cursor is right after the "n" in "Khan."

```
==== In Xanadu did Kubla Khan_
==== A stately pleasure-dome decree
==== Where Alph, the sacred river, ran
==== Through caverns measureless to man
==== Down to a sunless sea.
```

Figure 20.15 Cursor position just before pressing PF11 to join two lines.

Pressing PF11 moves the beginning of the following line to the cursor's location, as shown in Figure 20.16.

```
==== In Xanadu did Kubla KhanA stately pleasure-dome decree
==== Where Alph, the sacred river, ran
==== Through caverns measureless to man
==== Down to a sunless sea.
```

Figure 20.16 XEDIT file from Figure 20.15 after pressing PF11 to join two lines.

If PF11 is not at the end of a line, pressing it splits the line into two at the cursor's location. With

the cursor after the word "did" this time, as shown in Figure 20.17, pressing PF11 moves everything beginning at the cursor's location (including the character at the cursor—in this case, a space) to a new line underneath the one where the cursor is located, as shown in Figure 20.18.

```
==== In Xanadu did_Kubla KhanA stately pleasure-dome decree
==== Where Alph, the sacred river, ran
==== Through caverns measureless to man
==== Down to a sunless sea.
```

Figure 20.17 Cursor position just before pressing PF11 to split a line.

```
==== In Xanadu did
====  Kubla KhanA stately pleasure-dome decree
==== Where Alph, the sacred river, ran
==== Through caverns measureless to man
==== Down to a sunless sea.
```

Figure 20.18 XEDIT file from Figure 20.17 after pressing PF11 to split a line.

Chapter 21 Using a VM/CMS System

21.1 Printing Text Files

Printing a text file is simple. Just enter

```
PRINT fn ft fm
```

at the `Ready;` prompt, where `fn`, `ft`, and `fm` show the filename, filetype, and filemode of the file you wish to print. If you leave off the filemode, the system assumes that the file is on your A minidisk. For example, to print a file named `SEPT1294 MEMO A`, just type this:

```
PRINT SEPT1294 MEMO
```

You'll need to ask where the printer is located in your office. There are probably multiple printers attached to the system, with a particular one specified as your default printer. Ask your system administrator which one your user ID is configured to print on.

21.1.1 Checking the Print Queue

To display a list of the jobs waiting to print, enter the following command:

```
QUERY PRINTER ALL
```

Your output will look similar to this:

ORIGINID	FILE	CLASS	RECORDS	CPY	HOLD	DATE	TIME	NAME	TYPE	DIST
ACCLN	0098	K	PRT 000832	001	NONE	10/18	12:24:43	SALES	DATA	RM234
ACCLN	0099	K	PRT 000110	001	NONE	10/18	12:25:08	MEETING	DATA	RM234
JOEUSER	0103	K	PRT 000110	001	NONE	10/18	12:25:12	SEPT1294	MEMO	RM101
MJONES	0112	K	PRT 000359	001	NONE	10/18	12:26:23	SETUP	EXEC	RM112

The `NAME` and `TYPE` of each waiting file shows its file ID. The `TIME` shows when it was added to the print queue, and the `RECORDS` column shows the length of each file. This is important because it can let you know if there's a huge job ahead of yours in the queue.

The `ORIGINID` column shows who sent each waiting job, and the `FILE` column shows the number, or "spoolid" (pronounced "spool ID") that the printing program uses to refer to the print job. This number is important if you want to cancel the print job.

21.1.2 Cancelling Your Print Job

Cancelling your print job is known as "purging" because of the command used to do it. Use the following syntax:

```
PURGE PRINTER spoolid
```

where `spoolid` is the number that shows up in the `FILE` column when you issue the `QUERY PRINTER ALL` command. (If your file doesn't show up when you query the print queue, either it already printed—a likely possibility, if it was a small file and no other files were ahead of it in the print queue—or it never got into the queue in the first place.) You must include the word `PRINTER` because the `PURGE` command can be used to purge other temporary storage areas, like your reader. (Section 21.3, "Communicating with Other Users," covers some uses of the reader.)

If Joe User wants to purge the `SEPT1294 MEMO` file from the print queue shown above, he enters the following:

```
PURGE PRINTER 0103
```

If he wants to purge the `SALES DATA` file to speed up the printing of his own file, it's too bad. That file belongs to the `ACCLEN` user ID, and only a system administrator can purge other people's files.

21.2 Command Files

Command files in CMS are called EXECs (pronounced "egg-zeks") after their filetype, which is short for "executable." Creating and using them is simple; just use `XEDIT` to create a file with a file type of `EXEC` that has a valid CMS or CP command on each line. To run the EXEC, type in the file's filename.

Nearly all IBM operating systems (with the notable exception of PC/DOS, thanks to the Byzantine politics of IBM's relationship with Microsoft) include some version of a programming language called REXX that is a superset of the EXEC command language. In other words, REXX has all the same commands as EXEC, plus more. Once you learn it on any computer, you can transfer that knowledge (and maybe even the REXX program) to other IBM computers. You can tell a REXX program from an EXEC program by its first line: if it begins with a slash and an asterisk (`/*`) and ends with an asterisk and a slash (`*/`), then it's a REXX program. Anything between these asterisks is considered a comment to the program, so the system ignores it. It doesn't really matter what you put there, but it's best to put a line describing the title and purpose of the program.

You should begin all EXECs (except for the `PROFILE EXEC` described in section 21.2.1, "The Automatic Logon Command File") with this so that you can indicate the purpose of your program and have access to the REXX commands. I've written EXECs that didn't work until I added this first line, so I always add it now.

For example, let's say that you often need to link to a particular minidisk on your system so that you can run a program called `MAINMENU` that displays the main menu of available reports for you to run. It would make your life easier to have a command file that performs the following steps:

- Link to the RPTPROG user ID, which has a virtual address of 192.
- Access the disk as your B disk.
- Type MAINMENU to run the program.

If you want to call your program RPTMENU, you use XEDIT to create a file with a filename of RPTMENU and a filetype of EXEC. Enter the following at the Ready ; prompt:

```
X RPTMENU EXEC
```

Once in XEDIT, you add these four lines:

```
/* RPTMENU EXEC link to the RPTPROG disk and start the MAINMENU program */  
LINK RPTPROG 192 193  
ACCESS 193 B  
MAINMENU
```

Note how the first line describes the purpose of the program. Save it and quit out of XEDIT. From then on you only need to type RPTMENU to use the MAINMENU program instead of the three commands you needed to type before.

In section 18.4.2 ("Help Menus") we saw that entering HELP by itself at the Ready ; prompt displays a menu of various types of information for which help is available. One of the menu choices was the following:

```
STATEMETS - Show statements for REXX, EXEC2, and EXEC
```

Choose this to learn more about EXEC, EXEC2 (a variation on the EXEC command language), and REXX programming.

21.2.1 The Automatic Logon Command File

Whenever you log on to your CMS account, the system looks for a special EXEC called PROFILE EXEC and executes it if found. If there's any command that you want executed every time you log on, use XEDIT to add a line with that command to your PROFILE EXEC file. If you don't have one, create it as you would create any other EXEC.

Many systems are set up to give a default PROFILE EXEC to each new user. In other words, you may have one the first time you log on without even knowing it. If so, take a good look at it; if your system doesn't seem to behave the way this book describes, it's probably because of something in your PROFILE. (For example, if a function key acts differently than I've described or if the system automatically displays a menu when you log on.)

21.3 Communicating with Other Users

Sending mail or files to another user means putting a copy of it into their "reader," which is like a mailbox. It's actually a software simulation of the machine that was once used to read punch cards. The sidebar "This is the Nineties. Why Do I Need a Card Puncher or a Punch Card Reader?" in section 18.1.3.2 ("Aborting Screen Output") explains why we still use the term "reader" even though the hardware being simulated is so archaic.

All CMS systems have the NOTE program to send electronic mail to other users. NOTE gets the job done, but it's limited enough that many sites purchase a fancier program. On my first CMS system, I was unaware of the existence of NOTE because of a program that we used called MAIL. Keep your eye out for a more full-featured alternative to NOTE.

Sending a note is simple: type

```
NOTE userid
```

where `userid` is the ID of your note's recipient. The recipient doesn't need to be logged on. If not, the next time he or she logs on a message will appear telling them that something is in their reader.

If Joe User wants to send a note to Mary Jones at her MJONES user ID, he enters this:

```
NOTE MJONES
```

CMS then displays the note editing screen, as shown in Figure 21.1.

```

JOEUSER NOTE      A0  V 132  Trunc=132 Size=9 Line=9 Col=1 Alt=0
* * * Top of File * * *
OPTIONS: NOACK    LOG    SHORT    NOTEBOOK ALL

Date: 6 February 94, 19:20:11 EST
From: Joe User           212/930-3342 x4288    JOEUSER at JUPITER
To:  MJONES

* * * 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

====>

```

Figure 21.1 Screen for entering a note to another user.

It looks a lot like the XEDIT screen when you're creating a new file, except that it automatically added a header to your message. The "at JUPITER" part shows that Joe's particular VM system is named JUPITER. The system name is necessary when multiple systems are connected and someone on one wants to send mail to a user ID on another.

All the XEDIT commands described in this book are valid here. You can start with `SET PREFIX ON` and then use the `a` prefix command to add some blank lines on which to type your message.

Two function keys speed the creation and delivery of your message:

- Press PF2 to add blank lines to your message. Move your cursor to any line above "End of File," (for a start, put it on the line that says "To:") and press PF2 to add a new blank line under the line with the cursor.
- Use PF5, the "Send" key, to show that you have completed your note. When you press it, the NOTE program sends your message to the recipient's reader.

It also saves a copy in a file on your A disk called `ALL NOTEBOOK`. This is handy when you want to see exactly what you sent to someone.

21.3.1 Sending Files

The `SENDFILE` command makes it easy to send an existing file to someone's reader. At the `Ready;` prompt, enter

```
SENDFILE fn ft fm userid
```

where `fn`, `ft`, and `fm` show the ID of the file to send and `userid` shows the ID of the user who gets the file. If CMS does not find a single letter as the third parameter here (in other words, if you leave out the filemode and just say `SENDFILE fn ft userid`), it assumes that `fn ft` represents a file on your A minidisk. Make sure to type all eight letters of the `SENDFILE` command; CMS has a separate, unrelated command called `SEND`.

If you enter `SENDFILE` by itself, you will see a screen similar to Figure 21.2.

```

----- 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

```

Figure 21.2 SENDFILE screen for sending a file to another user.

In addition to filling out the file ID and recipient's name, you can set the options shown in the bottom half of the screen. Most of them are self-explanatory. For example, the "log entry" means that SENDFILE adds a line describing what you sent, when, and to whom to a file called `userid NETLOG`, where `userid` is your user ID name.

When you are done with this screen, press PF5 to send the file whose ID you entered or PF3 to abort the SENDFILE program.

21.3.2 Receiving Mail and Files

When you first log on, you may see a message similar to this:

```
FILES: 0003 RDR,   NO PRT,   NO PUN
```

This shows that your reader has three files. To get them out, you could use the RECEIVE command, which people originally used to read real punch cards and turn their information into a disk file. However, the RDRLIST program makes it much easier. (If your reader is empty and you want to play with the RDRLIST program for practice, use the NOTE and SENDFILE commands to send a few things to your own ID's reader.)

Enter RDRLIST at the Ready; prompt, and you see a screen similar to Figure 21.3.

```

JOEUSER RDRLIST A0 V 108 Trunc=108 Size=2 Line=1 Col=1 Alt=0
Cmd  Filename Filetype Class User  at Node   Hold  Records  Date      Time
     MJONES   NOTE      PUN A MJONES  JUPITER NONE    14 10/16   09:30:59

```

```

ACCLEN  NOTE      PUN A ACCLEN  JUPITER  NONE      4 10/16   13:01:44
SALES   DATA     PUN A ACCLEN  JUPITER  NONE      832 10/16 13:02:45

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

```

Figure 21.3 RDRLIST screen for viewing files in the reader.

RDRLIST is a specialized version of FILELIST, which is explained in the next section. Most of the columns of information here are similar to the information displayed when you enter the LISTFILE command with the (LABEL option, but with two important new columns: the User column shows who sent the file, and the Cmd column provides a place where you can enter commands to act on the file named on a given line. You can also move your cursor to the line with a particular file's name and press function keys to perform actions on that file. (See section 19.1.3.1, "Listing More than File Names," for more on the (LABEL option to the LISTFILE command.)

The three most important things that you want to do to a file in your reader are:

- Look at it while it's still in the reader.
- Pull it out of the reader and save it as a disk file.
- Delete it from the reader.

21.3.2.1 Looking at a File in the Reader

Unless someone told you exactly what they sent you, you'll want to look at each file first, so that you can decide whether to keep it. Use the Peek function key for this. Move your cursor to anywhere on the line showing the file you want to peek at and press PF11. For example, let's say Joe recognizes the ACCLLEN user ID as that of Larry Niven, his friend in Accounting. If Joe moves his cursor to the ACCLLEN NOTE line and presses PF11 to see the note that Larry sent him, he might see something like the screen shown in Figure 21.4.

```

0023      PEEK      A0  V 132  Trunc=132 Size=12 Line=0 Col=1 Alt=0
Note from ACCLLEN at JUPITER Format is NETDATA
* * * Top of File * * *
Date: 16 October 94, 11:43:24 EST
From: Larry Niven          212/930-3342 x3277  ACCLLEN  at Jupiter
To:   JOEUSER

Joe -

I can't make lunch with you today.  Preparing for the sales meeting Monday
looks like a lot more work than I had originally thought.  A lot of the
figures still aren't in yet.  How about Monday after the meeting for lunch?

- Larry
* * * End of File * * *

1= Help      2= Add line  3= Quit      4= Tab      5= Clocate  6= ?/Change
7= Backward  8= Forward   9= Receive 10= Rgtright 11= Spltjoin 12= Cursor

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

```

Figure 21.4 Sample received note.

It looks like an XEDIT screen because it is. The RDRLIST program uses XEDIT to display the received message. As the function key list at the bottom shows, pressing PF3 quits out of this display and returns you to the list of files in your reader.

21.3.2.2 Saving a File from the Reader

PF9 "receives" the file, or takes it out of the reader and saves it as a disk file. If someone sent it to you with the NOTE command, receiving it appends it onto your ALL NOTEBOOK file. If someone sent it to you with the SENDFILE command, receiving it saves it on your disk with the file ID specified in the Filename and Filetype columns—in other words, the name it had before it was sent to you. PF9 also sends a brief message to the message's sender telling them

that you've received it. (This brings up one advantage of PF11, the Peek function key: you can look at a received file, but the sender never knows that you saw it.)

Note how the function key lists on the RDRLIST screen and on the screen used to peek at files both show that PF9 receives the file. At the RDRLIST screen, your cursor must be on the same line as the file you want to receive when you press PF9, so that the system knows which file to receive. At the PEEK screen, your cursor's location doesn't matter, because the file you're peeking at is obviously the one you want to receive.

PF9 is a shortcut to use instead of typing out the RECEIVE command. It won't handle special situations, like receiving a file that has the same name as an existing file. To do this you need to type RECEIVE at the PEEK screen command line or in the RDRLIST screen's Cmd column so that you can take advantage of one of its options.

Although the Cmd column isn't very wide, you can type longer commands that begin there, even if this means typing over the Filename and Filetype columns. When you press Enter, the system knows what you mean.

To receive a file that is displayed on the PEEK screen and assign it a new name, enter the RECEIVE command at the PEEK screen prompt followed by the file ID. For example, to save a displayed file and give it the name LNIVEN 101694, enter

```
RECEIVE LNIVEN 101694
```

at the command line. To receive a file that is displayed on the PEEK screen and have it replace your existing file that has the same name, enter

```
RECEIVE (REPLACE
```

at the PEEK screen command line.

These commands have the same effect when you type them in the RDRLIST screen's Cmd column, except that you must put a slash character after the RECEIVE command to stand in for the name of the file being received. For example, to save the file with the name LNIVEN 101694 type

```
RECEIVE / LNIVEN 101694
```

over the beginning of the line that describes that file, as shown in Figure 21.5.

Cmd	Filename	Filetype	Class	User	at	Node	Hold	Records	Date	Time
RECEIVE /LNIVEN	101694		PUN	A	ACCLEN	JUPITER	NONE	4	10/16	13:01:44
	SALES	DATA	PUN	A	ACCLEN	JUPITER	NONE	832	10/16	13:02:45

Figure 21.5 Receiving a file from the reader.

After you press Enter, the system acknowledges your command with a message similar to the one in Figure 21.6.

```

JOEUSER RDRLIST A0 V 108 Trunc=108 Size=2 Line=1 Col=1 Alt=0
Cmd  Filename Filetype Class User  at Node    Hold  Records  Date      Time
*   MJONES   NOTE      PUN A MJONES  JUPITER NONE    14  10/16    09:30:59
    LNIVEN   101694   received from ACCLLEN at JUPITER
    SALES    DATA    PUN A ACCLLEN  JUPITER NONE    832 10/16    13:02:45

```

Figure 21.6 System acknowledgement of receipt of file from the reader.

Similarly, you type `RECEIVE /` (REPLACE in the same place to pull it out of the reader and store it on your disk with its original name regardless of whether you already had a file with that name).

21.3.2.3 Discard a File from the Reader

Don't think of it as "deleting" a file from the reader, but as "discarding" it. After all, you'll remember the command more easily: `DISCARD`. You can type it at the `PEEK` screen command line or on the `RDRLIST` screen beginning in the `Cmd` column. No parameters are necessary. Either way, when you press Enter, `RDRLIST` displays the words "has been discarded" after the file ID.

21.4 The FILELIST Program

Many people find the `FILELIST` program useful to carry out basic operations on files. It displays a list of file IDs and you enter commands next to the names of the files that you want to manipulate. In fact, the `RDRLIST`'s interface is based on `FILELIST`.

Entering `FILELIST` by itself lists your A disk's files on a screen similar to the `RDRLIST` screen. You can also enter wildcards to show that you only want to see certain files listed in `FILELIST`. For example, if you type

```
FILELIST * EXEC
```

FILELIST starts up and displays a screen similar to the one shown in Figure 21.7.

```

DOCBED  FILELIST A0  V 108  Trunc=108  Size=126  Line=1  Col=1  Alt=0
Cmd  Filename  Filetype  Fm  Format  Lrecl  Records  Blocks  Date  Time
BUZZ  EXEC  A2  V  70  36  1  8/17/92  17:54:44  DSK191
C_ATTACH  EXEC  A1  F  80  2  1  1/30/92  16:48:41  DSK191
CRITBETA  EXEC  A1  V  19  4  1  1/21/94  11:23:18  DSK191
DIR  EXEC  A1  V  24  1  1  1/29/92  16:27:20  DSK191
DOWNLOAD  EXEC  A1  V  73  31  1  2/02/92  11:01:21  DSK191
HTRACK  EXEC  A1  V  24  2  1  7/27/92  13:17:05  DSK191
INFONT  EXEC  A1  V  19  2  1  12/10/92  13:30:28  DSK191
IOBETA  EXEC  A1  V  19  4  1  1/07/94  12:34:47  DSK191
MBAILEY  EXEC  A1  V  22  1  1  1/30/92  16:49:02  DSK191
PROFILE  EXEC  A1  V  29  8  1  1/16/94  12:47:00  DSK191
TEST1  EXEC  A1  V  27  9  1  2/02/92  10:53:40  DSK191
TEST4  EXEC  A1  V  18  8  1  8/21/92  16:20:19  DSK191
TUTOR  EXEC  A1  V  7  1  1  9/23/93  16:16:57  DSK191
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

```

Figure 21.7 FILELIST screen listing EXEC files.

As with RDRLIST, you enter a command that acts on a file in the Cmd column on the line listing that file. Unlike the commands you enter at the Ready ; prompt, you don't need to indicate the file that you want to act on, because FILELIST can tell from your cursor's position. Instead, enter the slash character where you would have put the file's name.

21.4.1 Copying Files

To make a copy of a file and call it TEST5 EXEC A, enter

```
COPY / TEST5 EXEC A
```

right on the line with the file you want to copy. It doesn't matter whether you enter it in upper or lower case. Figure 21.8 shows a FILELIST screen with this command entered.

```

DOCBED  FILELIST A0  V 108  Trunc=108  Size=126  Line=1  Col=1  Alt=0
Cmd  Filename  Filetype  Fm  Format  Lrecl  Records  Blocks  Date  Time
BUZZ  EXEC  A2  V  70  36  1  8/17/92  17:54:44  DSK191
C_ATTACH  EXEC  A1  F  80  2  1  1/30/92  16:48:41  DSK191

```

```

    CRITBETA EXEC      A1 V      19          4          1  1/21/94 11:23:18 DSK191
copy / test5 exec a  A1 V      24          1          1  1/29/92 16:27:20 DSK191
    DOWNLOAD EXEC      A1 V      73         31          1  2/02/92 11:01:21 DSK191
    HTRACK  EXEC      A1 V      24          2          1  7/27/92 13:17:05 DSK191
    INFONT  EXEC      A1 V      19          2          1 12/10/92 13:30:28 DSK191
    IOBETA  EXEC      A1 V      19          4          1  1/07/94 12:34:47 DSK191
    MBAILEY EXEC      A1 V      22          1          1  1/30/92 16:49:02 DSK191
    PROFILE EXEC      A1 V      29          8          1  1/16/94 12:47:00 DSK191
    TEST1   EXEC      A1 V      27          9          1  2/02/92 10:53:40 DSK191
    TEST4   EXEC      A1 V      18          8          1  8/21/92 16:20:19 DSK191
    TUTOR   EXEC      A1 V       7          1          1  9/23/93 16:16:57 DSK191
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

```

Figure 21.8 Copying a file using FILELIST.

When you press Enter, FILELIST executes the command. (Or commands—you can enter multiple commands on multiple lines before you press Enter.) Make sure to include the file's mode (as in Figure 21.8) or FILELIST will display an error message.

21.4.2 Renaming Files

Renaming is similar to copying. To rename a file as TEST5 EXEC A, enter the command

```
RENAME / TEST5 EXEC A
```

beginning in the Cmd column on the line with the file you want to copy. After you press Enter, FILELIST displays the message

```
** Discarded, Renamed, or Relocated **
```

on that line.

21.4.3 Deleting Files

Deleting a file is like deleting it from the reader: enter the DISCARD command, beginning in the Cmd column, on the line with the file you want to delete. After you press Enter, FILELIST displays the same message that it displays when you rename a file, showing you that the file no longer exists.

21.4.4 Displaying A Text File's Contents

Displaying a file's contents in FILELIST is similar to doing it from the Ready; prompt, except that you don't have to enter the file ID. Move your cursor to the Cmd column next to the file whose contents you want to view, enter the TYPE command, and press Enter. Use the Clear key and HT command just as you would when using the TYPE command from the Ready; prompt. When you are done, you will be back at the FILELIST screen.

21.4.5 Editing a File

Note how the bottom of the FILELIST screen shows that pressing PF11 invokes XEDIT. To edit a file whose name is displayed by the FILELIST program, move your cursor to that file's line and press PF11. When you finish editing it, CMS returns you to the FILELIST screen.

21.4.6 Printing a File

By now, you can probably guess how to do this: move your cursor to the Cmd column next to the name of the file you want to print, enter PRINT, and press Enter. This has the same effect as when you enter

```
PRINT filename filetype
```

at the Ready; prompt.

21.5 A Sample VM/CMS Session

One morning you log on and the system displays the following message, which shows that your reader has two files:

```
FILES: 0002 RDR, NO PRT, NO PUN
```

You enter RDRLIST at the command prompt, and the RDRLIST program shows you information about the two files in your reader, as shown in Figure 21.9.

```

JOEUSER RDRLIST A0 V 108 Trunc=108 Size=2 Line=1 Col=1 Alt=0
Cmd  Filename Filetype Class User at Node Hold Records Date Time
ACCLN NOTE PUN A ACCLN JUPITER NONE 4 10/16 13:01:44
SALES DATA PUN A ACCLN JUPITER NONE 148 10/16 13:02:45

```

Figure 21.9 Sample reader contents.

It looks like the user with user ID ACCLLEN (your good pal Larry in accounting) sent you a note and a medium-sized file. You move your cursor to the ACCLLEN NOTE file and press PF11 to peek at the file, and RDRLIST displays the screen shown in Figure 21.10.

```
0023      PEEK      A0  V 132  Trunc=132 Size=12 Line=0 Col=1 Alt=0
Note from ACCLLEN at JUPITER Format is NETDATA
* * * Top of File * * *
Date: 16 October 94, 13:01:44 EST
From: Larry Niven          212/930-3342 x3277  ACCLLEN  at Jupiter
To:  JOEUSER

Joe -

I'm sending the SALES DATA file to your reader for you to look over.  Don't
read the whole thing (you probably won't have time) but look over the
monthly totals before the meeting Monday.  Sorry to do this to you on a
Friday!

- Larry
* * * End of File * * *
```

Figure 21.10 Sample message to you from Larry.

So now you know what that other file is. You save the note from Larry by entering RECEIVE at the PEEK screen's command line. The system returns you to the RDRLIST screen, which shows you that the ACCLLEN NOTE file has been "added to ALL NOTEBOOK A0." Now you know the name of the file in which to find this note if you need to refer to it again.

While viewing the RDRLIST screen, you pull the SALES DATA file out of the reader by moving your cursor to that line and pressing PF9. Then, you quit the RDRLIST program by pressing PF3.

Next, you take a quick look at this SALES DATA file that Larry mentioned. You enter

```
TYPE SALES DATA
```

at the Ready; prompt and see the screen shown in Figure 21.11.

```
TYPE SALES DATA

Page 01                      O'Rourke Enterprises          10/16/94
                          Monthly Sales Figures by Region
```

Chapter 21 Using a VM/CMS System

JANUARY	Gross	Net
Northeast	77,423	12,756
Mid-Atlantic	65,812	11,447
Southeast	58,222	8,800
Southwest	42,342	5,663
Midwest	73,242	12,000
Northwest	14,433	1,500
West Coast	68,443	7,445
JANUARY TOTAL	399,917	59,611
FEBRUARY	Gross	Net
Northeast	75,732	11,890
Mid-Atlantic	69,348	11,008

MORE . . .

Figure 21.11 Beginning of SALES DATA file.

You press the Clear key, and see the screen shown in Figure 21.12.

Southeast	61,835	8,890
Southwest	42,445	5,556
Midwest	70,762	12,934
Northwest	14,242	1,634
West Coast	65,447	7,556
FEBRUARY TOTAL	399,811	39,468
MARCH	Gross	Net
Northeast	76,788	12,020
Mid-Atlantic	67,789	11,003
Southeast	62,640	9,126
Southwest	40,400	4,900
Midwest	69,730	11,500
Northwest	13,554	1,330
West Coast	73,354	8,020
MARCH TOTAL	399,811	59,468
APRIL	Gross	Net

MORE . . .

Figure 21.12 Second screen of SALES DATA screen.

You get the pattern, and don't need to do this for the whole file, so you type HT to Halt the Typing, press Enter so that the HT takes effect, and then press Clear one more time.

If Larry said that you should only bother with the monthly totals, it might be a good idea to consolidate those figures by making a copy of the SALES DATA file and editing out the details. You enter

```
COPY SALES DATA A TOTALS DATA A
```

and then call up the TOTALS DATA file with XEDIT:

```
X TOTALS DATA
```

With the beginning of the file displayed in XEDIT, you enter the XEDIT prefix command dd at the beginning and end of the first block to get rid of in the file, as shown in Figure 21.13.

```
SALES DATA A1 F 80 Trunc=80 Size=148 Line=0 Col=1 Alt=0
===== * * * Top of File * * *
===== Page 01 O'Rourke Enterprises 10/16/94
===== Monthly Sales Figures by Region
=====
=====
=====dd== JANUARY Gross Net
=====
===== Northeast 77,423 12,756
===== Mid-Atlantic 65,812 11,447
===== Southeast 58,222 8,800
===== Southwest 42,342 5,663
===== Midwest 73,242 12,000
===== Northwest 14,433 1,500
===== West Coast 68,443 7,445
=====dd
===== JANUARY TOTAL 399,917 59,611
=====
===== FEBRUARY Gross Net
=====
===== Northeast 75,732 11,890
=====>
X E D I T 1 File
```

Figure 21.13 Deleting a block from the SALES DATA file.

After pressing Enter, you indicate the next block to get rid of with the same command, as shown in Figure 21.14.

```

SALES      DATA      A1 F 80  Trunc=80 Size=139 Line=0 Col=1 Alt=1
===== * * * Top of File * * *
===== Page 01                      O'Rourke Enterprises          10/16/94
===== Monthly Sales Figures by Region
=====
===== JANUARY TOTAL                399,917  59,611
==dd==
===== FEBRUARY                      Gross      Net
===== -----
===== Northeast                    75,732   11,890
===== Mid-Atlantic                 69,348   11,008
===== Southeast                   61,835    8,890
===== Southwest                    42,445    5,556
===== Midwest                     70,762   12,934
===== Northwest                   14,242    1,634
===== West Coast                   65,447    7,556
====dd
===== FEBRUARY TOTAL                399,811  39,468
=====
===== MARCH                          Gross      Net
=====>
                                           X E D I T  1 File

```

Figure 21.14 Deleting another block from the SALES DATA file.

You press Enter and continue this until it is a one-page file that only shows the header and the monthly totals. As you do this, you periodically enter XEDIT's SAVE command; when you are finished, you enter SAVE one last time and quit out of XEDIT with the QUIT command.

To print, you enter

```
PRINT TOTALS DATA
```

at the Ready ; prompt. It's a short file, so a minute later the system displays a message that tells you that it has printed your file. You're done with your CMS session for now, so you enter

```
LOGOFF
```

to log off.