z/VM
7.2

CMS Primer

IBM

> **Note:**
>
> Before you use this information and the product it supports, read the information in "Notices" on page 129.

Contents

Figures

Tables

x

About This Document

This document provides an introduction to the IBM® z/VM® conversational monitor system (CMS). The objective of this primer is to teach you how to use some of the basic CMS functions so you can get started doing productive work on z/VM.

The tasks that you will learn include:

- Signing on to the system
- Creating and changing files
- Managing, organizing, and sharing your files
- Printing files
- Sending messages, notes, and files to other users
- Asking for information from the system.

This primer presents some of the functions and commands available with CMS and z/VM. Some topics include pointers to other documents that contain additional information.

Intended Audience

Intended Audience

This information is intended for anyone who needs an introduction to basic CMS and z/VM functions.

We have attempted to present the material in a way that will interest you and enable you to be successful. But the best way to learn the system is to use it. To this end, we suggest that you do the exercises, which are woven throughout the text.

Conventions Used in This Document

The examples in this document have been created for an 80-column, IBM 3270-compatible terminal. However, the keys on your keyboard might be different. See Appendix A, "Keyboard Key/Function Conversions," on page 127.

In the examples, *variables* are shown in a different font. A variable is a word or sequence of characters that is not a literal value but a representation of the value that will be displayed or that you should type. For example:

```
filename
```

is a variable that represents the actual name of a file.

Where to Find More Information

For more information about z/VM topics, see "Bibliography" on page 133.

Links to Other Documents and Websites

The PDF version of this document contains links to other documents and websites. A link from this document to another document works only when both documents are in the same directory or database, and a link to a website works only if you have access to the Internet. A document link is to a specific edition. If a new edition of a linked document has been published since the publication of this document, the linked document might not be the latest edition.

How to Send Your Comments to IBM

We appreciate your input on this publication. Feel free to comment on the clarity, accuracy, and completeness of the information or give us any other feedback that you might have.

To send us your comments, go to z/VM Reader's Comment Form (www.ibm.com/systems/campaignmail/z/zvm/zvm-comments) and complete the form.

If You Have a Technical Problem

Do not use the feedback method. Instead, do one of the following:

- Contact your IBM service representative.
- Contact IBM technical support.
- See IBM: z/VM Support Resources (www.ibm.com/vm/service).
- Go to IBM Support Portal (www.ibm.com/support/entry/portal/Overview).

Summary of Changes for z/VM CMS Primer

This information includes terminology, maintenance, and editorial changes. Technical changes or additions to the text and illustrations for the current edition are indicated by a vertical line to the left of the change.

SC24-6265-01, z/VM 7.2 (December 2021)

This edition includes maintenance updates.

- In file editing examples, "West Germany" is updated to "Germany", "Peking" is updated to "Beijing".

SC24-6265-01, z/VM 7.2 (September 2020)

This edition supports the general availability of z/VM 7.2.

SC24-6265-00, z/VM 7.1 (September 2018)

This edition supports the general availability of z/VM 7.1.

Chapter 1. Getting Started

The Conversational Monitor System (CMS) is the part of the z/VM system that provides the interactive user environment, the file systems, the application programming platform, and various tools. To get the full use of z/VM, you need to know how to use CMS.

This chapter tells you how to log on to the z/VM system and tells you about the z/VM system status notices.

A Note about Keyboards

The z/VM user interface is based on the IBM 3270 terminal keyboard. Current workstation and laptop keyboards are somewhat different from the 3270 keyboard. Appendix A, "Keyboard Key/Function Conversions," on page 127 contains a chart that shows the main differences between key appearances and key combinations required to perform common functions.

Logging On and Logging Off

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

- Your user identification (user ID)
- Your password
- Whether your system has **automatic IPL**. (We will explain this later.)

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

Your terminal should display a **logo** on the screen similar to this:

```
                    /  VV            VVV  MM         MM
                   /   VV           VVV   MMM       MMM
        ZZZZZZ    /    VV          VVV    MMMM     MMMM
           ZZ    /     VV      VVV        MM MM MM MM MM
          ZZ    /      VV  VVV           MM   MMM   MM
         ZZ    /       VVVVV             MM    M    MM
        ZZ    /        VVV              MM         MM
       ZZZZZZ /        V              MM         MM

              built on IBM Virtualization Technology



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

       USERID    ===>  _
       PASSWORD  ===>


       COMMAND   ===>
                                                  Running System
```

Move the cursor so it is next to USERID ===>. Use the Tab key. (The system leaves one space between USERID ===> and the cursor.)

Now let us get started:

1. Type your user ID. You can use either capital or lowercase letters, or a combination of the two. If your user ID is less than eight characters long, press the Tab key or the New-Line key to get to the next line.

2. Type your password. You can use either capital or lowercase letters, or a combination of the two. In some systems, you will not see your password on the screen as you type it. This is a security measure. It prevents other people from learning your password and using your system without authorization.

Now press the Enter key. When you press Enter, both your user ID and password are sent to the system.

What If I Make a Mistake in My User ID or Password?

If you made a mistake in either your user ID or your password, you will receive this error message:

```
LOGON unsuccessful--incorrect userid and/or password.
```

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

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

```
Enter one of the following commands:

  LOGON userid              (Example:  LOGON VMUSER1)
  DIAL userid               (Example:  DIAL VMUSER2)
  MSG userid message        (Example:  MSG VMUSER2 GOOD MORNING)
  LOGOFF
```

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

```
logon userid
```

For example, if your user ID is JONES, you would type:

```
logon jones
```

Note: You can type commands in either uppercase or lowercase letters, or a combination of the two.

After typing the LOGON command, press Enter. When you press Enter, the command you type is sent to the system.

z/VM responds with this message:

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

Type your password and press Enter.

If you make another typing mistake while entering either your user ID or password, you will receive one of the two error messages listed previously and you must start over again.

Note: If you enter an incorrect password several times, the system might lock out your user ID and refuse to log you on even if you enter the correct password. Then you will have to ask your system operator to unlock your user ID.

You Have Successfully Entered Your User ID and Password

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

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

> If your system does not have automatic IPL, you need to enter another command. Type the following, then press Enter.
>
> ```
>         ipl cms
> ```
>
> You need to enter the IPL CMS command only once during a terminal session, but it is an important command. IPL stands for Initial Program Load. You will need to use the IPL CMS command after you receive the logon message whenever you want to create files. Now press Enter. z/VM responds by repeating the command in capital letters.

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

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

This is called the *ready message*. It tells you that z/VM is ready to continue. The T stands for time. The first series of numbers tells how much time it took z/VM to do the last task you gave it. The second set of numbers gives you the time of day.

Where Will My Work Be Stored?

Before you begin your work, there is one more thing you need to know—where the work you do will be stored. In z/VM, your data can be stored in the Shared File System (SFS) or on a minidisk.

If you have **SFS** storage, your system administrator has enrolled you in a file pool. A **file pool** is an amount of storage that is set aside to contain your work and other users' work. If your files are stored on a **minidisk**, you have an amount of storage in the computer set aside specifically for your work.

Depending on how your system administrator sets up your system, you might have SFS storage, minidisk storage, or the capability to use both types of storage. To determine whether your files are stored in an SFS file pool or on a minidisk, enter:

```
query accessed
```

The output you see will help you determine what kind of storage you have.

Shared File System Storage

If your system administrator has set up SFS storage for you, your output will look something like this:

```
Mode   Stat  Files  Vdev  Label/Directory
A      R/W       0  DIR   VMSYSU:yourid.
S      R/O    1321  190   MNT190
Y/S    R/O     337  19E   MNT19E
```

You can tell you have SFS storage because the first line (the line for Mode A) shows

```
 DIR
```

in the Vdev column. DIR stands for DIRECTORY. (In Chapter 3, "Managing Your Files," on page 53, we will discuss directories and explain how they are used to store your files.)

You will also notice that the `Label/Directory` column shows:

```
 VMSYSU:yourid.
```

VMSYSU is the file pool ID. This is the name of an IBM-supplied file pool that was shipped with your system. If your file pool ID is different, you will still be able to complete the examples in this document; in later commands, substitute your file pool ID for VMSYSU. The file pool ID is followed by your user ID (shown here as *yourid*) and a period.

If you have SFS storage, you can take advantage of the organization and sharing functions of SFS. In Chapter 3, "Managing Your Files," on page 53, we will discuss SFS commands and show you how to use them.

Minidisk Storage

If your system administrator has set up minidisk storage for you, your output will look something like this:

```
Mode   Stat  Files  Vdev  Label/Directory
A       R/W      0   191   DAB191
S       R/O   1321   190   MNT190
Y/S     R/O    337   19E   MNT19E
```

You can tell you have minidisk storage because the first line (the line for Mode A) shows

```
191
```

in the Vdev column. 191 is the ***address*** of the minidisk. It shows where the minidisk is located within your computer system.

You will also notice that the first line in the `Label/Directory` column shows something like this:

```
DAB191
```

DAB191 is a minidisk label. Your output will actually show a label of up to six characters that identifies your 191 minidisk to the system.

Even though your system administrator has set up minidisk storage as your primary storage, you might also have space in a file pool and might be able to use SFS commands. In Chapter 3, "Managing Your Files," on page 53, we will discuss these commands and show you how to use them.

What Do I Do Next?

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

```
logoff
```

Press Enter. z/VM displays the time you logged off plus other accounting information. It also displays this message:

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

Now you can turn off your terminal. Or, if you want to start another terminal session, press Enter. The z/VM logo will appear again, and you can log on.

Here is a summary of the logon procedure:

Turn on your display to get the logo.
Type your user ID next to USERID ===→
Type your password next to PASSWORD ===→
Press Enter.

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

Without automatic IPL:
You type: `ipl cms`
Press Enter. When VM READ appears, press Enter again.

To log off:

You type: `logoff`
Press Enter.

How to Tell the System to Stop Displaying or Processing

There might be times when you type a command and then change your mind. For example, you might have requested that a document be printed then realize that it is the wrong document. Or, maybe the system gets into what we call a loop — displaying the same information each time you clear the screen. Use the following commands to ***halt*** either displaying or processing:

```
HT (means halt typing — displaying)
HX (means halt execution — processing)
```

After you enter either HT or HX, you will see a few letters on your screen (either CMS, or whatever name your system uses for CMS). Then, type the following command and press Enter:

```
b
```

The b is short for BEGIN—it makes the system begin again and gives you a ready message.

Status Notices

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

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

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

b (short for BEGIN)

and press Enter.

VM READ
This notice means that CMS is waiting for you to input text or enter a command.

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

MORE...
This means that there is more information than can fit on the screen. After a short period of time, the next screen of information is displayed. To see the next screen right away, press CLEAR. To hold this information on the screen, press Enter. This changes the MORE... notice to HOLDING.

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

NOT ACCEPTED
This means that the system is working on something and is too busy to accept another command from you. Wait several seconds and enter your command again.

Note
There are commands which allow you to alter the function of the MORE... and HOLDING notices, called CP TERMINAL MORE and CP TERMINAL HOLD. These commands will not be discussed in this document. For additional information see the *z/VM: CP Commands and Utilities Reference*.

Note
If, after entering a command, you receive the message ``` File pool VMSYSU is unavailable or unknown ``` you might be temporarily unable to continue because of a system problem. Wait a few minutes, then type IPL CMS and press Enter twice to restart. If you receive the following message: ```   DMSACC112S A(191) device error ``` this means that a disk (A or 191 disk) has not been prepared yet for CMS files. Your z/VM administrator will have to format the new minidisk according to the instructions in the Formatting Minidisks section in *z/VM: CMS User's Guide*.

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

Chapter 2. Editing Files

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

Creating Files

What Is a File?

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

One thing you need to know about creating files with z/VM is that each file needs its own three-part identifier. The first part of the identifier is the *file name*. The second part is the *file type*. And the third part is the *file mode*. These three file identifiers are often abbreviated *fn ft fm*.

The file name and the file type can be any combination of letters and numbers. Each can be no longer than eight characters. (Certain special characters can also be used, but we will use only letters and numbers in our examples.)

You will probably want to give your files identifiers that describe their contents and are easy to remember. For example, if you wanted to create a file containing information about animals that were in danger of becoming extinct, you might use the file name *rare*. For the file type, you might use the word *animals*.

The file mode is always a letter followed by a number. (You can ignore the number that appears after the file mode letter for the time being.) In this chapter, we do not specify the file mode. Therefore, the file mode for all the examples defaults to *A*. So your file about endangered animals would have the identifier *rare animals a.*

If you wanted to create another file about endangered animals, you could use a different file identifier. For example, your new file could be called *endanger animals* or *rare beasts.*

Here is another important thing you need to know about creating files and using the z/VM system. **When you type commands, be sure to type them as they are shown in the text, including spaces if there are any. If a space is missing or if a word is misspelled, the system will not recognize the command.** If the command has a number 1 in it, be sure to use the number 1 key instead of a small letter L. **Commands can be typed in capital or lowercase letters, or a combination of the two.**

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

```
GET filename filetype filemode
```

The word GET must be typed exactly as shown, although you can use capital or lowercase letters. Because the file name, file type, and file mode are shown in all lowercase italicized letters, you would substitute your own three-part identifier in their places. You might type your command like this:

```
get rare animals a
```

Some command names can be abbreviated. We will show the minimum abbreviation in capital letters. For example:

```
Input
```

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

The last thing you need to know before you create your first file is that you must have followed the logon procedure outlined in the previous chapter. After logging on, you must enter the IPL CMS command (if you do not have automatic IPL).

If you forget to enter IPL CMS and try to enter other commands, you will receive an error message that begins with

```
Unknown CP command
```

Creating Your First File

One way to create a file is by using the editor, called by entering the XEDIT command. The format of the XEDIT command is:

```
Xedit filename filetype filemode
```

If the file *fn ft fm* already exists, the XEDIT command begins an **editing session** with this file. An editing session is the time between entering an XEDIT command and entering a command (discussed later) to let the editor know you are finished working on the file.

If the file did not exist before, XEDIT will create one and start an editing session with it.

Let us create a simple file. It will contain information about things that are found at picnics. The file name is **picnic.** The file type is **ingredts**. (We cannot use **ingredients** because it has more than eight letters, so we will shorten it to **ingredts**). The file mode is **A** followed by a number *n*, but you do not have to type it. If you leave out the file mode, it is assumed to be **A**n. Your cursor should be at the bottom left of the screen.

Type:

```
X picnic ingredts
```

Press Enter. Your screen now looks like this:

```
 PICNIC   INGREDTS A1  F 80  Trunc=80 Size=0 Line=0 Col=1 Alt=0  1
Creating new file:   2



                                          6



    5
===== * * * Top of File * * *  7
 |...+....1....+....2....+....3....+....4....+....5....+....6....+....7...  8
===== * * * End of File * * *



                            6



====>  3
                                        4   X E D I T  1 File
```

Note: If XEDIT does not work as described in this document, it is probably because someone in your computer center has defined an *XEDIT profile* for you. (We will be discussing XEDIT profiles in"Creating an XEDIT Profile" on page 50.) You should ask your supervisor or z/VM administrator how to avoid using that profile.

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

1

File Identification Line

The first line on the screen gives you the three-part file identifier — file name, file type, and file mode. F 80 means that each record of the file is fixed at 80 characters. Trunc means **truncation column**. Trunc=80 means that if you type more than 80 characters in a line, the additional characters are rejected by the system. Size=0 is the number of lines in the file. Because we have not entered any data yet, the file contains zero lines. Line=0 is the file line number of the current line. This is explained in number 7, which follows. Alt=0 is the alteration count, which is explained in "Protecting Your Files" on page 45.

2

Message Line

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

3

Large Arrow Command Line

The large arrow (====>) at the bottom of the screen points to the XEDIT command line. One of the ways you communicate with the editor is by entering XEDIT commands on this line. XEDIT commands can be typed in either uppercase or lowercase, or a combination of both, and many can be abbreviated. For example, **INPUT**, **Input**, and **i** are all valid ways to type the INPUT command.

After typing a command on this line, **you must press Enter** to send the command to the editor. When you press Enter, the command disappears from the command line and is **processed** or carried out.

4

Status Area

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

5

Prefix Area

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

6

File Area

The rest of the screen is available to display the lines in the file. The file lines are inserted between the ***Top of File*** and ***End of File*** notices you now see on the screen.

7

Current Line

The current line is the file line in the middle of the screen (above the scale). It appears brighter than the other file lines or is **highlighted**. In the sample screen, the current line is the Top of File line.

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

8

Scale

The scale appears under the current line to assist you in editing. It is like the margin scale on a typewriter. If your screen does not show a scale, type SET SCALE ON at the command line and press Enter.

Entering Data

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

```
 Input
```

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

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

Notice how the screen changes in input mode.

```
 PICNIC   INGREDTS A1  F 80  Trunc=80 Size=9 Line=0 Col=1 Alt=0
Input mode:




* * * Top of File * * *
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7...+....
_  2



               1


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

In input mode, the prefix areas (=====) disappear. The message line and status area display the notice, Input mode. You type your data in the *input zone.* The input zone **1** is the bottom half of the screen, between the scale and the command line.

The command line contains the notice ∗∗∗Input Zone∗∗∗ because *you cannot enter commands* from the command line while using input mode. The notice marks the end of the input zone and reminds you that you cannot enter commands. You have to go back to edit mode to enter commands. We will show you how later.

The cursor **2** is placed automatically on the first line in the input zone. This is where you should begin typing the data. After you have typed each line, press the new-line key to return the cursor to the beginning of the next line. You can use one of the other cursor-movement keys, but the new-line key is faster.

Now type the following input:

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

Press Enter. Your screen should look like this:

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




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

_




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

The lines that you typed moved to the top half of the screen. The last line you typed is the new current line (THOU). After you press Enter, the editor changes all the letters you typed to capitals. Later, you will learn how to tell the editor not to change them to capitals. If you had more data to type, you could continue typing in the input zone (between the current line and the command line).

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

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




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




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

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

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

We are finished working with this file, so we must tell the editor what to do with it. The editor created your file in **_virtual storage._** Virtual storage is temporary storage. If you want to keep the file, you must enter a FILE command. It tells the editor to put the file in the permanent storage your system administrator

allocated for you, either in a Shared File System (SFS) file pool or on a minidisk. The format of the FILE command is:

```
FILE
```

Now type the command to file PICNIC INGREDTS:

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

Press Enter. The file disappears from your screen, but the editor saves it in permanent storage. Because you did not specify another file name or file type, the file is saved as PICNIC INGREDTS. Because you did not specify a file mode, the file mode is assumed to be **A**.

Now, you will see a ready message (Ready;). It tells you how much computer time you used in creating this file. It also tells you that this editing session is over and z/VM is ready to do more work for you. Notice that the cursor has returned to a blank line at the bottom of the screen. The blank line is called the **CMS command line**. That is where you would type the command to create another file.

Note: By now, you should be used to pressing Enter after you type a command on the command line. We will not remind you to press Enter when we ask you to **enter** a command.

Creating a Second File

Let us create another simple file. It will be about crime. The file name will be **crime.** The file type will be **theft.** Because we are not going to specify a file mode, the file mode will be **A.**

Enter:

```
X crime theft
```

Your screen should look like this:

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




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




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

Capital and Lowercase Letters

In your first file, all the letters you typed were changed to capitals when they were entered in the file. In this file, we want the letters to be entered in the file exactly the way we type them. You can use the SET CASE command to tell the editor to use mixed case, that is, both capital and lowercase letters, when the data is entered in the file. Its format is:

```
SET CASE Mixed
```

Enter:

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

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

====> i (short for INPUT)

Type the following lines in the input zone:

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

Press Enter *twice* to return to edit mode. Your screen should look like this:

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



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




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

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

All the files we use to practice editing are short. Your real files will probably be longer. If you fill up the input zone but want to continue typing data in input mode, press Enter one time. The data moves up on the screen, and you can continue typing in the cleared input zone. When the input zone is filled up again, press Enter once again. You can stay in input mode and continue typing if you need to. When you are ready to return to edit mode, then press Enter twice.

Splitting and Joining Lines

To split or join lines, you can press a PF key instead of typing a command in the command line and then pressing Enter. There is nothing magical about PF keys. During an editing session, each one is set to an XEDIT command. (When you are not editing a file, the PF keys are set to different functions, but we will talk more about this later.) You do not press Enter when you use PF keys. The command is sent to the editor automatically when you press the PF key.

Important Note: In this document, we will be talking about PF keys 1 through 12. If your terminal has 24 PF keys, PF keys 13 to 24 are used like PF keys 1 to 12 as discussed here.

To split a line in two:

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

• *Press PF11.*

Let us split the first line in this file. Use the cursor movement keys to place the cursor under the *S* in *Stole*, like this:

```
===== Tom, Tom, the piper's son, Stole a pig and away he run.
                                 ↑
                                 └── cursor
```

Press PF11. Now you have two lines:

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

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

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

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

To join two short lines:

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

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

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

The result is:

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

Now, move the cursor to the command line and enter the command to file your work:

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

Setting Tabs

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

Let us create a file that needs tabs. The file will be a list of shortwave radio frequencies. Its file name is *radio.* Its file type is **waves.** The file mode is **A.**

Enter:

```
x radio waves
```

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

```
Query TABS
```

Now enter:

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

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

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

```
SET TABS n1 n2 n3 n4
```

You substitute the column numbers where you want the tabs set for *n1, n2,* and so forth.

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

Enter:

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

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

Enter:

====> i (short for INPUT)

In the first line of the input zone, type:

```
voice of america
```

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

Type:

```
6125
```

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

Type:

```
australia
```

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

Then type:

```
6035
```

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

```
greece
```

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

```
9615
```

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

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




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






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

Enter the command that tells the editor to file your work:

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

Let us create another file that uses tab settings. If you were a stockroom manager, you might want an inventory file that lists part numbers, part descriptions, locations, and quantities. Use the file name **stock,** the file type **inven,** and the file mode **A.**

Enter:

```
x stock inven
```

Now you are ready to set the tabs. Enter:

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

Enter:

```
====>  i     (short for INPUT)
```

Using PF4 for tabbing, type:

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

Press Enter twice to return to edit mode.

Displaying PF Key Settings

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

```
====> query pf
```

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

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

```
====> query pf10
```

The PF key setting is displayed in the message line.

Now enter the command to save this file:

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

Inserting Words in a Line

Let us create one last file. Its file identifier will be **vacation camping a.**

Enter:

```
x vacation camping
```

Enter the command that puts you in input mode:

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

Type, with the error:

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

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

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

- Press the **PA2 key.** (Do not confuse this with the PF2 key.) When characters are inserted in a line, the other letters move over to make room for them. The PA2 key makes it possible for the characters to move over. The PA2 key may have a different identifier, depending on your terminal. If you do not have a key labeled PA2, ask your supervisor to show you its equivalent.
- Then move your cursor to the position in the line where you would like to insert one or more characters.
- Then press the **insert mode key.** The insert mode key is labeled either with INS MODE or a special character.
- Then you would be in insert mode, and you can type the characters to be inserted.
- Press **RESET** to type normally again.

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

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

```
wood nearby forcooking
```

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

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

```
no wil bears
```

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

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

```
nothing to fear
```

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

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

```
soothing surroundings forthe eye
```

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

```
nothing to ofend the ear
```

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

```
no early rises camped nearby
```

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

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

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

Now press Enter twice to get out of input mode. Then enter:

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

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

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

```
log
```

Summary

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

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

Changing Files (Part One)

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

Displaying a File on Your Screen

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

```
XEDIT filename filetype filemode
```

If the file mode is **A**, you do not have to type it because **A** is the default.

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

I Do Not Remember the File Identifier

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

```
FILEList
```

Now enter this command on the CMS command line.

```
filel
```

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

```
 yourid    FILELIST A0  V 108  Trunc=108 Size=19 Line=1 Col=1 Alt=0
Directory = VMSYSU:yourid.
Cmd    Filename Filetype Fm Format Lrecl    Records    Blocks    Date     Time
       VACATION CAMPING  A1 F         80         8         1  4/14/90 10:02:43
       STOCK    INVEN    A1 F         80         7         1  4/14/90  9:57:11
_  1   RADIO    WAVES    A1 F         80         3         1  4/14/90  9:53:14
       CRIME    THEFT    A1 F         80         4         1  4/14/90  9:50:54
       PICNIC   INGREDTS A1 F         80         3         1  4/14/90  9:48:03
         .
         .
         .
         .
         .




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

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

Note that the screen shown here appears if your files are stored in an SFS file pool. If your files are stored on a minidisk, your FILELIST screen will **not** contain the following line:

```
Directory = VMSYSU:yourid.
```

Also, some of the PF keys shown at the bottom of this screen will perform different functions if your files are stored on a minidisk.

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

- Move the **cursor** 1 to the line describing the file you want to edit.
- Then **press PF11** (which is set to the XEDIT command). This does exactly what entering the XEDIT command from the command line does. The file appears on the screen, replacing the FILELIST screen.

A more complete description of the FILELIST screen is in Chapter 3, "Managing Your Files," on page 53. In this chapter, we are using FILELIST only to list the possible files available for editing or copying.

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

- Move your cursor to the line that has RADIO WAVES.
- Press PF11.

The RADIO WAVES file appears on your screen.

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

Enter:

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

Adding Lines

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

Another way is to use a *prefix command* that adds a blank line. It is called a prefix command because you type it in the prefix area, not on the large arrow command line. After the blank line is added, you type the new data in that line.

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

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

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

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

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

Press Enter. A blank line is added.

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

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

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

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

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

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

Use PF4 for tabbing, and type:

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

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

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

Type:

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

Moving through a File

If you were to continue adding lines, eventually this file would not fit on one screen. To look at each screen of the file, you use commands that *scroll* the file. Scrolling is like turning the pages of a book.

Moving Forward or Backward

Two PF keys are set to commands that scroll the file. PF8 scrolls forward one screen. PF7 scrolls backward one screen.

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

Moving to the Bottom or Top

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

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

```
Bottom
```

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

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

====> b (short for BOTTOM)

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

Now enter:

====> i (short for INPUT)

Using PF4 for tabbing, type:

```
spain           6090
beijing         9780
moscow          9450
```

Now press Enter twice to return to edit mode.

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

```
TOP
```

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

Enter:

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

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

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

Then type:

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

Moving Down or Up

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

Use the DOWN command to move the display one or more lines toward the end of a file. It is entered in the large arrow command line and has the format:

```
Down n
```

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

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

```
Up n
```

You substitute the number of lines you want to move up for *n*. An UP command changes which line in the file is the current line on your screen.

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

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

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

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

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

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

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

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




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

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

Enter:

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

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

More Practice in Adding Lines

The next file you are going to change is the one that lists the ingredients for a picnic. Find that file on the FILELIST screen. Move the cursor to that line, and press PF11.

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

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




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




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

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

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

In the blank line, type:

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

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

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

Type

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

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

Then enter:

====> b (short for BOTTOM)

Enter:

====> i (short for INPUT)

Type these lines:

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

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

Enter:

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

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

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

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

Deleting Lines

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

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

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

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

When you press Enter, the two lines are deleted.

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

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

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

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

The rest of this section does not involve any terminal exercises for you to do, only examples of things that you can try later on. Because you are finished editing files for now, press PF3 to exit from FILELIST.

Deleting a Block of Lines

A **block** of lines is a group of consecutive lines. Sometimes you may want to delete an entire block of lines, but you do not want to bother counting how many.

To delete a block of lines, type the letters **dd** in the first line of the block. Then type **dd** on the last line of the block. Then press Enter. For example:

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

When you press Enter, all of the lines are deleted.

What if you want to delete a large section of a file? A block of lines does not have to be all on one screen. You can type **dd** on the first line of the block and then scroll the display until you find the last line of the block. Then you can type the second **dd** and press Enter. Remember how to scroll the file? Use PF8 to scroll forward, PF7 to scroll backward.

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

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

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

RECOVER Command

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

```
RECover n
```

You substitute the number of lines you want to recover for *n*. The last lines that you deleted are the first to be recovered. If, for example, you delete 10 lines and recover 5, the last 5 lines deleted are those recovered.

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

Duplicating Lines

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

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

When you press Enter, the file looks like this:

```
===== I want three more copies of this line.
===== I want three more copies of this line.
===== I want three more copies of this line.
```

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

Copying Lines

Suppose you have a table that appears in one part of a file, and you want it to appear again in another part of the same file. Do you have to retype it? No, you can use the **c prefix command.**

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

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

For example:

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

When you press Enter, file would look like this:

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

Copying More Than One Line

You can type a number before or after the **c** to copy more than one line. The following are all valid ways to type the **c** prefix command.

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

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

Copying a Block of Lines

To copy a block (a group of consecutive lines), type the letters **cc** in the prefix area of the first line of the block. Type **cc** in the prefix area of the last line of the block. Then type an **f** or **p**. This way, you do not need to count the number of lines you want copied. For example:

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

When you press Enter, a copy of the block of lines is placed before the line containing the **p**. (If the line contained an **f**, the block goes after that line.)

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

While you scroll the file, the status area displays:

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

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

Moving Lines

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

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

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

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

RESET Command

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

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

```
====> reset
```

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

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

Summary

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

Command/PF key	Function
FILEList	Displays a list of your files, from which you can call out a copy of a file for editing.
Bottom	Makes the last line of the file the new current line.
TOP	Makes the ***Top of File*** line the new current line.
Down	Moves the current line toward the end of a file.
Up	Moves the current line toward the top (beginning) of a file.
RECover	Puts back lines deleted by a *d* prefix command.
RESet	Cancels prefix commands when the status area displays a pending notice.
a (prefix)	Adds blank line(s).
/ (prefix)	Makes this line the new current line.

Command/PF key	Function
d (prefix)	Deletes a line(s).
" (prefix)	Duplicates a line(s).
c (prefix)	Copies a line(s). Must show destination with f or p.
m (prefix)	Moves a line(s). Must show destination with f or p.
f (prefix)	Moves/copies line(s) following this line.
p (prefix)	Moves/copies line(s) preceding this line.
PF7 (BACKWARD)	Scrolls backward one screen.
PF8 (FORWARD)	Scrolls forward one screen.

Changing Files (Part Two)

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

If you are not already logged on, use the logon procedure now (see "Logging On and Logging Off" on page 1).

LOCATE and CHANGE Commands

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

Now enter:

`filel` (short for FILELIST)

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

- Move the cursor to the line containing RADIO WAVES.
- Press PF11.

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

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

```
Locate /your data/
```

You must type a separator between the command name (LOCATE) and *your data*. In the previous example, a slash (/) is the separator, indicating the beginning and end of your character data. If the data to be located contains a slash, like VM/CMS or 06/04/90, you must use a different special character as a separator. For example,

```
locate $VM/CMS$
```

or

```
locate %06/04/90%
```

The following special characters cannot be used as separators: ¬ (not), + (plus), - (minus), or . (period).

Now enter:

====> L /havana/ (short for LOCATE /havana/)

Note: Generally, we use lowercase letters for examples of input commands, but the lowercase L, l, is so similar to the number one, 1, we chose to use a capital L. Remember either capital or lowercase letters are acceptable.

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

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

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

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

```
c ?6165?9525?
```

Regardless of the separator character you choose to use, be sure to use the same character throughout the command. For example, this command would not work:

```
c /6165?9525/
```

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

Now enter:

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

The editor changed the line to read HAVANA 9525. The message

```
1 occurrence(s) changed on 1 line(s)
```

tells you how many changes were made. The HAVANA entry is now in the current line. (Remember, the current line is in the middle of the screen and appears brighter than the other file lines.) What happens if you use the LOCATE /HAVANA/ command now? Let us see. Enter:

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

The following message is displayed:

```
Target not found
```

It means the editor could not find HAVANA. In other words, it could not find the target of the search. The editor started looking for HAVANA in the line *following* the current line. It then looked in each following line, until it reached the end of the file. Because HAVANA was in the current line, the editor did not find it. Notice that the command you typed is displayed again on the command line. This lets you see if you made a typing mistake. Note also that the End of File line is now the current line.

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

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

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

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

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

When you press Enter, the following message is displayed:

```
No line(s) changed
```

The editor looked in the current line (the Top of File line) and did not find the frequency *6125*.

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

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

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

Enter:

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

Let us put the file back into permanent storage.

Enter:

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

The revised file is now in permanent storage. The FILELIST display returns to the screen.

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

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

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

Enter:

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

Here is the crime file.

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




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


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

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

Then enter:

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

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

Changing a Word Throughout a File

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

Enter:

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

The message

```
 3 occurrence(s) changed on 2 line(s)
```

tells you how many words were changed. The editor searched every line of the file, from the current line to the last, and changed Tom to Mort. Note the format of this command. You must type a separator, which is a slash (/) in the previous example, after the new word (Mort), and you must type two asterisks ***with a space between them*** (* *). The first asterisk tells the editor to search every line of the file starting from the current line. The second tells it to change every occurrence of the old word in each line.

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

Remember that the change starts with the current line. If you enter this kind of command when the current line is someplace in the middle of a file, the change is made until the editor reaches the end of the file. It does not go back to the beginning to make changes there.

Just as you think you have the entire story of the crime, a late flash comes in. The animal stolen was not a pig. It was a cow. To change the word *pig* to *cow* wherever it appears, first make the top line current:

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

Then enter:

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

Your file looks like this:

```
 CRIME    THEFT    A1  F 80  Trunc=80 Size=4 Line=5 Col=1 Alt=6
2 occurrence(s) changed on 2 line(s)



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







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

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

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

Your FILELIST display returns to the screen.

Tips on Using LOCATE and CHANGE

The LOCATE and CHANGE commands both seem simple, but there are a few things you have to be careful of. Let us look at the VACATION CAMPING file. Move your cursor to that line in the FILELIST display, and press PF11. Your file looks like this:

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






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

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

Enter:

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

Here is the changed file.

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




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

Not quite what you expected, is it? XEDIT has changed the letters EAR to the letters NOSE wherever they appeared. As you can see, it is not a good idea to use the CHANGE command followed by * * unless you are absolutely sure of what you are doing. When you put * * after the CHANGE command, the editor takes you at your word that you want it to search every line of the file, from the current line to the last line, to find the letters and make the change.

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

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

Then enter:

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

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

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

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

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

Enter:

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

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

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

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

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

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

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

Then you would enter the command:

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

Returning to the vacation file, let us change NOSE back to EAR to practice another command.

First enter:

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

Enter:

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

Backward Locate

A backward LOCATE works just like the LOCATE command, except the editor searches backward (toward the top of file) rather than forward (toward the end of file).

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

Enter:

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

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

Locating Adjacent Words

The LOCATE command can be used to locate two or more adjacent words in a line. For example, you could enter LOCATE/ WILD BEARS/ in the VACATION CAMPING file. These two words are on the same line. (It is possible to use the LOCATE command to locate two words that are not on the same line, but that is beyond the scope of this document.)

Let us return this file to permanent storage.

Enter:

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

Your FILELIST display returns to the screen.

The = (Equal) Command

Call up the STOCK INVENTORY file by moving the cursor to that line in your FILELIST display and pressing PF11.

The file looks like this:

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




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

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

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

Enter the LOCATE command to find AA:

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

The first line containing AA is now the current line. You want to quickly scan the file, locating AA in every line it appears. Do you have to retype the LOCATE command? No, you can use the PF9 key, which is set to the *= (equal) command*. The = command automatically repeats the last command you entered. It saves you from having to retype it. It can be used to repeat any command, not just a LOCATE command.

Now press PF9. The next line that contains AA becomes the current line. Press PF9 again. The same thing happens. Press PF9 a third time. You see the message

```
 Target not found
```

This means that there are no more lines containing AA.

We are finished working with this file. Press ERASE EOF to clear the command line.

Then enter:

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

Now press the PF3 key to exit from FILELIST.

If you have had enough for this terminal session, log off and turn off your terminal (see "Logging On and Logging Off" on page 1). If not, continue with the next section.

Summary

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

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

Changing Files (Part Three)

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

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

Inserting Data from Another File

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

Inserting an Entire File

We do not have any files that logically belong together. So we will just combine files to practice the GET command.

Now enter:

`filel` (short for FILELIST)

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

- Moving the cursor to the line containing PICNIC INGREDTS,
- Then pressing PF11.

The GET command inserts another file *after the current line* in the file you are editing. Therefore, you must make your current line the line you want the data inserted after. Let us insert the camping file after the last line of the picnic file. We will make the last line current by using the BOTTOM command.

Enter:

====> b (short for BOTTOM)

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

```
GET filename filetype filemode
```

You substitute the file name and file type of the file you want to insert.

Filemode Note: It is necessary to specify the filemode if it is other than the default of *****.

Now enter:

```
====> GET vacation camping
```

The following message is displayed:

```
EOF reached
```

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

Let us put the file back into permanent storage.

Enter:

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

The revised file is now in permanent storage. The FILELIST display returns to the screen.

What has happened to the original VACATION CAMPING file? Is it gone from permanent storage? No, the editor has only made a copy of it and inserted the copy into the PICNIC INGREDTS file.

Inserting Part of a File

Let us look at the camping file. Move your cursor to the VACATION CAMPING line on the FILELIST screen and press PF11.

We will insert part of the radio waves file after the Top of File line of the camping file. We do not have to change which line is current, because Top of File is the current line. You show the *line number* of the first line and the *number of lines* you want to insert at the end of the GET command, like this:

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

Filemode Note: It is necessary to specify the filemode if it is other than the default of *****.

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

Enter:

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

The editor inserted lines 1 through 11 of the shortwave file after the Top of File line in the camping file. The last line inserted

```
(BEIJING        9780)
```

is the new current line.

I Do Not Want to Keep the Changes!

Because you have been using the camping file only to practice the GET command, you have really created a mess. You do not want to keep this version. How can you get rid of the changes and keep the camping file as it was originally? That is easy. Press PF3, which is set to the QUIT command. (You can also enter QUIT on the command line, but pressing PF3 is easier.)

Now press PF3. The following message is displayed:

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

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

Before proceeding, press Enter to clear the warning message and return the cursor to the typing position.

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

Now enter:

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

The FILELIST screen returns. What has happened to the camping file? Nothing. Because you entered the QQUIT command instead of the FILE command, the camping file in permanent storage is not changed. It remains as it was when you started this editing session.

When you get a copy of a file on your screen just to look at, not to change, it is better to QUIT instead of FILE when you are finished. The FILE command tells the editor to rewrite the changed file over the old file in permanent storage. (The QUIT command is faster because the editor does not have to rewrite the file.) And because you have not made any changes, you really do not need to have the editor rewrite the file.

You can also use QUIT instead of FILE when you are creating a new file and decide not to keep it. Let us create a new file like we did in "Creating Your First File" on page 8, only this time we will use QQUIT to exit the file without saving a copy. We will call it PHONE NUMBERS.

Now exit from the FILELIST screen using PF3. On the command line at the bottom of the left of the screen, enter:

```
x phone numbers
```

Once in the editor, set tabs by entering:

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

Enter the command that puts you in input mode:

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

Now type these lines, using PF4 for tabbing:

```
Gable, Clark                212-111-1212
Hari, Mata                  312-333-1313
Hun, Attila the             514-555-1414
Ripper, Jack D.             915-999-1515
```

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

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

When you enter FILELIST again you will see that the file PHONE NUMBERS does not exist.

SET NUMBER ON Command

When you used the GET command to insert part of a file, you had to know the starting line number and the number of lines you wanted to insert. So you are probably thinking that the GET command is great if you know the numbers of the lines you want to insert. But what if you are working with a long file and you want to get lines from somewhere in the middle? How are you going to count all those lines? You do not have to. You can use the SET NUMBER ON command to number the lines of the file. Its format is:

```
SET NUMber ON
```

Let us insert part of the radio waves file into the camping file. This time we will ask the editor to number the file lines.

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

```
====> set num on
```

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

The line numbers are not saved when you enter FILE or QUIT. If you want line numbers the next time you edit the file, you will have to enter the SET NUMBER ON command again.

When you no longer need the line numbers, you can enter the SET NUMBER OFF command. Its format is:

```
SET NUMber OFF
```

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

Reviewing the GET command format:

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

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

Now you are finished with this file. Press PF3 to quit.

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

Now enter the GET command:

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

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

We do not want to save these changes, so press PF3.

Then enter:

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

Press PF3 again to quit from FILELIST.

PUT Command

Another way you can insert part of a file into another file is to use the PUT command. The PUT command lets you insert lines from a current file into a second file.

The PUT command stores a copy of a certain number of lines, ***starting with the current line***. You can then append the stored lines to the end of another file.

The format of the PUT command is:

```
PUT number-of-lines filename filetype filemode
```

Filemode Note: It is necessary to specify the filemode if it is other than the default of **=** (equal to the file mode of the file you are currently editing).

The command ***puts*** the number of lines into the file you specify.

Another way to tell the editor how many lines you want to ***put*** is by typing a string. Remember the LOCATE command? When you learned about the LOCATE command, you searched files for a certain word or words—a string. You can also use a string with the PUT command. You can tell the editor to ***put*** all the lines, starting with the current line, up to but not including the line with the string.

To do so, use this command format:

```
PUT /string/ filename filetype filemode
```

Filemode Note: It is necessary to specify the filemode if it is other than the default of **=**.

Therefore, both PUT *number-of-lines* and PUT */string/* do the same thing.

Let us insert part of the STOCK INVEN file into CRIME THEFT.

Enter:

```
x stock inven
```

The file contains the following lines:

```
03469  GASKET          AA0306  9432
32554  RATCHET         AA0205  3762
32868  DOOHICKEY       BB0206  5248
64400  GIZMO           CC0101  9942
09263  WIDGET          AA0403  8645
60637  THINGAMABOB     BB0202  5432
26023  WHATCHAMACALLIT BB0306  9321
```

We are going to insert the first two lines of STOCK INVEN after the last line in CRIME THEFT. So make the first line of STOCK INVEN the current line.

Now enter:

```
====> put 2 crime theft
```

This command will append the lines

```
03469  GASKET          AA0306  9432
32554  RATCHET         AA0205  3762
```

to the bottom of the CRIME THEFT file.

Press PF3 to quit the STOCK INVEN file (leaving it unchanged).

Then enter

```
x crime theft
```

to view the new contents of the CRIME THEFT file.

CRIME THEFT now contains these lines:

```
Mort, Mort, the plumber's son,
Stole a cow and away he run.
The cow got loose and scared a goose,
And Mort got put in the calaboose.
03469  GASKET          AA0306  9432
32554  RATCHET         AA0205  3762
```

The lines you *put* in were inserted at the end.

At this point, you can delete the last two lines and enter FILE to return the file to its original form. Or, if you prefer, simply quit the file, leaving the additional lines you *put* in from STOCK INVEN.

Sorting a File

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

Enter:

```
filel
```

On the FILELIST screen, move the cursor to the RADIO WAVES line and press PF11.

Your file looks like this:

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




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

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

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

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

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

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

  SORT *number-of-lines column1 column2*

You substitute the number of lines you want sorted for *number-of-lines*. You can substitute an asterisk (*) here when you want all lines, from the current line to the last, to be sorted. *Column1* is the first column in the sort field. *Column2* is the last column.

The SORT command begins with the current line, so you need to make the top line current.

Enter:

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

Now enter:

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

As you can see, the place names are now in alphabetic order. The SORT command automatically defaults to **ascending order** unless otherwise specified.

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




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

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

Enter:

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

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

Enter:

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

Your file should look like this:

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



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

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

What if you wanted the numbers sorted in descending order, from highest to lowest? That is easy. Just type a letter *d* in the SORT command, before the first column number. The *d* stands for ***descending order***. (If you do not type a *d*, the default is assumed and the file is sorted in ascending order.)

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

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

Then enter:

```
====> sort * d 20 23
             │
             └──────> means "descending order"
```

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

Enter:

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

Your file looks like this:

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




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

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

Sorting Part of a File

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

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

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

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

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

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

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

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

Now press the PF3 key to exit from FILELIST.

Getting Help

If you are editing a file and forget how to use a command, or you would like to see information about XEDIT commands not covered in this document, press PF1. In XEDIT, PF1 is set to the HELP command.

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

```
XEDIT MENU                 Menu Help Information        line  1 of  25
(c) Copyright IBM Corporation 1990, 2004

Help for XEDIT subcommands

To view a Help panel, move the cursor to any character of the name
and press the Enter key or the PF1 key.
An asterisk (*) preceding the name indicates a MENU panel.
A colon (:) preceding the name indicates a TASK panel.
For a general description of the XEDIT command, type HELP CMS XEDIT.


&          CDelete   CP        Help      Next       REFRESH   SOS
*Prefix    CFirst    CReplace  HEXType   NFind      RENum     SPlit
*QUERY     Change    CURsor    Input     NFINDUp    REPEat    SPLTJOIN
*SET       CInsert   DELete    Join      Overlay    Replace   STAck
?          CLAst     Down      LEft      PARSE      RESet     STATus
:XEDIT     CLocate   DUPlicat  LOAD      POWerinp   RESTore   SUPerset
=          CMS       EMSG      Locate    PREServe   RGTLEFT   TOP
PF1= Help    2= Top      3= Quit    4= Return    5= Clocate   6= ?
PF7= Backward 8= Forward  9= PFkeys  10=          11=         12= Cursor

====>
                                               Macro-read 1 File
```

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

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

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

Summary

You learned about the following commands and PF keys in this section:

Command/PF Key	Function
GET	Inserts all or part of one file in another.
SET NUMber ON/OFF	Puts line numbers in the prefix area (ON) or removes them (OFF).
PUT	Inserts all or part of one file into another. You can insert a certain number of lines or insert all the lines up to a certain word or phrase.
PF3 (QUIT)	Ends an editing session if you did not make any changes.
QQuit	Enter this command if you want to quit after making changes.
SORT	Arranges a file in alphabetic or numeric order, according to sort fields.
PF1 (HELP)	Displays a menu of all XEDIT commands.

Protecting Your Files

How can you protect your files from computer or communication line failure? First, you should understand that your files in permanent storage are safe if the system malfunctions (or **goes down**, in computer talk). It is the new file you are creating or the changes that you are making to an existing file that can be lost if the system fails. You can minimize this risk by using either the SAVE command or the SET AUTOSAVE command, or both.

SAVE Command

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

```
SAVE
```

After you press Enter, the file you are working on is written to permanent storage (in an SFS file pool or on a minidisk), just like the FILE command. However, it stays on the screen. After you add lines or make more changes, you will have to enter the SAVE command again to save those new lines or changes.

If the system goes down, all the work you completed before you entered the last SAVE (or FILE) command would be saved. To display your file when the computer begins to function again, you would enter the XEDIT command (or enter FILELIST and select the file from the list). Sometimes you do not even need to do that. The file might appear on the screen when the system comes back on. If you made any changes between the time you entered your last SAVE command and the time the system went down, you will have to redo them.

If you entered a SAVE command while working on a file, you still have to enter a FILE (or QUIT) command to end the editing session. If you use a SAVE command while working on a file and then enter QQUIT, the changes you made before you entered SAVE are written to permanent storage. Any changes you make after you enter SAVE but before you enter QQUIT, however, are not saved.

SET AUTOSAVE Command

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

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

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

The SET AUTOSAVE command can be entered at any time during an editing session. However, it is best to enter it at the beginning of an editing session, right after the XEDIT command.

If you entered a SET AUTOSAVE command and the system goes down, your file would not be saved under its own file identifier. It will have a new file identifier. Every time the automatic save occurs, you will see the message

```
AUTOSAVED as nnnnnn AUTOSAVE A1
```

The part of the message that reads *nnnnnn* AUTOSAVE A1 is the new identifier. The file name will be a number (like 100001) and the file type will be AUTOSAVE. You would now have two files, one with the original identifier, and one with the AUTOSAVE identifier. However, the AUTOSAVE file is the more current one, so you want to replace the old file with it. Here is how.

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

```
 yourid   FILELIST    A0  V 108  Trunc=108 Size=6 Line=1 Col=1 Alt=0
Directory = VMSYSU:yourid.
Cmd   Filename Filetype Fm Format Lrecl Records Blocks   Date     Time
_     100001   AUTOSAVE A1 F        80     12      1   5/04/90 21:14:04
      RADIO    WAVES    A1 F        80      3      1   5/04/90 21:12:04
      PICNIC   INGREDTS A1 F        80      3      1   5/04/90 20:58:07
      VACATION CAMPING  A1 F        80      8      1   5/04/90 17:59:00
      CRIME    THEFT    A1 F        80      4      1   5/02/90 15:33:05
      STOCK    INVEN    A1 F        80      7      1   4/25/90 12:10:03




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

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

Note: Remember that your FILELIST screen will not look exactly like this if your files are stored on minidisks. The directory line will not be shown, and some of the PF keys will have different settings.

Your current RADIO WAVES file would be stored in the file 100001 AUTOSAVE A. You would use the REPLACE option of the COPYFILE command to copy the 100001 AUTOSAVE A file over the RADIO WAVES A file. You would then use the DISCARD command from FILELIST to discard 100001 AUTOSAVE A. The COPYFILE and DISCARD commands are explained in Chapter 3, "Managing Your Files," on page 53.

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

```
Query AUtosave
```

A message like the following is displayed:

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

AUTOSAVE 10 is the number you specified in the SET AUTOSAVE command. *Filename 100001* means the file name for this AUTOSAVE file is 100001. (The file type is always AUTOSAVE.) If an AUTOSAVE file already exists, the editor picks the next available number. For example, if the file 100001 AUTOSAVE A exists, the editor would name this file 100002 AUTOSAVE A. *Alterations : 3* is the number of the changes that have been made. When the number of alterations reaches ten, the file is saved. Every time the file is saved, this number is set back to zero. The number of alterations is also shown in the file identification line ($Alt=n$).

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

If you enter a SET AUTOSAVE command while you are creating a file, and then enter QQUIT, the file will not be saved. The same is true if you enter a SET AUTOSAVE command while you are revising a file and enter QQUIT. If, however, you have made enough alterations for an automatic save to occur, the AUTOSAVE file will be in permanent storage.

Summary

You learned the following commands in this section:

Command	Function
SAVE	Writes the file you are working on to permanent storage.

Command	Function
SET AUtosave	Automatically issues a SAVE periodically.
Query AUtosave	Displays autosave information.

More About Editing

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

Power Typing

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

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

```
POWerinp
```

Let us create a file to practice power typing. Follow the logon procedure if you are not already logged on.

Then enter:

```
x marigold story
```

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

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

Now enter:

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

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

If you want a sentence to begin on a new line, you must precede it with a pound/number sign (#). It will not appear in your file. It simply tells the editor to start that text on a new line. Now type the following information.

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

```
   make
   excellent cut flowers.
```

You could fill the entire screen if you wanted to. When you run out of space on the screen, simply press Enter, and more typing room will be provided for you. When you want to stop power typing, press Enter twice. Do that now.

Then enter:

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

Your screen looks like this:

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




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

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

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

If you want to insert characters or spaces in a line while you are using power typing, just press the insert mode key. You do not have to press the PA2 key, as you did when in regular input mode or edit mode. Remember to press RESET when you are finished using insert mode.

Splitting the Screen

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

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

A format of the SET SCREEN command is:

```
SET SCReen n
```

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

Multiple Views of the Same File

Let us practice splitting the screen.

Enter:

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

Two identical views of the same file appear:

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



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



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

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

When you decide you have had enough of seeing double, you can enter:

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

Multiple Views of Different Files

Now let us look at two different files at the same time.

Enter:

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

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

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

Two different files are displayed.

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




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




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

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

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

Then enter:

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

Now enter:

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

Creating an XEDIT Profile

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

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

A sample PROFILE XEDIT file might contain these lines:

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

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

Note: If your SET command contains a special character, it may cause an error, unless you enclose the command in ' ' (quotation marks). Therefore, we have entered each command with surrounding quotation marks to insure proper programming technique and prevent future problems.

Each command must be typed on a separate line. You should be familiar with all the commands. You already know that when you edit a file, the PF keys have various functions. For example, PF3 is set to the QUIT command, PF7 is set to the BACKWARD command, and so forth. If you would rather have one or more PF keys set to different commands, you can do this in the PROFILE XEDIT file. The command has the format:

```
SET PFn command
```

You substitute the number of the PF key for *n*, and you substitute the command you want assigned to that PF key for *command*. In our example, we have assigned the TOP command to PF12. After this PROFILE XEDIT file is written to permanent storage, pressing PF12 does the same thing as entering TOP at the command line.

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

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

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

Summary

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

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

For more information about XEDIT, see *z/VM: XEDIT User's Guide* and *z/VM: XEDIT Commands and Macros Reference*.

Chapter 3. Managing Your Files

In Chapter 2, "Editing Files," on page 7, you learned how to use the editor to make changes to the contents of your files. But sometimes you may want to make other kinds of changes. For example, you may want to rename a file or even discard it. Or what do you do when you forget a file identifier and you want to edit that file? The commands discussed in this chapter make it easy for you to manage your files, no matter how many you have.

This chapter teaches you how to:

- Display a list of your files
- Discard or rename a file
- Compare or copy files
- Organize your files
- Share your files.

What Files Do I Have?

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

```
FILEList
```

Remember that every file has a three-part identifier: file name, file type, and file mode. Unless you specify a different file mode, the files you create and change have a file mode of **A**. This means that they are all kept in permanent storage that is accessed with a file mode of **A**.

In Chapter 1, "Getting Started," on page 1, we explained that your files can be stored in a Shared File System (SFS) file pool or on a minidisk. We also discussed how to find out which type of storage you have.

If your files are stored in an SFS file pool, new files you create will, by default, be created in a directory accessed with a file mode of **A**—your ***top directory***. A directory is a structure for organizing your files. When your system administrator enrolled you in a file pool, he or she also set up a top directory for you. (We will discuss directories in more detail later in this chapter.)

If your files are stored on a minidisk, new files you create will, by default, be created on a minidisk accessed with a file mode of **A**.

The FILELIST command displays a list of all the files on ***file mode A*** (your top directory or minidisk accessed with a file mode of A). Later, you will see how to display a list of files with other file modes, such as file mode B, file mode C, and so forth.

If you did the exercises in Chapter 2, "Editing Files," on page 7, your list of files looks similar to that shown on the following screen.

```
 yourid    FILELIST A0  V 108  Trunc=108 Size=6 Line=1 Col=1 Alt=0  1
Directory = VMSYSU:yourid.  2
Cmd  Filename Filetype Fm Format Lrecl   Records    Blocks   Date     Time  3
_    STOCK    INVEN    A1 F        80        7         1   4/26/90 14:48:20
     MARIGOLD STORY    A1 F        80        8         1   4/14/90 14:45:02
     PICNIC   INGREDTS A1 F        80        3         1   5/06/90 13:27:28
     VACATION CAMPING  A1 F        80        8         1   5/04/90 10:50:59
     CRIME    THEFT    A1 F        80        4         1   5/03/90 10:45:19
     RADIO    WAVES    A1 F        80        3         1   5/04/90 10:35:00




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

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

The FILELIST screen shows information about the files on file mode A.

1

The first line has information about the FILELIST display itself. Your user ID appears instead of *yourid*. The Size field shows how many files are in this list. Here, Size=6 means there are 6 files on this file mode A. The Line field tells you which file in the list is the first on this screen. Here, Line=1. As you scroll through the list, the number will be greater than one. The other fields are not important to you now.

2

The next line shows the name of the directory where these files reside. This line will appear only if your files are stored in an SFS file pool; it will not be displayed if your files are stored on a minidisk.

3

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

Cmd
is a space where you can type certain commands. We will discuss this later.

Filename
is the file name of the file.

Filetype
is the file type of the file.

Fm
is the file mode of the file. (You can ignore the number that appears after the file mode letter, for example, A1.)

Format Lrecl Records Blocks
These fields give information about the file format but are not important to you now.

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

4

The PF keys are set to various useful functions, which are shown at the bottom of the screen. The PF keys in this example are those you will see if your files are stored in a Shared File System file pool. Some of the PF keys will perform different functions if your files are stored on a minidisk. We will discuss the FILELIST PF keys in more detail throughout this chapter. You will also notice that the PF keys shown on the FILELIST screen are different from those shown when you are editing a file.

For now, note that when you are finished using the list, press PF3 to quit the FILELIST screen.

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

Finding Files in the List

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

Scrolling through the List

If the list takes up more than one screen, you can use PF keys to **scroll** through it. Scrolling is like turning the pages of a book. PF7 scrolls backward one full screen. PF8 scrolls forward one full screen. By pressing these keys repeatedly, you can scroll through the list for as many screens as you want.

Using the LOCATE Command

If you know a file name or file type, you can use the XEDIT command LOCATE to find it quickly. This method is faster than scrolling through the screens, looking for a certain file. You enter a LOCATE command on the command line at the bottom of the screen, next to the large arrow. Then press Enter.

For example:

```
====> Locate /crime theft/
```

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

More Ways to Use FILELIST

You have seen how to use the FILELIST command to display a list of **all** the files on file mode A, but FILELIST has many additional uses. You can use it to display all or **some** of your files using the FILELIST command options. You can also delete, change, rename, copy or compare your files from the FILELIST screen.

Using FILELIST to Display Some of Your Files

You can also use FILELIST to display only **some** of your files. For example, all files that have the same file name, or the same file type, or all files that begin with a certain letter, and so forth.

The FILELIST command has the format:

```
FILELIst filename filetype filemode
```

If you do not substitute a letter for *filemode,* a file mode of A is assumed. In our examples, all the files are on file mode A (the directory or minidisk accessed with a file mode of A).

Suppose you have several files that concern stock inventory. You know that they all have the same file name, STOCK, but you do not remember the different file types. To see a list of all the files on file mode A with the file name STOCK, you would enter:

```
filel stock *
```

The asterisk means **all**. Typing an asterisk where you would usually type a file type means that you want the list to contain all file types with the file name STOCK.

The FILELIST display might look like this:

```
 SOMEONE     FILELIST   A0  V 108  Trunc=108 Size=6 Line=1 Col=1 Alt=0
Directory = VMSYSU:yourid.
Cmd  Filename Filetype Fm Format Lrecl Records Blocks   Date     Time
_      STOCK    BOLTS    A1 V        95       7      1  5/04/90 21:15:04
       STOCK    GADGETS  A1 V        95       8      1  5/04/90 21:12:04
       STOCK    NUTS     A1 V        95       4      1  5/04/90 20:58:07
       STOCK    SCREWS   A1 V       107       5      1  5/04/90 17:59:00
       STOCK    SURPLUS  A1 V        92       3      1  5/02/90 15:33:05
       STOCK    WIDGETS  A1 V        75      12      1  4/25/90 12:10:03




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

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

Note: Remember, your FILELIST screens will look slightly different if your files are stored on a minidisk.

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

```
filel * data
```

The list would contain all files whose file type is DATA.

If you want to create a file and do not remember if you already have a file with the file identifier you want to use, you can enter:

```
FILEL filename filetype
```

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

Sometimes you may not remember an entire file name or a whole file type. When you have many files, this can happen. You can give the computer clues to display a list of files containing the file you want.

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

```
filel com* data
```

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

```
COMBAT     DATA  A
COMMERCE   DATA  A
COMPARE    DATA  A
COMPOST    DATA  A
COMPUTER   DATA  A
```

You can use an asterisk to represent any letters you cannot remember. The asterisk can be used anywhere in the file name or file type. For example:

The command

```
filel *data daily1
```

might show these files:

```
HISDATA    DAILY1  A
HERDATA    DAILY1  A
```

```
MYDATA     DAILY1  A
YOURDATA   DAILY1  A
```

And this command

```
filel *data daily*
```

might list these files:

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

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

```
filel * data b
```

lists all the files on file mode B whose file type is DATA. Similarly,

```
filel * data *
```

lists all files whose file type is DATA.

Editing a File Using FILELIST

You know that to get a copy of a file on your screen, you use an XEDIT command. You can type an XEDIT command on the FILELIST screen, next to the large arrow at the bottom of the screen. Then press Enter. An easier way is to:

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

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

When you are finished editing the file and enter FILE or QUIT, the FILELIST display returns to the screen. The cursor remains on the line that describes the file you just edited, and an asterisk (*) appears in the Cmd space (next to the file name). This is just to remind you that you did something with this file. It does not prevent you from working with this file again.

To create a file, you type an XEDIT command (XEDIT filename filetype) on the command line next to the large arrow at the bottom of the FILELIST display. Then press Enter. When you are finished creating the file and enter a FILE command, the FILELIST display returns.

Your new file will not appear in the list until you *press PF2*. PF2 is set to the *refresh* function. This means that the list is refreshed with new information.

Discarding a File

Suppose you want to get rid of some files you have created. If they were on paper, you would tear them up and throw them away. But how do you destroy files that are cluttering up your top directory or minidisk? You can use the DISCARD command.

To discard a file, you would:

- Type DISCARD in the Cmd space next to the file (or files) you no longer want. If the command name is longer than the space provided, just type right over the information already in that line.
- Clear the remainder of the information on the line by pressing the ERASE EOF key or by typing consecutive blanks.

- Press Enter to process the DISCARD command.

For example, suppose you wanted to discard the STOCK NUTS and STOCK SCREWS files. You would follow the first two steps for discarding a file.

Here is the result:

```
 SOMEONE  FILELIST    A0  V 108  Trunc=108 Size=6 Line=1 Col=1 Alt=0
Directory = VMSYSU:yourid.
Cmd   Filename Filetype Fm Format Lrecl Records Blocks   Date     Time
      STOCK    BOLTS    A1 V           95       7       1  5/04/90 21:15:04
      STOCK    GADGETS  A1 V           95       8       1  5/04/90 21:12:04
discard
discard
      STOCK    SURPLUS  A1 V           92       3       1  5/02/90 15:33:05
      STOCK    WIDGETS  A1 V           75      12       1  4/25/90 12:10:03




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

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

In this example, two files would be discarded. After you press Enter, a notice appears in the line, telling you the file has been discarded.

The lines would then look like this:

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

You must press Enter to process the DISCARD command typed on one screen before scrolling to another screen.

If you have many files to discard, you do not have to type out the word *discard* next to each file. You can take a shortcut.

```
 SOMEONE  FILELIST    A0  V 108  Trunc=108 Size=6 Line=1 Col=1 Alt=0
Directory = VMSYSU:yourid.
Cmd   Filename Filetype Fm Format Lrecl Records Blocks   Date     Time
      STOCK    BOLTS    A1 V           95       7       1  5/04/90 21:15:04
discard
=     STOCK    NUTS     A1 V           95       4       1  5/04/90 20:58:07
=     STOCK    SCREWS   A1 V          107       5       1  5/04/90 17:59:00
=     STOCK    SURPLUS  A1 V           92       3       1  5/02/90 15:33:05
=     STOCK    WIDGETS  A1 V           75      12       1  4/25/90 12:10:03




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

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

Suppose the only file you want to keep is STOCK BOLTS. To discard the rest, you need type DISCARD only once on a screen. Just type an equal sign (=) in the Cmd space next to the other files you want to discard. The equal sign operates only on the lines following the command. (See the previous example.) It means

that you want the same command that you typed to be carried out for this file, too. Then press Enter. The screen would look like this:

```
 SOMEONE  FILELIST    A0  V 108  Trunc=108 Size=6 Line=1 Col=1 Alt=20
Directory = VMSYSU:yourid.
Cmd   Filename Filetype Fm Format Lrecl Records Blocks   Date     Time
      STOCK    BOLTS    A1 V          95       7     1  5/04/90 21:15:04
*     STOCK    GADGETS  A1 has been discarded
*     STOCK    NUTS     A1 has been discarded
*     STOCK    SCREWS   A1 has been discarded
*     STOCK    SURPLUS  A1 has been discarded
*     STOCK    WIDGETS  A1 has been discarded




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

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

Be very cautious when you are discarding files. After a DISCARD command is processed, the computer does not ask if you would like to think it over. The file is gone.

Renaming a File Using FILELIST

What if you decide that a file identifier is not quite right? Perhaps it is too similar to the identifier for another file. Or maybe you want to create a new file with that identifier. You can change the file name or file type (but not the file mode) with the RENAME command.

You type the RENAME command on the FILELIST screen, in the Cmd space next to the file you want to rename (just like the DISCARD command). You process it by pressing Enter. Use this format of the RENAME command:

```
 Rename  /  newfilename newfiletype filemode
```

First you type RENAME in the Cmd space. You continue typing the rest of the command right over the information that is already on the line. The slash (/) stands for the file identifier of the file you want to rename. You then type the new file identifier (file name, file type, and file mode). Press Enter to carry out the RENAME command.

Suppose you want to rename VACATION CAMPING A to GREAT OUTDOORS A. (If you did the exercises in Chapter 2, "Editing Files," on page 7, you can rename your file by following these instructions.)

This line in the FILELIST display looks like this:

```
 VACATION CAMPING A1 F          80       8     1  5/04/90 10:50:59
```

You type the RENAME command right over this line, as shown in the following example. Remember to type all ***three*** parts of the new file identifier: file name, file type, and file mode.

```
 rename / great outdoors a1 F          80       8     1  5/04/90 10:50:59
                              └──────> filemode
              │       └───> newfiletype
              └───> newfilename
      └─────> stands for VACATION CAMPING A1
```

When you press Enter, your command is processed. It is replaced with this notice:

```
 *       VACATION CAMPING A1 ** Discarded, Renamed, or Relocated **
```

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

Copying a File Using FILELIST

Sometimes you may want to make changes in a file, but you want to keep both the original file and the revised one. By using the COPYFILE command, you can make a copy of a file, make changes in it, and keep both the original file and the copy.

You type the COPYFILE command on the FILELIST screen, starting in the Cmd space next to the file you want to copy (just like the DISCARD command). Press Enter to process the command. The format of the COPYFILE command is:

```
COPYfile  /  newfilename newfiletype newfilemode
```

First you type COPYFILE in the Cmd space. You continue typing the rest of the command right over the information that is already on the line. The slash (/) stands for the file identifier of the file in this line. You then type the new file identifier (file name, file type, and file mode). Press Enter to carry out the COPYFILE command.

For example, you have a file RADIO WAVES A. You want to make changes in the file, but you want to keep the original. (If you did the exercises in Chapter 2, "Editing Files," on page 7, you can copy your file by following these instructions.) The line in the FILELIST display looks like this:

```
RADIO    WAVES    A1  F        80      12      1  5/04/90 10:35:00
```

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

```
copy / radio newaves a  F      80      12      1  5/04/90 10:35:00
                    │             └──> newfilemode
                    │        └──>newfiletype
                    └──> newfilename
              └──> stands for RADIO WAVES A1
```

When you press Enter, the COPYFILE command is processed. To see the entry for the new file in your FILELIST display, press PF2. The new file appears at the top of the FILELIST display. Now you have two copies of the same file. Each has its own file identifier. You can make changes in RADIO NEWAVES, and still have the original file, RADIO WAVES.

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

Comparing Two Files Using FILELIST

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

You type the COMPARE command on the FILELIST screen, in the Cmd space next to the file you want to compare. Press Enter to process the command. The format of the COMPARE command is:

```
COMPARE  /  filename filetype filemode
```

First you type COMPARE in the Cmd space. You continue typing the rest of the command right over the information that is already on the line. The slash (/) stands for the file identifier of the file in this line. You then type the file identifier (file name, file type, and file mode) of the file you want to compare it with. Press Enter to carry out the COMPARE command.

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

```
RADIO     NEWAVES A1     F        80     12     1  5/06/90 12:10:03
```

Here is how your COMPARE command would look:

```
compare / radio waves a      F       80     12     1  5/06/90 12:10:03
                     |        |       |
                     |        |       └──> file mode of file to be compared
                     |        └─────> file type of file to be compared
                     └─────> file name of file to be compared
              └──> stands for RADIO NEWAVES A
```

After you press Enter, this message appears on a cleared screen:

```
Comparing RADIO NEWWAVES A with RADIO WAVES A
```

If the files are identical, nothing else is displayed. The status area (lower right corner) displays a MORE... notice. To get the FILELIST display back on the screen, press CLEAR. The FILELIST display comes back on the screen with an asterisk in the Cmd space.

If the files are different, the lines that do not match are also displayed. The first line is from the first file (RADIO NEWAVES A), and the second line is from the second file. After all lines that are different are displayed, the message:

```
Files do not compare
```

appears. If one file is shorter than the other, you will see this message:

```
Premature EOF on file filename filetype filemode
```

The screen will display MORE... in the status area. Press CLEAR to return to the FILELIST display. A number will appear next to the asterisk in the Cmd space (*40). This number is a return code; you can ignore it.

We are done working with the RADIO NEWAVES file. If you wish, you can discard it. Type:

```
discard
```

in the Cmd space next to the file. Then press Enter to discard the file. Press PF3 to quit from FILELIST.

Renaming, Copying, and Comparing Files From the Command Line

You have learned three new commands to enter from the FILELIST screen, but they can also be entered directly from the command line. Although the function of each command is the same as when entered from FILELIST, their formats are a little different. More specifically, the slash character (/) cannot be used with these commands outside of the FILELIST screen.

The command line format for each commands is:

```
Rename filename filetype filemode filename filetype filemode
       |        |        |        | |        |        |
       +--------+--------+--------+ +--------+--------+
                |                           |
          old file identifier        new file identifier


COPYfile filename filetype filemode filename filetype filemode
         |        |        |        | |        |        |
         +--------+--------+--------+ +--------+--------+
                  |                           |
            file identifier of the     new file identifier
             file you wish to copy


COMPARE filename filetype filemode filename filetype filemode
        |        |        |        | |        |        |
        +--------+--------+--------+ +--------+--------+
                 |                          |
           first file identifier     second file identifier
```

Summary

You learned the following commands in this chapter:

Command	Function
FILEList	Displays a list of your files.
DISCARD	Discards a file.
Rename	Renames a file.
COPYfile	Copies a file.
COMpare	Compares two files.

This is a good stopping point if you want to end your terminal session. If not, you can continue with the next section.

Using the Shared File System

The remainder of this chapter provides details on how to use the Shared File System (SFS) to organize and share your files. Depending on how your system administrator set up your system, you may have SFS storage, minidisk storage, or the capability to use both types of storage.

If, in Chapter 1, "Getting Started," on page 1, you determined that you have space in an SFS file pool, you can go to "Organizing Your Files" on page 63.

If you do not have space in a file pool, you will not be able to complete the exercises in the remainder of this chapter. However, you can use several SFS commands if you are enrolled in an SFS file pool without space. See *z/VM: CMS User's Guide* for details. Even if you are not enrolled in an SFS file pool, you can still share files with other users by linking to their minidisks. *z/VM: CMS User's Guide* has further details on how to do this.

If your files are stored on a minidisk, you may be able to complete these exercises if your system administrator has given you space in a file pool. Ask your administrator if you have space in a file pool. Also determine the name of your file pool (the ***file pool ID***).

Because you have used a minidisk to store your files, you will now need to complete the following steps to use SFS storage:

• Set your default file pool. Enter:

```
set filepool vmsysu
```

VMSYSU is the name of an IBM-supplied file pool that was shipped with your system. If your file pool is not VMSYSU, substitute the name of your file pool. This command will set up the file pool you specify as your default file pool.

- Access your top directory with a file mode of A. (We will discuss what a top directory is and what it can contain in the following section.) Enter the following command. Be sure to leave a space before and after the period.

```
acc . a
```

Note
If, after you enter this command, you receive the message   `File pool VMSYSU is unavailable or unknown` enter the command again to make sure you typed your file pool ID correctly. If you receive the same message, your file pool may be unavailable because of a system problem. Wait a few minutes and try entering the command again.

- Access your 191 minidisk with a file mode of M (or with another available file mode). Enter:

```
acc 191 m
```

This command will access your 191 minidisk (previously accessed as A) with another file mode. File mode A will now be used to access your top directory.

- Copy the files you previously created on your minidisk (now accessed as M) to your top directory (now accessed as A). Enter:

```
copy * * m = = a
```

This command will copy all files (regardless of their file name or file type) from the minidisk to the top directory. The use of the equal signs (=) here cause the identical file names and file types to be copied.

Now you are ready to complete the exercises that follow.

Note

If you log off and log on again while completing the exercises in this chapter, you will need to re-enter the first three commands
 	
`set filepool vmsysu:`
`acc . a`
`acc 191 m`
 	
to set up SFS storage for users who usually store their files on a minidisk and to re-establish the original CMS session.

Organizing Your Files

In SFS, you are given a certain amount of space, a ***file space***, to store your files. The file spaces for many users are kept in an area of real disk storage called a ***file pool***.

Within your file space, you can organize groups of related files into ***directories***. When your system administrator assigned you to a file pool, one directory was set up for you: your top directory. This top directory has as its name your user ID followed by a period. For example, if the user Mona Lisa has LISAM as her user ID, she would have LISAM. as her top directory.

When you complete the examples in this chapter, your own user ID will appear where we have shown *yourid*.

Your top directory can contain files. In fact, if you completed the exercises earlier in this document, your top directory now contains several files. You can also create lower-level directories below your top directory. Each of these directories can also contain files and other directories (up to eight levels of subdirectories). This file organization structure is called a *hierarchy* or *tree*. Actually, it resembles an upside down tree. Your top directory is at the top; lower-level directories branch out below it.

For example, suppose Dan is writing a mystery novel. His top directory (which has the same name as his user ID followed by a period) contains miscellaneous files. It also has subdirectories with information for the novel. The subdirectories contain files with the detailed information: the scene of the crime, the theft, the suspects, and, of course, the solution.

Such a tree would look like this:

Figure 1. Sample Tree

The first five items in this top directory

1 PROFILE EXEC, PROFILE XEDIT, LETR TOMOM, OLD STUFF, and NEW STUFF

are CMS files. The top directory also has four subdirectories, one for each part of the novel:

2 SCENE, THEFT, SUSPECTS, and SOLUTION.

The name

```
RICH WIDOW VMSYSU:DAN.SCENE
```

is the entire name of the RICH WIDOW file in the SCENE directory.

```
RICH WIDOW VMSYSU:DAN.SCENE
                          └──────> name of the subdirectory
                     └──────> Dan's top directory
                └──────> file pool identifier
           └──────> file type
      └──────> file name
```

RICH is the file name; WIDOW is the file type. VMSYSU: is the name of the file pool where the files are stored. DAN followed by a period is the name of Dan's top directory; SCENE is the name of the subdirectory (below his top directory) where the file is located.

You do not need to create lower-level directories. If you wish, you can easily keep all of your files in your top directory. You may find, however, that using several lower-level directories lets you better organize your files and group them by project or topic.

Working With Directories

There are several commands that allow you to work with your directories. These commands let you list, create, rename, and erase directories. Also, you can access and work with other users' directories if they grant you the authority to do so.

File Control and Directory Control Directories

There are two kinds of SFS directories that you can create: file control directories and directory control directories. File control directories let you control the sharing of individual files. Directory control directories provide less control and have characteristics similar to those of minidisks. Directory control directories are intended for special uses.

By default, all SFS directories you create are file control directories. For simplicity, this document describes only file control directories. If you want to learn more about directory control directories, see *z/VM: CMS User's Guide*.

Listing the Contents of Directories

To see what a directory contains, use the DIRLIST command. DIRLIST is a good starting place because it lets you to get the big picture of a directory.

The format of the DIRLIST command is:

```
DIRList dirid
```

For *dirid*, you can enter the name of the directory you wish to list. If the directory is accessed, you can specify the file mode letter of the directory. If you specify the command without a *dirid*, your top directory and its subdirectories will be listed.

If you completed the previous exercises in this document, you created several files in your top directory, which is accessed with a file mode of A. To list your top directory, enter:

```
dirl
```

Your screen will look like this:

```
 yourid    DIRLIST  A0  V 319  Trunc=319 Size=1 Line=1 Col=1 Alt=0
Cmd   Fm Directory Name
_     A  VMSYSU:yourid.




















1= Help       2= Refresh  3= Quit   4= Sort(fm)    5= Sort(dir)   6= Auth
7= Backward   8= Forward  9=        10=           11= Filelist   12= Cursor

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

DIRLIST lists the directory you specified and any subdirectories below it. You have not yet created any subdirectories, so only your top directory is shown.

As you can see, your top directory, VMSYSU:*yourid*., is accessed with a file mode of A.

At the bottom of the DIRLIST screen is a listing of PF keys available to you now. You are already familiar with some of these keys; we will explain how to use the others in the following exercises. For now, press PF3 to quit the DIRLIST screen.

Creating a New Directory

All the files you created earlier in this document are stored in your top directory. If you want to create new directories, you can do so by using the CREATE DIRECTORY command.

The CREATE DIRECTORY command has the format:

```
CREate DIRectory dirid
```

For *dirid*, specify the directory name. By default, the directory you create will be a file control directory.

For example, to create the directories shown in Figure 1 on page 64, Dan used the following commands:

```
create dir .scene
create dir .theft
create dir .suspects
create dir .solution
```

You can leave out your file pool ID (VMSYSU:) when entering commands. The system will assume that you want the command to be processed in your default file pool.

What does the period mean? When you are referring to your own top directory, there is no need to write out your user ID. You can simply specify a period; the system understands that you mean your top directory.

The first command in the previous list tells the system to begin with the top directory (specified by the period) and create the SCENE directory one level below it. The following commands create the THEFT, SUSPECTS, and SOLUTION directories, which are also one level below the top directory.

Suppose you want a directory to contain files on leisure activities. Enter the following command:

```
create dir .leisure
```

The command will create the LEISURE directory one level below your top directory.

If you could see your tree, it would look like this:
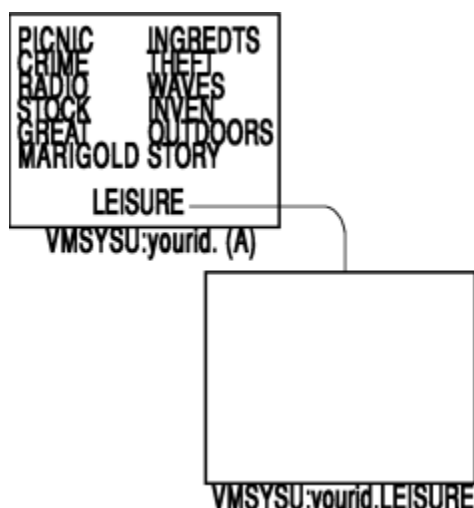
Figure 2. Creating the LEISURE Directory

The files shown in your top directory are those you created in Chapter 2, "Editing Files," on page 7.

You can create another directory to include information on summer leisure activities. To set up the SUMMER directory one level below the LEISURE directory, enter:

```
create dir .leisure.summer
```

Now your tree looks like this:
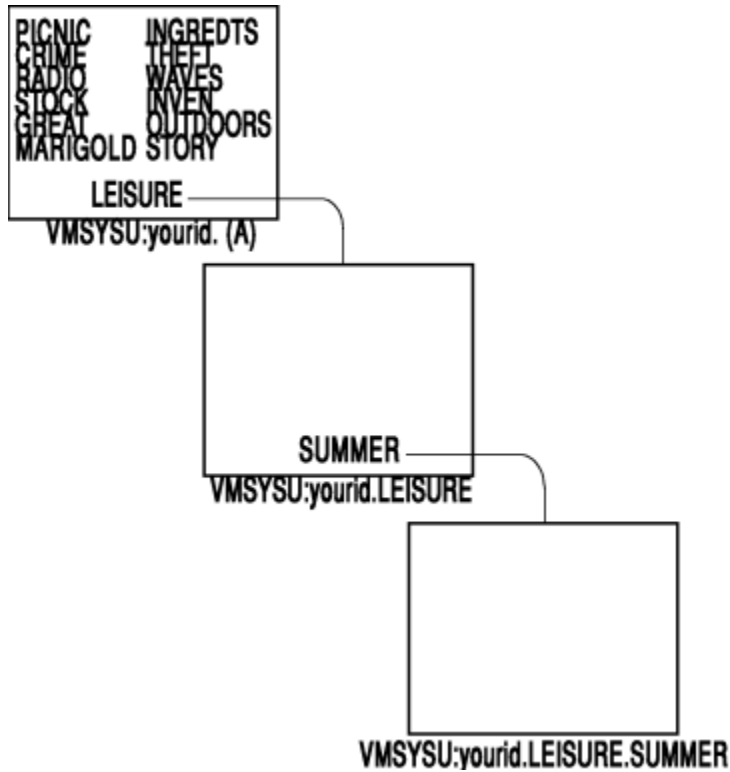
Figure 3. Creating the SUMMER Directory

LEISURE is one level below your top directory; SUMMER is one level below LEISURE.

Create one more directory. This one can contain miscellaneous lists and news. Enter:

```
create dir .misc
```

This command creates the MISC directory one level below your top directory (specified by the period).

Now your tree looks like this:
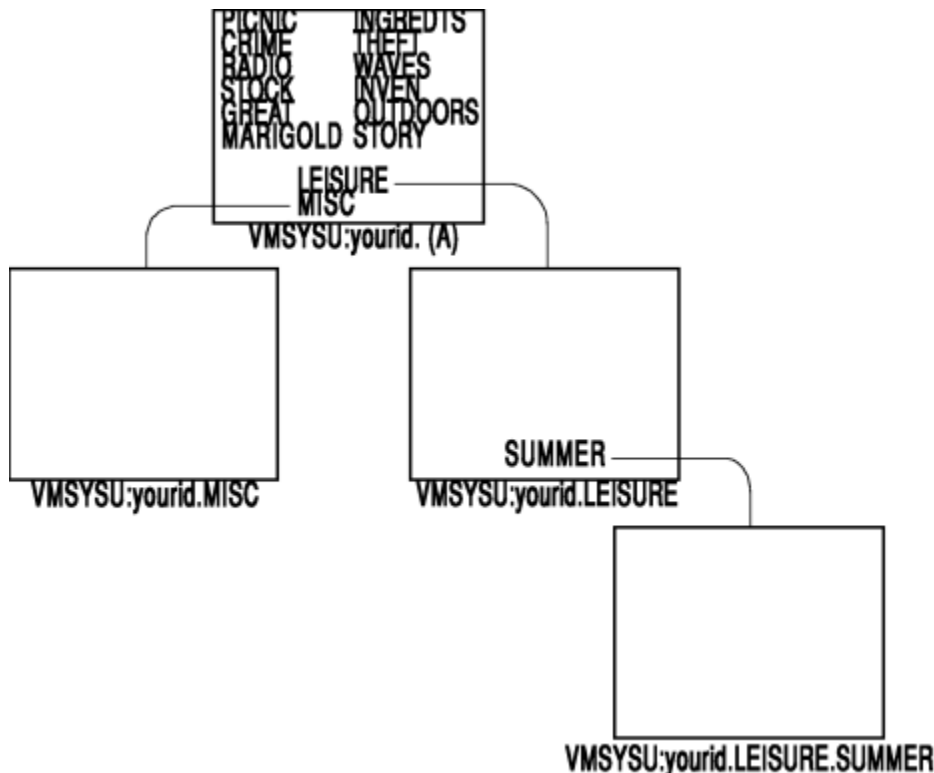
Figure 4. Creating the MISC Directory

To see a list of all the new directories you have created, use the DIRLIST command. Enter:

```
dirl
```

Your screen will look like this:

```
 yourid    DIRLIST  A0  V 319   Trunc=319 Size=4 Line=1 Col=1 Alt=0
Cmd    Fm Directory Name
_         A  VMSYSU:yourid.
          -  VMSYSU:yourid.LEISURE
          -  VMSYSU:yourid.LEISURE.SUMMER
          -  VMSYSU:yourid.MISC




















1= Help      2= Refresh  3= Quit   4= Sort(fm)    5= Sort(dir)   6= Auth
7= Backward  8= Forward  9=        10=           11= Filelist   12= Cursor

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

Because you did not specify a *dirid*, the DIRLIST command defaults to your top directory. You see a listing of the top directory and all the directories below it.

Your top directory is accessed with a file mode of A. The other directories show a dash (-) in the Fm column because you have not accessed them yet.

For now, do not quit the DIRLIST screen.

Erasing and Discarding Your Directories

The directories you just created and any other directories you create will remain in your tree until you erase or discard them.

The ERASE command lets you erase empty directories—that is, ones that do not contain any files. If you want to erase directories that contain files, erase each file and then erase the directory. You must be the owner of a directory to erase it.

To erase an entire file, simply specify the file name, file type, and *dirid* (directory name or file mode):

```
ERASE filename filetype dirid
```

To erase a directory, specify only the *dirid*:

```
ERASE dirid
```

If you want to erase **only the data in the file**, leaving just an empty file with that file name, you can use the DATAONLY option of the ERASE command. Then, if you later decide to eliminate the empty file, repeat the ERASE command without using the DATAONLY option.

If you are sure you want to erase a directory **and all the files it contains**, you can use the FILES option of the ERASE command. Unless you specify the FILES option, however, you cannot erase a directory that contains files. Also, you cannot erase a directory that contains subdirectories; you must first erase the subdirectories.

Another way to remove directories is by using the DISCARD command. DISCARD can only be entered from the Cmd area of DIRLIST or FILELIST. The DISCARD command will not remove any directories that contain files or subdirectories.

Renaming Your Directories

If you wish to change the name of a file or directory, you can do so by using the RENAME command.

Earlier in this chapter we practiced renaming files from the FILELIST screen. You can use the same method to rename your directories from DIRLIST.

The format of the RENAME command is:

```
Rename  /  newdirid
```

First you type RENAME in the Cmd space. Then you continue typing the rest of the command right over the information that is already on the line, just as you did when you renamed a file. Press Enter to rename the directory.

Suppose you want to rename the MISC directory and call it the LIST directory. The DIRLIST display is currently showing on your screen. Therefore, you can move your cursor to the Cmd space next to the MISC directory and enter the following command. Be sure to leave a space before and after the slash.

```
rename / .list
        |   |
        |   |_____> new directory name
        |_____>  stands for yourid.MISC
```

Your screen will look like this:

```
 yourid   DIRLIST  A0  V 319  Trunc=319 Size=4 Line=1 Col=1 Alt=0
Cmd   Fm Directory Name
      A  VMSYSU:yourid.
      -  VMSYSU:yourid.LEISURE
      -  VMSYSU:yourid.LEISURE.SUMMER
rename / .listU:yourid.MISC




















1= Help       2= Refresh  3= Quit   4= Sort(fm)    5= Sort(dir)   6= Auth
7= Backward   8= Forward  9=        10=           11= Filelist   12= Cursor

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

When you press Enter, your command is processed. You will then see this notice:

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

When you press PF2 to refresh your screen, the new directory name, LIST, will replace the old name, MISC.

Press PF3 to quit the DIRLIST screen.

You can use the RENAME command to change the name of your own directories or files. You can also change the name of another user's file if you have write authority to the file and to the directory where it resides. (We will discuss write authority later in this chapter.)

You cannot use RENAME to move a directory to another location. For example, you could not rename the LIST directory to the LEISURE.LIST directory. If you want to move the LIST directory to below the LEISURE directory, you must use the RELOCATE command. (We will discuss RELOCATE later in this chapter.)

Note: When a directory is renamed, any ACCESS commands that you entered remain in effect. If you enter a QUERY ACCESSED command after renaming a directory that you accessed previously, the command display will show the new directory name.

Working with Files

Now that you have several directories to work with, you can use CMS commands to list your files, create new files, and copy or relocate existing files.

Listing Your Files

If you completed the exercises in Chapter 2, "Editing Files," on page 7, you currently have the following files in your top directory:

- PICNIC INGREDTS
- CRIME THEFT
- RADIO WAVES
- STOCK INVEN
- GREAT OUTDOORS
- MARIGOLD STORY

Because you did not specify a file mode when you created these files, the files were, by default, created on file mode A (the directory or minidisk accessed with a file mode of A). Your top directory is now

accessed with a file mode of A. Therefore, you can use the FILELIST command to see a list of the contents of your top directory.

Enter the DIRLIST command again:

```
dirl
```

Your cursor will be located on the line for the VMSYSU:*yourid*. directory.

```
 yourid   DIRLIST  A0  V 319   Trunc=319 Size=4 Line=1 Col=1 Alt=0
Cmd    Fm Directory Name
_      A  VMSYSU:yourid.
       -  VMSYSU:yourid.LEISURE
       -  VMSYSU:yourid.LEISURE.SUMMER
       -  VMSYSU:yourid.LIST




1= Help      2= Refresh  3= Quit   4= Sort(fm)    5= Sort(dir)   6= Auth
7= Backward  8= Forward  9=        10=           11= Filelist   12= Cursor

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

You can use the PF keys shown on the DIRLIST screen to move further down in your tree and see a FILELIST display of any of these files. This will work just as if you entered a FILELIST command on the file.

Press PF11. You will see a FILELIST display of the files and directories currently in VMSYSU:*yourid*. (your top directory). Your screen will look something like this:

```
 yourid   FILELIST A0  V 108   Trunc=108 Size=8  Line=1 Col=1 Alt=0
Directory = VMSYSU:yourid.
Cmd    Filename Filetype Fm Format Lrecl    Records    Blocks   Date      Time
_      LIST              A  DIR       -        -          -    5/07/90 10:11:44
       LEISURE           A  DIR       -        -          -    5/07/90 10:10:04
       MARIGOLD STORY    A1 F        80        8          1    4/14/90 10:02:14
       GREAT    OUTDOORS A1 F        80        8          1    5/04/90  9:59:18
       STOCK    INVEN    A1 F        80        3          1    4/25/90  9:57:31
       RADIO    WAVES    A1 F        80        3          1    5/04/90  9:41:44
       CRIME    THEFT    A1 F        80        4          1    5/03/90  9:40:05
       PICNIC   INGREDTS A1 F        80        3          1    5/04/90  9:37:19




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

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

There are several options of the FILELIST command that display different information about your files. The one you are currently viewing is the default FILELIST display. It is called the FILELIST STATS screen.

As you can see, FILELIST displays all the files within the directory you specified—here, your top directory—and all the names of the subdirectories one level below it (the LEISURE and LIST directories). The LEISURE.SUMMER directory is not shown because it is two levels below your top directory.

You can always tell where you are in your tree by looking at the line labeled `Directory`.

Now it shows:

```
Directory = VMSYSU:yourid.
```

This means the files and directories you are viewing are within your top directory.

The FILELIST screen also contains some PF keys we have not yet discussed. You can use these PF keys to work with the files and directories shown on your screen:

- Use PF5 to sort your directories to the top of the list (directories are sorted alphabetically; files are sorted by date).
- Use PF6 to sort files to the top of the list and display them by block size, from largest to smallest.

Try these PF keys if you would like.

PF11 is also quite useful. It lets you XEDIT any files shown on the FILELIST screen or see a FILELIST display of any directories shown.

For example, suppose you just realized that you wanted to add more data to MARIGOLD STORY. You can easily do so now. Move your cursor to anywhere on the line for MARIGOLD STORY and press PF11.

Your screen will look like this:

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




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

Enter

```
====> b
```

to make the last line of the existing file the current line.

Then enter

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

so the new data will be entered in the file in both capital and lowercase letters. Then enter

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

for input mode.

Then add the following information at the bottom of the file:

```
Some gardeners claim marigolds are also useful for keeping
insects away from other plants.
In fact, marigolds are often planted between rows of tomato
plants to help protect the tomatoes from aphids.
```

Press Enter twice to get out of input mode and return your cursor to the command line. Then enter

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

to save your new version of the file and return to the FILELIST display.

Press PF3 to quit the FILELIST display. The DIRLIST display will return to your screen.

Creating Files within Your Directories

In Chapter 2, "Editing Files," on page 7, you used the XEDIT command to create several files. Because you did not specify a file mode, it defaulted to A. The files were created in your top directory.

To create files in another directory, access the directory with the ACCESS command and assign it a file mode letter. Then use that file mode letter when you create the file.

Suppose you want to create a file called PHOTO TIPS in your LEISURE directory. You can access the LEISURE directory on the DIRLIST display, which is still showing on your screen.

Move your cursor to the Cmd column across from the listing for VMSYSU:*yourid*.LEISURE and enter the following command. Be sure to leave a space before and after the slash.

```
acc / b
```

Your screen will look like this:

```
 yourid   DIRLIST  A0  V 319  Trunc=319 Size=4 Line=1 Col=1 Alt=7
Cmd    Fm Directory Name
*      A  VMSYSU:yourid.
acc / b  VMSYSU:yourid.LEISURE
       -  VMSYSU:yourid.LEISURE.SUMMER
       -  VMSYSU:yourid.LIST




1= Help       2= Refresh  3= Quit   4= Sort(fm)    5= Sort(dir)   6= Auth
7= Backward   8= Forward  9=        10=           11= Filelist   12= Cursor

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

Press Enter to access the LEISURE directory with a file mode of B.

Now your screen will look like this:

```
 yourid    DIRLIST  A0  V 319  Trunc=319 Size=4 Line=1 Col=1 Alt=7
Cmd   Fm Directory Name
*     A  VMSYSU:yourid.
*     B  VMSYSU:yourid.LEISURE
      -  VMSYSU:yourid.LEISURE.SUMMER
      -  VMSYSU:yourid.LIST




1= Help       2= Refresh  3= Quit   4= Sort(fm)    5= Sort(dir)   6= Auth
7= Backward   8= Forward  9=        10=            11= Filelist   12= Cursor

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

Press PF3 to remove the DIRLIST display from your screen.

Now that the LEISURE directory is accessed with a file mode of B, you can specify the file mode letter (B) in many commands. This is much faster than specifying the entire directory name.

To create a file called PHOTO TIPS in the LEISURE directory, enter:

```
 x photo tips b
```

This command creates within your LEISURE directory (which is accessed as B) a file with a file name of PHOTO and a file type of TIPS.

Enter

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

and input the following data into the PHOTO TIPS file:

```
 Check your film speed.
 Monitor the light on the subject.
 Adjust your camera's exposure.
 Focus carefully.
 Squeeze the shutter release slowly.
```

After you are done typing the last line, press Enter twice to get out of input mode and return your cursor to the command line. Then file PHOTO TIPS. Enter:

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

Now your tree looks like this:
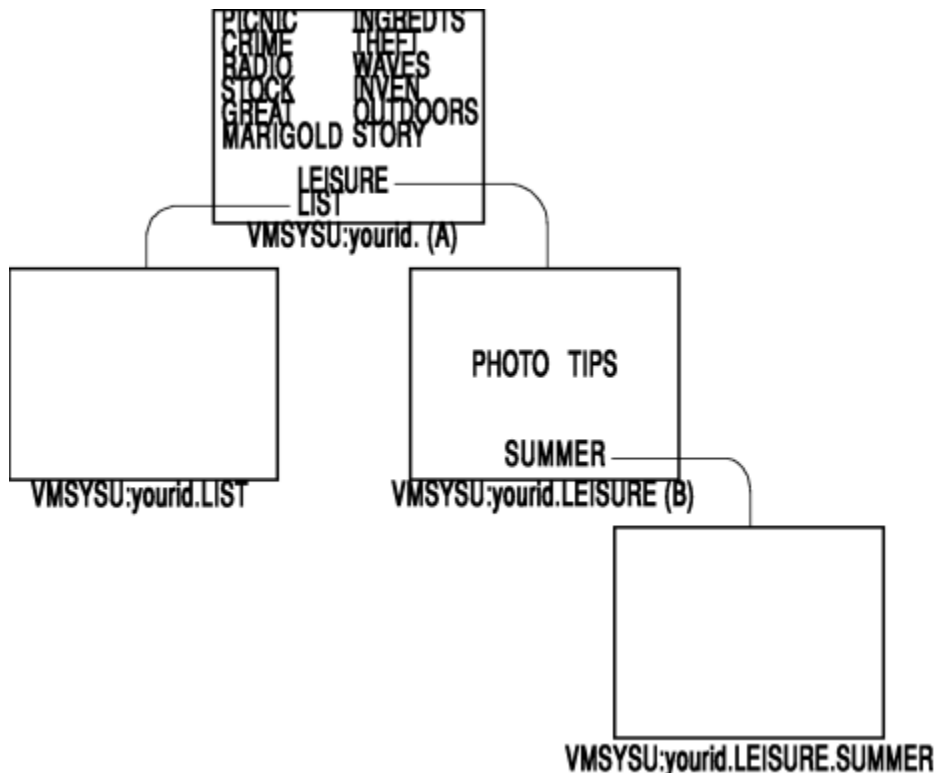
Figure 5. Creating the PHOTO TIPS File

The PHOTO TIPS file is contained in the LEISURE directory. If you want to see a FILELIST display of the LEISURE directory, enter:

```
filel * * b
```

This command will show a FILELIST screen of all the files (regardless of their file names or file types) in the directory accessed with a file mode of B (the LEISURE directory), and any subdirectories. You will see that PHOTO TIPS is included in the list. When you are finished, press PF3 to quit.

Another way to create a file called PHOTO TIPS in the LEISURE directory (file mode B) is with the CREATE FILE command. CREATE FILE creates a new empty file in your SFS directory when you enter:

```
create file photo tips b
```

CREATE FILE also lets you specify the record format, record length, recoverability and overwrite attributes for the files you create. For more information on this command, see *z/VM: CMS Commands and Utilities Reference*.

Copying Files from One Directory to Another

Earlier in this chapter you saw how to create a copy of a file and place the copy in your top directory. If you completed the exercises in Chapter 2, "Editing Files," on page 7, you have several files in your top directory. The COPYFILE command also lets you copy files to your other directories.

The format of the COPYFILE command is:

```
COPYfile filename1 filetype1 filemode1 filename2 filetype2 filemode2
```

- The first file name, file type, and file mode refer to the original file.
- The second file name, file type, and file mode refer to the copy you want to create.

The PICNIC INGREDTS file is currently in your top directory. Because picnicking is a leisure activity, it would also fit nicely into your LEISURE directory.

You can give the copy a new name or use the same name (PICNIC INGREDTS). The system lets you use the same file name and file type for two files when the files are in different directories. For this example, use the same name.

Your top directory is accessed as A; your LEISURE directory is accessed as B. To place a copy of PICNIC INGREDTS in the LEISURE directory, enter the following command:

```
copy picnic ingredts a = = b
```

This command makes a copy of the PICNIC INGREDTS file in your top directory and places the copy in your LEISURE directory, which is accessed as B.

What do the equal signs (= =) mean? The equal signs tell the system to use the same file name and file type that you just specified. There is no need to type PICNIC INGREDTS twice.

Now your tree looks like this:
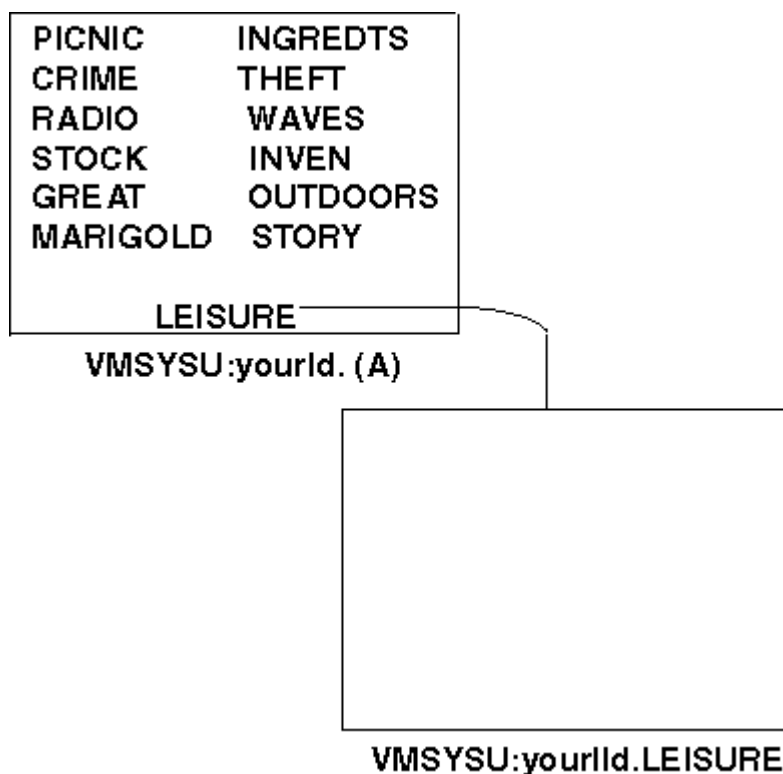
Figure 6. Copying the PICNIC INGREDTS File

Relocating Your Files

Another way to put files into your directories is to relocate them from other directories. You can do so with the RELOCATE command. Unlike COPYFILE, which makes a duplicate copy of a file in a new location, RELOCATE moves the file entirely from one directory to another.

To relocate a file, you specify the file name, file type, and *dirid* (directory name or file mode letter) of the directory you wish to move, followed by the word TO and the *dirid* of the destination directory:

```
RELOcate filename filetype dirid1 TO dirid2
```

You can also relocate a directory. To do so, specify the *dirid* of the directory you wish to move, followed by the word TO and the *dirid* of the destination directory:

```
RELOcate dirid1 TO dirid2
```

Look again at the list of files you created in your top directory:

```
      PICNIC INGREDTS
      CRIME THEFT
      RADIO WAVES
      STOCK INVEN
      GREAT OUTDOORS
      MARIGOLD STORY
```

The STOCK INVEN file seems to be a good candidate for your LIST directory. STOCK INVEN is currently located in your top directory. The name of your top directory is your user ID followed by a period. A shortcut way of referring to your top directory is to specify just the period (with a space before and after it).

The LIST directory is one level below your top directory. Therefore, you can refer to it as the .LIST directory. Because the RELOCATE command will accept a *dirid* (either a directory name or a file mode letter), you can relocate files to the LIST directory even though you have not accessed it to establish a file mode letter for it.

To relocate STOCK INVEN, enter the following command. Be sure to specify the word TO before the name of the destination directory:

```
relo stock inven . to .list
```

Now the STOCK INVEN file is no longer in your top directory; it is in the LIST directory.

Your tree looks like this:
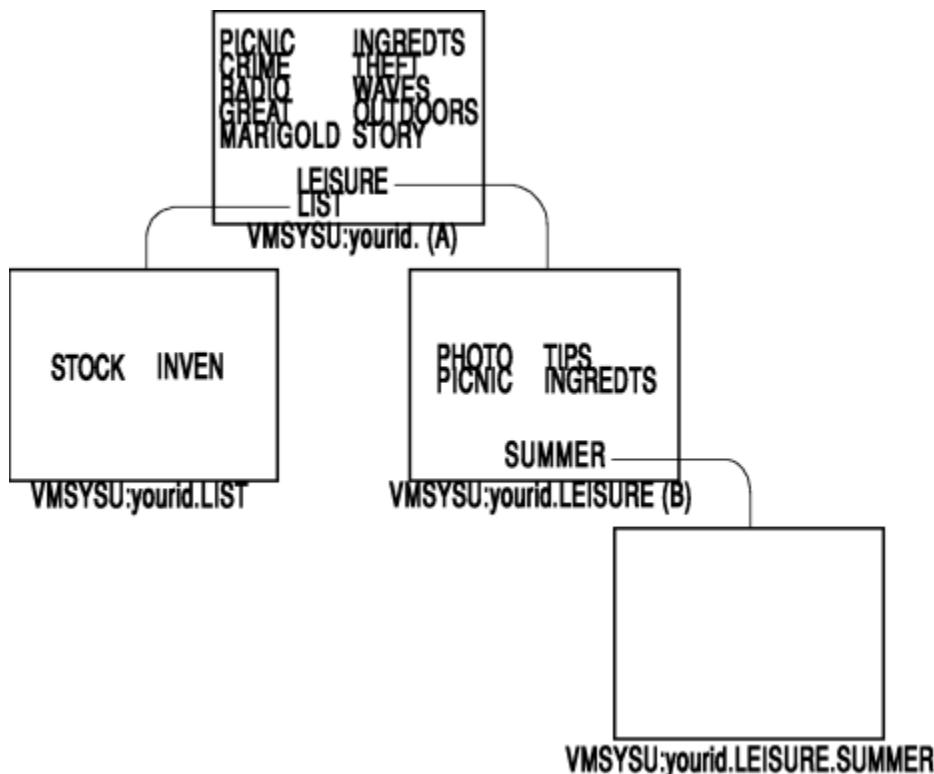
Figure 7. Relocating the STOCK INVEN File

To see the results of the RELOCATE command, you can use another option of the FILELIST command. The SEARCH option of FILELIST will show you every occurrence of a specific file. Enter:

```
filel stock inven a (search
```

Because you entered the command with a file mode of A, the system will begin with your top directory (the directory you have accessed as A), and search each directory below it to locate every occurrence of the STOCK INVEN file.

When the first directory you specify is accessed (has a file mode letter), the SEARCH option of FILELIST searches all the directories below it, regardless of whether they are accessed.

Your screen will look like this:

```
 yourid   FILELIST A0  V 355  Trunc=355 Size=1 Line=1 Col=1 Alt=0
Cmd   Filename Filetype Fm Directory Name
_     STOCK    INVEN    -  VMSYSU:yourid.LIST




















1= Help      2= Refresh  3= Quit   4= Dirlist    5= Sort(name)  6= Auth
7= Backward  8= Forward  9= Alias 10= Filelist   11= XEDIT/LIST 12= Cursor

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

There is only one occurrence of the STOCK INVEN file—in the LIST directory. When you relocated the file, a copy was not retained in your top directory. Press PF3 to quit FILELIST.

You can also choose at any time to relocate an entire directory and all the files it contains. To do so, simply specify the original directory name and the destination directory name. The directory that you relocate then becomes a subdirectory of the destination directory.

For example, you could decide to relocate the LIST directory to place it below the LEISURE.SUMMER directory in your tree. To do so, you would enter:

```
relo .list to .leisure.summer
```

If you entered this command, the LIST directory would be a subdirectory of the LEISURE.SUMMER directory. It would become the .LEISURE.SUMMER.LIST directory.

Would you like to stop for now? If so, this is a good place to end your terminal session. Otherwise, you can continue.

Summary

You learned the following commands in this section:

Command/PF Key	Function
DIRList	Displays a list of your directories.
CREate DIRectory	Creates a new directory.
CREate FILe	Creates a new file.
ERASE	Erases a file or directory.
Rename	Renames a file or directory.
PF11 (XED/FILEL)	Displays the FILELIST screen from DIRLIST; XEDITs a file from FILELIST.
FILEList (STAts	Displays information about your files and directories. It is the default option of FILELIST.

Command/PF Key	Function
COPYfile	Copies a file from one directory to another.
ACcess	Accesses a directory and gives it a file mode letter.
RELOcate	Relocates a file or directory to another directory.
FILEList (SEArch	Displays information about your files.

Sharing Your Files

To share a file with other users, you can grant them authority to read from or write to your files or directories. Other users can also grant you read or write authority to their files or directories.

After another user grants you authority (or you grant authority to them), you can share files by creating an **alias**, another image of a file, or by accessing the directory where the file resides.

Note

If you logged off after the last section and have just logged on again, you will need to re-enter some commands before we continue.

If you entered the following commands at the beginning of this chapter

```
set filepool vmsysu:
acc . a
acc 191 m
```

enter them again now. These commands set up SFS storage for users who usually store their files on a minidisk and re-establish the original CMS session.

All users who logged off will need to re-access the LEISURE directory. (Directories are automatically released at the end of a terminal session.) Enter:

```
acc .leisure b
```

Now you are ready to continue.

Creating an Alias to Another User's File

Until now, we have created only **base files**. You can also create an **alias**. An alias provides you with another means of referring to the information in the base file.

When you create an alias, the base file does not move, nor do you create another copy of it. Unlike the COPYFILE command, which creates a new file, an alias simply provides you with another means of getting to the data in the base file. Both the alias and base file reference the same data; a change to one will be reflected in the other.

Note: For most of your work in CMS you do not need to worry whether a file is a base file or an alias. Most CMS commands will work the same on either type of file.

Aliases are very useful when you share your files with other users. They allow several users to work with the same file and see each other's changes. When you create an alias to another user's file, you can give the alias any name you wish and group it with your own files anywhere you want within your own directory structure.

For example, if you want to allow Margaret to see one of your files, you can grant her authority to your file. She can then create an alias to your file in one of her directories. She can use any name she chooses to refer to the alias.

Note: The following example is dependent on the set up completed by your system administrator.

You have been granted authority to an IBM-supplied file that was shipped with your system. The name of the file is DEEP SEA, and it is owned by the MAINT user ID. The file resides in the MAINT.FISHING directory.

To see what the DEEP SEA file contains, you can access the MAINT.FISHING directory (if you have been given authority to it) and XEDIT the file. Or, you can create an alias to the file and XEDIT the alias. Let us create an alias.

Fishing is a summer leisure activity. Therefore, your LEISURE.SUMMER directory is a logical place to put an alias.

To create an alias, use the CREATE ALIAS command. The format of the command is:

```
CREate ALIas filename1 filetype1 dirid1 filename2 filetype2 dirid2
```

- The first file name, file type, and *dirid* refer to the base file.
- The second file name, file type, and *dirid* refer to the alias.

Although you can use the CREATE ALIAS command by specifying the entire directory name for each file, the easiest way to enter the command is to refer to directories by file mode letter.

You can establish a file mode for the LEISURE.SUMMER directory by accessing it. To find out which file mode letters you have already used, use the QUERY ACCESSED command. Enter:

```
q accessed
```

Your output will look something like this:

```
Mode   Stat  Files  Vdev  Label/Directory
A       R/W      7  DIR   VMSYSU:yourid.
B       R/W      3  DIR   VMSYSU:yourid.LEISURE
S       R/O   1321  190   MNT190
Y/S     R/O    337  19E   MNT19E
```

As you can see, your top directory, VMSYSU:*yourid*., is accessed as A, and VMSYSU:*yourid*.LEISURE is accessed as B. The next available file mode letter is C.

You can access the LEISURE.SUMMER directory with a file mode of C.

Enter:

```
acc .leisure.summer c
```

Now you are ready to create an alias to the DEEP SEA file and place the alias in your LEISURE.SUMMER directory. You can use the name DEEP SEA for your alias or choose a name of your own. Let us call the alias FISH TALES. Enter:

```
create alias deep sea maint.fishing fish tales c
```

This command will create an alias for MAINT's DEEP SEA file and place the alias in your LEISURE.SUMMER directory (accessed with a file mode of C).

Note: When you refer to your own directories, you do not need to specify your own user ID. However, when you refer to another user's directory, you must explicitly state their user ID. (That is why you must include the MAINT user ID as part of the name of the FISHING directory.)

To see that the alias, FISH TALES, has been added to your LEISURE.SUMMER directory, enter:

```
filel * * c
```

You will see a FILELIST display of all the files (regardless of their file names or file types) located in your LEISURE.SUMMER directory (accessed with a file mode of C), and any subdirectories.

You will see that FISH TALES is included in the list. After you are finished with FILELIST, press PF3 to quit.

Because you accessed the LEISURE.SUMMER directory with a file mode of C, you can use the file mode letter to XEDIT the FISH TALES alias:

```
x fish tales c
```

Your screen will look like this:

```
 FISH     TALES    C1  F 80  Trunc=80 Size=1 Line=0 Col=1 Alt=0
Warning: Not authorized to lock file FISH TALES C1




===== * * * Top of File * * *
      |...+....1....+....2....+....3....+....4....+....5....+....6....+....7...
===== THAT REMINDS ME OF THE TIME I ALMOST
===== CAUGHT THE 50-POUND FISH.
===== IT WAS EARLY IN THE MORNING, AND I WAS ALONE.
===== I HAD A SUDDEN BITE ON THE LINE, AND I
===== BEGAN FURIOUSLY REELING IT IN.
===== I HAD GOTTEN THE FISH ALMOST IN THE BOAT,
===== AND IT WAS A BEAUTY!
===== SUDDENLY, THE LINE WENT SLACK.
===== THE FISH WAS GONE!
===== * * * End of File * * *
====> _
                                               X E D I T  1 File
```

Do not worry about the warning notice that appears at the top of your screen. It appears because XEDIT, by default, tries to lock the file for you (to prevent other users from making changes while you are making changes). Because you only have read authority to the alias, you cannot lock it this way. (We will discuss how to lock files later in this chapter.)

You can see the contents of the FISH TALES alias, but you cannot change them because you have only read authority to the base file, DEEP SEA.

Press PF3 to quit from FISH TALES.

If the owner of the DEEP SEA file makes any changes to the data in the file, you will be able to see the changes the next time you XEDIT your alias, FISH TALES.

You can also use the CREATE ALIAS command to create an alias to one of your own files and place the alias in another directory. This way, you can refer to the same information in different directories.

Later, if you want to change the location of any of your files or directories, you can retain the aliases you set up by using the RELOCATE command. RELOCATE moves the files or directories but leaves the aliases unchanged.

Who Is the Owner of This File?

During your work, you can grant various users authority to your files. These users can then create aliases to the files. Other users can also grant you authority to their files, and you can create aliases to those files in your directories.

Later, you may forget who owns a file and what authority the owner has granted you. To check, you can use the SHARE option of the FILELIST command.

Enter the following command:

```
filel fish tales c
```

Your screen will look like this:

```
 yourid   FILELIST A0  V 108  Trunc=108 Size=1 Line=1 Col=1 Alt=0
Directory = VMSYSU:yourid.LEISURE.SUMMER
Cmd   Filename Filetype Fm Format Lrecl   Records    Blocks   Date     Time
_     FISH     TALES    C1 F         80         1         1  5/10/90 10:59:36




















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

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

Because you did not specify any options, you will see the default FILELIST screen, the STATS screen.

Each of the FILELIST screens contains several PF keys you can use to get further information. From FILELIST STATS, you can use PF10 to enter the FILELIST SHARE screen. PF10 is a toggle key. If you press it repeatedly, you will toggle between the FILELIST STATS screen and the FILELIST SHARE screen.

Press PF10 now to see the FILELIST SHARE screen.

```
 yourid   FILELIST A0  V 149  Trunc=149 Size=1 Line=1 Col=1 Alt=3
Directory = VMSYSU:yourid.LEISURE.SUMMER
Cmd   Filename Filetype Fm Owner    Type    R W
_     FISH     TALES    C1 MAINT    ALIAS   X -















1= Help       2= Refresh  3= Quit   4= Cancel      5= Sort(dir)   6= Auth
7= Backward   8= Forward  9= Alias 10= Stats       11= XEDIT/LIST 12= Cursor

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

The column labeled Owner shows you that MAINT is the owner of the base file for your FISH TALES alias. The Type column shows you that FISH TALES is an alias. The last two columns show you what authority you have to the file. There is an X in the first column indicating that you have read authority. The dash in the W column indicates that you do not have write authority to the file.

You can view the FILELIST SHARE screen at any time to determine the owner of a specific file. To get directly to the FILELIST SHARE screen, you can use the SHARE option of FILELIST when you enter the command at the command line. For example, you would have gotten directly to the FILELIST SHARE screen if you entered:

```
 filel fish tales c (share
```

For now, do not quit the FILELIST SHARE screen.

Determining Who Has Aliases

Another way to find out information on aliases is by using the ALIALIST command from FILELIST.

You can use the ALIALIST command:

- To see what aliases you have to other people's files

- To see information on aliases to your own files.

You can enter the ALIALIST command from the command line on the FILELIST screen or by using PF9 shown on the FILELIST SHARE or FILELIST SEARCH screen.

The FILELIST SHARE display is currently showing on your screen. Your cursor is located on the line for the FISH TALES file. Press PF9 to see an ALIALIST display.

```
 yourid   ALIALIST A0  V 190  Trunc=190 Size=1 Line=1 Col=1 Alt=0
Alias = FISH TALES VMSYSU:yourid.LEISURE.SUMMER
Userid   Num Filename Filetype Directory
MAINT      1 DEEP     SEA      .FISHING




















1= Help       2= Refresh 3= Return   4= Sort(type) 5= Sort(name)  6= Sort(dir)
7= Backward    8= Forward 9= S(user) 10=            11=            12=

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

The information on the ALIALIST screen is similar to what you see on the FILELIST SHARE screen. It shows you that MAINT is the owner of the base file, DEEP SEA, and that you have one alias called FISH TALES.

If you enter the ALIALIST command on a base file that you own, the aliases listed would include your own aliases and any aliases other users have to your file.

Press PF3 to quit the ALIALIST screen. Press PF3 again to quit FILELIST.

Note: If you want to find out information on aliases without getting into FILELIST, you can do so by entering QUERY ALIAS (followed by the file name, file type, and *dirid*) at the CMS command line.

Accessing Another User's Directory

An alias is a permanent addition to your directory (until you delete the alias or the base file). If you only want to work temporarily with another user's file, you may not want to create an alias. Instead, you can access the directory where the file resides.

You can access another user's directory if he or she grants you authority to it. If the user also grants you authority to files in the directory, you can work with those files after accessing the directory.

Note: The following example is dependent on the set up completed by the system administrator.

You have been granted read authority to the MAINT.FELINE directory, another directory owned by the MAINT user ID and shipped with your system. You have also been granted authority to a file it contains— the CAT TALES file.

To determine what authority you have been granted for the CAT TALES file, access the MAINT.FELINE directory with a file mode of D (or the next available file mode). Enter:

```
acc maint.feline d
```

Now you can see a FILELIST SHARE display of the CAT TALES file to determine what authority you have for the file. Enter:

```
filel cat tales d (share
```

Your screen will look like this:

```
yourid   FILELIST A0  V 149  Trunc=149 Size=1 Line=1 Col=1 Alt=0
Directory = VMSYSU:MAINT.FELINE
Cmd   Filename Filetype Fm Owner    Type    R W
_     CAT      TALES    D1 MAINT    BASE    X -












1= Help       2= Refresh  3= Quit   4= Cancel      5= Sort(dir)   6= Auth
7= Backward   8= Forward  9= Alias 10= Stats      11= XEDIT/LIST 12= Cursor

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

As you can see, you have read authority to CAT TALES. Therefore, you can XEDIT the file to view the data it contains. However, because you only have read authority, you cannot make changes to the file.

Press PF11 to XEDIT the file. Once again, ignore the warning message that appears. When you are finished reading the file, press PF3 to quit. Press PF3 again to remove the FILELIST display from your screen.

When you are finished working with the file, you no longer need to remain accessed to the MAINT.FELINE directory. Use the file mode letter you used to access the directory (file mode D) and release it.

Enter the RELEASE command as follows:

```
rel d
```

Note: If you forget the names of the directories to which another user granted you authority, you can easily find out what they are. Enter DIRLIST followed by the other person's user ID and a period. This will show all the directories, within the other user's top directory, to which you have authority. For example, to determine which directories MAINT has granted you authority to, you could enter:

```
dirl maint.
```

You would see a DIRLIST display of the MAINT.FISHING and MAINT.FELINE directories. You will also see the MAINT. and MAINT.SAMPLE directories, other directories to which MAINT has granted you authority. (We will not use MAINT. or MAINT.SAMPLE in these exercises.) When you are finished, press PF3 to quit DIRLIST.

Granting Other Users Authority to Your Files and Directories

When you create a file in one of your directories, as the owner, you automatically have the authority to read from and write to the file. You also can grant other users read or write authority to any of your base files or to any of your directories.

You can grant another user authority to one of your files or directories by using the GRANT AUTHORITY command.

To grant authority on a file, specify the file name, file type, and *dirid* (directory name or file mode) followed by the word TO, the user ID, and the type of authority you wish to grant:

```
GRAnt AUThority filename filetype dirid TO userid (authority
```

To grant authority on a directory, specify the *dirid*, followed by TO, the user ID, and the type of authority:

```
GRAnt AUThority dirid TO userid (authority
```

To grant another user write authority to your file or directory, follow the command with the option **(WRI** for write. Note that write authority includes read authority. If you do not specify an option, the default is read authority.

Only the person who owns a base file or directory can grant authority to another user. For example, if you grant Lisa authority to read one of your files, Lisa cannot extend that authority to Jim.

If you grant another user authority to one of your directories, that **does not imply** that he or she has any authority to the files within the directory. To grant authority on a directory and all the files it contains, you have to use two commands:

```
GRAnt AUThority dirid TO userid (authority
GRAnt AUThority * * dirid TO userid (authority
```

- The first command grants authority on the directory.
- The second command grants authority on all the files within the directory.

You can also grant NEWREAD and NEWWRITE authority on your SFS directories. A user with NEWREAD authority is automatically granted read authority to any file that is added to the directory after the authority is granted. A user with NEWWRITE authority is automatically granted write authority to any file that is added to the directory after the authority is granted. For example, to let user NANCY read any file that you add to your .TEST directory, you would enter:

```
grant authority .test to nancy (newread
```

To grant **everyone in your file pool** authority to one of your files, specify PUBLIC instead of a user ID:

```
GRAnt AUThority filename filetype dirid TO PUBlic (authority
```

To grant public authority on a directory, use this format:

```
GRAnt AUThority dirid TO PUBlic (authority
```

Note: If an external security manager is active on your system, you may need to use different commands to grant authority on your files or directories. For further information on the appropriate commands, see your system administrator.

Before you begin granting authority on your files and directories, see the following table to see what these authorities mean:

Type	File	Directory
READ	Other users can read, copy, browse, print, or create aliases to your file but cannot change or erase it.	Other users can see the names of the files and subdirectories in your directory, but cannot see the contents of the files and subdirectories unless you also grant authority on each file or subdirectory.

Type	File	Directory
WRITE	Other users can read, copy, browse, print, or create aliases to your file, and can also change your file.	Other users can create a new file in your directory (a base file) or aliases to their own files. (this will allow you to share those files.) they can erase or rename aliases within your directory if you also grant read authority on the base files; they can erase or rename base files if you also grant write authority on the files.
NEWREAD	N/A	Other users are granted read authority to any file that is added to the directory in the future.
NEWWRITE	N/A	Other users are granted write authority to any file that is added to the directory in the future.

The above authorities apply only to file control directories and to files within file control directories. By default the directories you create are file control directories.) For more information about the authorities that apply to directory control directories, see *z/VM: CMS User's Guide*.

To practice granting authority to another user, grant the user MAINT read authority to the CRIME THEFT file, which you created in your top directory (accessed as A). Enter:

```
grant auth crime theft a to maint
```

You do not need to specify the READ option because the default of the GRANT AUTHORITY command is READ.

After you enter this command, MAINT will be able to create an alias to your CRIME THEFT file and XEDIT it to see what the file contains. However, MAINT cannot change or erase the CRIME THEFT file. MAINT will retain authority to the file until you revoke the authority you granted. We will discuss revoking authority in "Revoking Authority" on page 88.

Note: An SFS administrator or a user with file pool administration authority can read from or write to your files or directories. Administrators should use this authority only in special situations. You should, however, be aware that an administrator can read from or write to your files and directories even though you have not granted that person authority to do so.

Determining Who Has Authority

At any time, you can check to see what authority you have granted others for a file or directory. You can do so by using the AUTHLIST command from FILELIST or DIRLIST.

Like ALIALIST, which we discussed earlier, you can enter the AUTHLIST command:

- By using the PF6 key displayed on the FILELIST SHARE, FILELIST SEARCH, or DIRLIST screen
- From the command line on the FILELIST or DIRLIST screen.

To see a FILELIST SHARE display of the CRIME THEFT file, enter:

```
filel crime theft a (share
```

Your screen will look like this:

```
 yourid   FILELIST A0  V 149  Trunc=149 Size=1 Line=1 Col=1 Alt=0
Directory = VMSYSU:yourid.
Cmd   Filename Filetype Fm Owner    Type    R W
_     CRIME    THEFT    A1 yourid    BASE    X X




1= Help       2= Refresh  3= Quit   4= Cancel     5= Sort(dir)   6= Auth
7= Backward   8= Forward  9= Alias 10= Stats     11= XEDIT/LIST 12= Cursor

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

Your cursor is located on the line for CRIME THEFT. Press PF6 to see an AUTHLIST display.

Your screen will look something like this:

```
 yourid   AUTHLIST A0  V 165  Trunc=165 Size=2 Line=1 Col=1 Alt=0
File = CRIME THEFT VMSYSU:yourid.
Grantee  R  W
MAINT    X  -
yourid   X  X




1= Help       2= Refresh  3= Return 4= S(Grantee)  5= Sort(W)     6=
7= Backward   8= Forward  9=        10=            11=            12=

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

In the column labeled `Grantee`, you see two user IDs. One is the MAINT user ID. There is an X to show that MAINT has read authority to the file. The dash in the `W` column shows that you did not grant write authority to MAINT.

Your own user ID is listed because you are the owner of the CRIME THEFT file. There is an X in both the R and W columns to show that you have both read and write authority to your own file.

When you use the AUTHLIST command for a directory, two additional columns having the headings NR and NW are displayed. These columns indicate whether the user has NEWREAD or NEWWRITE authority on the directory.

If you wish, you can use the PF4 and PF5 keys shown on the AUTHLIST screen to sort the display:

• PF4 sorts the display alphabetically by grantee.

• PF5 sorts grantees with write authority to the top of the list, followed by grantees with read authority.

For now, press PF3 to return to the FILELIST SHARE screen. Then press PF3 again to quit the FILELIST SHARE screen.

Note: If you want to find out information on authority without getting into FILELIST, you can do so by entering QUERY AUTHORITY (followed by the file name, file type, and *dirid*) at the CMS command line.

After you have granted other users authority to your files or directories, you may decide to change the location of the files or directories within your tree. To retain any authority you granted other users, use the RELOCATE command. RELOCATE moves files or directories but leaves the authorities unchanged.

Revoking Authority

Suppose that now you no longer want MAINT to have read authority to the CRIME THEFT file. You can revoke any authority you granted on a file or directory with the REVOKE AUTHORITY command.

To revoke all authority on a file, specify the file name, file type, and *dirid* (directory name or file mode) followed by the word FROM and the user ID:

```
REVoke AUThority filename filetype dirid FROM userid
```

To revoke all authority on a directory, (including NEWREAD and NEWWRITE authority), specify the *dirid* followed by FROM and the user ID:

```
REVoke AUThority dirid FROM userid
```

Like the GRANT AUTHORITY command, REVOKE AUTHORITY on a directory only revokes another user's authority **on the directory**.

To revoke another user's authority on a directory and all the files it contains, you have to use two commands:

```
REVoke AUThority dirid FROM userid
REVoke AUThority * * dirid FROM userid
```

- The first command revokes authority on the directory.
- The second command revokes authority on the files within the directory.

To revoke authority from all the users to whom you granted PUBLIC authority, use this command format:

```
REVoke AUThority filename filetype dirid FROM PUBlic
```

For example, specifying PUBLIC will remove authority from everyone that you granted authority to by specifying PUBLIC. This command will not revoke authority from any users to whom you individually granted authority.

If you want to revoke authority from **everyone in your file pool**, use the ALL option:

```
REVoke AUThority filename filetype dirid FROM ALL
```

Note: Remember that you will not be able to revoke authority from the SFS administrator or a user with file pool administration authority. Although administrators will not usually read from or write to your files, you cannot revoke their authority to do so.

To revoke the authority you granted MAINT on CRIME THEFT, enter:

```
revoke auth crime theft a from maint
```

To check to see if the command worked, enter:

```
filel crime theft a (share
```

Press PF6 to view an AUTHLIST display of the CRIME THEFT file.

```
 yourid    AUTHLIST A0  V 165  Trunc=165 Size=1 Line=1 Col=1 Alt=0
File = CRIME THEFT VMSYSU:yourid.
Grantee  R  W
yourid   X  X




















1= Help       2= Refresh  3= Return 4= S(Grantee)  5= Sort(W)     6=
7= Backward   8= Forward  9=      10=            11=           12=

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

As you can see, you are now the only user with authority to the CRIME THEFT file. Press PF3 to return to the FILELIST SHARE screen. Then press PF3 again to remove the FILELIST SHARE display from your screen.

Note: If an external security manager is active on your system, you may need to use different commands to revoke authority on your files or directories. For further information on the appropriate commands, see your system administrator.

This is another place where you might want to stop or end your terminal session. If you wish, you can continue with the last section of this chapter.

Summary

You learned the following commands in this section:

Command	Function
CREate ALIas	Creates an alias to a file.
ALIalist	Lists information on aliases.
FILEList (SHAre	Displays information about your files and directories.
GRAnt AUThority	Grants another user authority to a file or directory.
AUThlist	Lists information on authorities.
REVoke AUThority	Revokes the authority granted on a file or directory.

Locking Your Shared Files

If you have granted another user write authority to one of your files, or if you have write authority on another user's files, you need to understand how CMS handles multiple users working with the same file.

You do not usually need to worry about locking your files and directories. CMS prevents other users from changing a shared file while commands are processed. For example, another user could not make changes to a file at the same time you are copying or renaming it. Files are also locked automatically when you XEDIT them.

You may, however, want to lock a file or directory if you need to complete a series of commands. For example, you might need to copy a file, XEDIT and change the copy, and compare the two files. In such a

situation, a lock on the file or directory prevents other users with authority from changing the file before you complete your work.

Note.

If you logged off after the last section and have just logged on again, you will need to re-enter some commands before we continue.

If you entered the following commands at the beginning of this chapter

```
set filepool vmsysu:
acc . a
acc 191 m
```

enter them again now to set up SFS storage for users who usually store their files on a minidisk and to re-establish the original CMS session.

All users who logged off will need to re-access the LEISURE and SUMMER directories. Enter:

```
acc .leisure b
acc .leisure.summer c
```

Now you are ready to continue.

Creating a Lock on a File or Directory

When you lock a file, other users cannot make changes to it. When a directory is locked, no files within the directory can be changed by other users.

You can lock a file or directory by using the CREATE LOCK command.

To create a lock on a file, specify the file name, file type, and *dirid*, followed by the type of lock and the duration:

```
CREate LOCk filename filetype dirid SHAre|EXClusive|UPDate SESsion|LASting
```

To lock a directory, specify the *dirid* followed by the type of lock and the duration:

```
CREate LOCk dirid SHAre|EXClusive|UPDate SESsion|LASting
```

When you lock a file or directory, you must specify the duration of the lock. A SESSION lock is automatically unlocked at the end of your CMS session or when communication with your file pool is broken. A LASTING lock can only be removed with the DELETE LOCK command.

If you have read authority to a file or directory, you can only create a SHARE lock on that file or directory. If you have write authority, you can create any type of lock (SHARE, EXCLUSIVE, or UPDATE) on the file or directory.

The following table shows the meaning of each type of lock for a file or directory.

Type	File	Directory
UPDATE	Other users may read the file while you are reading or updating it. They cannot change, rename, relocate, or erase the file.	The person who locked the directory can read from or write to any files for which that person is authorized. Other users may only read files in the directory; they cannot write to the files (even if they have write authority to them). Nor can they create, delete, rename, or relocate any base file, alias, or subdirectory in the directory.

Type	File	Directory
EXCLUSIVE	Other users cannot read from or write to the file.	Other users cannot read from or write to any of the base files or aliases in the directory. Also, they cannot create, erase, rename, or relocate any base files or aliases until the directory is unlocked.
SHARE	Other users can read the file while you are reading it. No one, including the person who locked the file, can update the file until it is unlocked. Also, no users can rename, relocate, or erase the file.	Other users and the person who locked the directory can read from files in the directory (if they are authorized to do so). No users (including the person who created the lock) can update any files or subdirectories in the directory. Nor can they create, erase, rename, or relocate any base files, aliases, or subdirectories until the directory is unlocked.

Note: If you create a lock on an alias, it is the same as locking the base file—the owner of the base file and other users with aliases to the file will be affected by the lock.

Except for UPDATE the above lock types apply only to file control directories and to files within file control directories. (Remember that, by default, the directories you create are file control directories.)

When Should I Lock a File?

If you only want to XEDIT a file and make changes, you do not need to lock the file. Because XEDIT prevents other users from changing the file at the same time you are editing, a lock is not necessary. However, at other times, you may find it useful to lock your files.

Suppose you grant another user write authority to the file PHOTO TIPS. Now you plan to copy the file, XEDIT and change the copy, and then compare the two versions of the file. As you can imagine, it could be quite confusing if the other user decides to make changes to the PHOTO TIPS file before you complete your work.

To eliminate the possibility for this sort of confusion, you can lock the file (or the directory where it resides) before you begin your work.

Create an UPDATE SESSION lock on the PHOTO TIPS file, which is located in your LEISURE directory. This will allow other users with authority to the file to read but not change the file.

Enter:

```
create lock photo tips .leisure upd ses
```

Now you can complete your work. Enter

```
copy photo tips b = = c
```

to create a copy of the PHOTO TIPS file in your LEISURE.SUMMER directory (accessed as C).

To XEDIT the copy you just created, enter:

```
x photo tips c
```

Enter

```
====> b
```

to make the last line of the existing file the current line. Enter

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

for input mode and add some new information at the bottom of the file.

When you are finished, press Enter twice to get out of input mode. Then enter

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

to save the updated version of the file.

Determining If a File or Directory is Locked

While you can lock files you share to prevent simultaneous changes, other users with write authority to the files have the same option. You may find yourself unable to change a file if someone else locked it. In fact, if the other user locked the file with an EXCLUSIVE lock, you would not even be able to read the file.

If another user has write authority to the directory where a file resides, he or she can choose to lock the entire directory. Then, you could not change any file in the directory.

You can determine who has locked the file or directory by using the QUERY LOCK command.

To determine if a file is locked, specify the file name, file type, and *dirid*

```
Query LOCK filename filetype dirid
```

To query a directory lock, specify only the *dirid*:

```
Query LOCK dirid
```

You currently have an UPDATE SESSION lock on the PHOTO TIPS file in your LEISURE directory. It has been a while since you stopped working on PHOTO TIPS. Suppose you want to ensure that you locked the file earlier. You can query your own lock on it. Enter:

```
q lock photo tips .leisure
```

This command will query the lock status for the PHOTO TIPS file in your LEISURE directory. You will see:

```
Directory = VMSYSU:yourid.LEISURE
Filename Filetype Fm Type    Userid   Lock       Duration
PHOTO    TIPS     B1 BASE    yourid   UPDATE     SESSION
```

The screen shows that the file has an UPDATE SESSION lock. Your user ID is shown because you locked the file.

Other users who share your files may also use the QUERY LOCK command. For example, if Anne has write authority to your PHOTO TIPS file, she may find she can only read the file because a lock is in effect. To determine who has the file locked, she can use the QUERY LOCK command. After determining that you locked the file, Anne can call or send you a message and ask you when you plan to complete your work and unlock the file.

Although other users with authority to the file can read the contents of PHOTO TIPS, they cannot change the file until you unlock it. Therefore, you can complete your work.

Note: Remember that although your file is locked, an SFS administrator has the authority to delete the lock and read from or write to your file. Although administrators rarely use this authority, you should be aware that they can do so.

You have already copied the PHOTO TIPS file to your LEISURE.SUMMER directory and made changes to the new version of the file. Let us compare your new version of PHOTO TIPS (on file mode C) with the original version (on file mode B). Enter:

```
filel
```

You will see a FILELIST display of the files and directories within your top directory (accessed with a file mode of A). Position your cursor in the Cmd column on the line for the LEISURE directory and press PF11. You will see a FILELIST display of all the files in the LEISURE directory (accessed with a file mode of B). Then position your cursor on the line for PHOTO TIPS.

Enter:

```
compare / photo tips c
```

This command will compare the new copy of PHOTO TIPS (on file mode C) to your original version of PHOTO TIPS (on file mode B). Your screen will display the message

```
Premature EOF on file PHOTO TIPS B
```

because you added additional information to the copy of the file on file mode C. Press the CLEAR key to return to FILELIST. Then press PF3 *twice* to quit the FILELIST screens.

Unlocking Files and Directories

Now that you are finished working with the PHOTO TIPS file, you can free the lock. You locked the file with a SESSION lock. Therefore, you can free the lock by logging off (because a SESSION lock only lasts the duration of your CMS session). Or, you could use the DELETE LOCK command.

To delete a lock on a file, use this format:

```
DELete LOCk filename filetype dirid
```

To unlock a directory, specify:

```
DELete LOCk dirid
```

To free the lock on PHOTO TIPS, enter:

```
delete lock photo tips .leisure
```

Query the lock again to ensure that the file is unlocked:

```
q lock photo tips .leisure
```

The following message will appear:

```
No locks are held on PHOTO TIPS VMSYSU:yourid.LEISURE
```

Because the file is no longer locked, Anne can now XEDIT the file and update it with new data.

Concluding Your SFS Work

You are now finished with the Shared File System (SFS) exercises. If your files are stored in an SFS file pool, you can continue with the next chapter.

If your files were stored on a minidisk and you began storing your files in SFS file pool at the beginning of this section, you can decide whether to continue using the SFS file pool. If you wish to continue storing your files in an SFS file pool, simply go on to the next chapter. If you wish to revert to a minidisk as your primary storage, enter the following command:

```
acc 191 a
```

This command will access your 191 minidisk (which you accessed as M at the start of this chapter) with a file mode of A. This resets the minidisk for storing any new files you create.

Summary

You learned the following commands in this section:

Command	Function
CREate LOCk	Locks a shared file or directory.
Query LOCK	Determines if a shared file or directory is locked.
DELete LOCk	Deletes a lock on a shared file or directory.

For more information on using the commands discussed in this chapter, see *z/VM: CMS Commands and Utilities Reference*. For additional details on using the Shared File System (SFS) to organize and share your files, see *z/VM: CMS User's Guide*.

Chapter 4. Printing Files

If you need to print a file, this chapter teaches you how to:

- Tell the system how many copies you want printed.
- Use the PRINT command to print any file.

Where Do I Enter These Commands?

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

PRINT Command

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

```
PRint filename filetype filemode
```

If the file is on file mode A (the directory or minidisk accessed with a file mode of A), you do not need to type a file mode. When you press Enter, your command moves up on the screen and you get a ready message. Your file is now at the printer.

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

```
SPool PRINT COPY 3
```

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

```
PRint filename filetype filemode
```

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

```
PRT FILE 0765 SENT TO   RSCS   RDR AS  5026 RECS 002 CPY  003 NOHOLD NOKEEP
```

The number 0765 is the number that z/VM assigned to the file in the example. CPY 003 means that you will get three copies. The rest of the information is not important at this time and is beyond the scope of this document. The message means your file is at the printer.

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

Using PRINT on the FILELIST Screen

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

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

To get more than one copy, first enter a SPOOL PRINT COPY *n* command. Type it on the large arrow command line (at the bottom of the FILELIST display) and then press Enter.

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

```
 SNOWHITE FILELIST  A0  V 108  Trunc=108 Size=10 Line=1 Col=1 Alt=0
Directory = VMSYSU:yourid.DWARFS
Cmd   Filename Filetype Fm Format Lrecl Records Blocks   Date      Time
      ALL      NOTEBOOK A0 V          95      72      3  5/04/90 21:15:04
      WISTFUL  DATA     A1 V          95      97      4  5/04/90 21:12:04
      BOSS     DATA     A1 V          95      98      4  5/04/90 20:58:07
      DUMMY    DATA     A1 V         107     110      5  5/04/90 17:59:00
print GROUCHY  DATA     A1 V          92     146      7  5/02/90 15:33:05
      PRINCE   DATA     A2 V          75      66      2  4/25/90 12:10:03
      SNOOZY   DATA     A2 V         120     188      7  4/24/90  9:14:02
      SNIFFLES DATA     A1 V          26     150      7  4/23/90  8:50:06
      MINING   SCRIPT   A1 V         132      63      2  3/26/90  7:05:08
      KITCHEN  SCRIPT   A1 V         132     100      5  3/26/90  6:05:08




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

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

When you press Enter, a message like this appears on a cleared screen:

```
 PRT FILE 0785 FOR SNOWHITE COPY 001 NOHOLD
```

This message means that the file GROUCHY DATA is at the printer.

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

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

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

Summary

You learned the following commands in this chapter:

Command	Function
SPool	Specifies number of copies to be printed.
PRint	Prints a file.

For more information on the PRINT command, see *z/VM: CMS Commands and Utilities Reference*.

Chapter 5. Communicating with Other Users

This chapter teaches you how to:

- Put information about the people with whom you communicate in a special file of *names.*
- Send *messages* to someone who is logged on to your system or another system.
- Prepare and send *notes* to one person or to many people on a list.
- Send *files* to other people.
- *Receive* notes and files from other people.

You should know how to edit files using XEDIT (Chapter 2, "Editing Files," on page 7) before reading this chapter.

Keeping Track of Names

You need to know two things about other users to communicate with them:

- Their *user ID* (user identification)
- Their *node* (computer identification).

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

To find out a person's user ID, just ask. (A user ID is sometimes called a logon id.) If you are enrolled in the Shared File System, you can find out the user IDs that are enrolled in your file pool by typing:

```
query enroll user for all
```

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

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

What Is a Names File?

A names file contains the user IDs and nodes of the people you communicate with. It can also contain other information, like mailing addresses and phone numbers. After you make an entry for a person in your names file, you no longer have to think about the person's user ID or node.

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

How to Create a Names File

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

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

```
names
```

You will see the following screen. (When you enter NAMES, your user ID appears (automatically) in the first line instead of *userid*.)

```
 ====> NAMES (Mail panel)      File: userid   NAMES A0               <====
Fill in the fields and press a PFkey to display and/or change your names file
Nickname:                      Notebook:
Userid:
  Node:
                      Name:
                     Phone:
                   Address:
                         :
                         :
                         :
            List of Names:
                         :
                         :
                         :
                         :
 Tag:              Value:
 Tag:              Value:

 1= Help     2= Add      3= Quit     4= Clear     5= Find       6= Change
 7= PrevNick 8= NextNick 9= Delete  10= PrevScrn  11= NextScrn  12= Cursor
                                ======> Screen 1 of 1 <======
 ====>
                                                Macro-read 1 File
```

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

When you create your names file, the file will, by default, be created on file mode A (your top directory or minidisk accessed with a file mode of A), with your user ID as the file name, and **names** as the file type.

Creating a Sample Names File

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

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

All these people had user IDs.

To create a names file, *fill in a screen for each entry.* The first entry you create should be for yourself. (You probably do not want to communicate with yourself. But the information in your entry is used for note headings, which are discussed later in this chapter.)

This is Snow White's entry:

```
 ====> NAMES (Mail panel)      File: SNOWHITE NAMES A0               <====
Fill in the fields and press a PFkey to display and/or change your names file
Nickname: SNOW                 Notebook:
Userid: SNOWHITE
  Node: FOREST
                      Name: Snow White
                     Phone: 555-5555
                   Address: 201 Redwood Lane
                         : Forest Primeval, California
                         :
                         :
            List of Names:
                         :
                         :
                         :
                         :
 Tag:              Value:
 Tag:              Value:

 1= Help     2= Add      3= Quit     4= Clear     5= Find       6= Change
 7= PrevNick 8= NextNick 9= Delete  10= PrevScrn  11= NextScrn  12= Cursor
                                ======> Screen 1 of 1 <======
 ====>
                                                Macro-read 1 File
```

The following section explains the fields shown on the screen.

```
Nickname:   Userid:    Node:
```

You **must** fill in the Nickname and Userid fields. The nickname can be any name you want of 21 or less letters. You must also fill in the Node field if the person's node is different from yours. If the person's node is the same as yours, you can leave the Node field blank. You can type these fields using capital or lowercase letters. They will be changed to capitals automatically.

```
Name:  Phone:  Address:
```

These fields are optional. But the more information you can supply, the better, because it can be used in note headings. You can type these fields in capital or lowercase letters or a combination of the two. They remain the same way you type them.

```
Notebook:
```

We will discuss this field later in this chapter.

```
List of Names:
```

We will discuss this field later in this chapter.

```
Tag:  Value:
```

These fields are optional. You can use them to save additional information about someone.

```
Tag:
```

is a label you choose.

```
Value
```

is its value. For example, in Snoozy's entry you might add:

```
Tag: Job title    Value: Second class miner
```

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

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

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

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

```
 ====> NAMES (Mail panel)      File: SNOWHITE NAMES A0                 <====
Fill in the fields and press a PFkey to display and/or change your names file
Nickname: SNOOZY                  Notebook:
Userid: SNOOZY
  Node: COTTAGE
                    Name: I. M. Dozing
                   Phone: 777-7777
                 Address: Dwarf Cottage
                        : Forest
                        :
                        :
```

```
 ====> NAMES (Mail panel)      File: SNOWHITE NAMES A0                  <====
Fill in the fields and press a PFkey to display and/or change your names file
Nickname: GROUCHY                Notebook:
Userid: GROUCHY
  Node: COTTAGE
                      Name: E. B. Scrooge
                     Phone: 777-7777
                   Address: Dwarf Cottage
                         : Forest
                         :
                         :
```

```
 ====> NAMES (Mail panel)      File: SNOWHITE NAMES A0                  <====
Fill in the fields and press a PFkey to display and/or change your names file
Nickname: WITCH                  Notebook:
Userid: QUEEN
  Node: CASTLE
                      Name: Bad Queen
                     Phone: UGLY-1111
                   Address: Vanity Lane
                         : Mirror City
                         :
                         :
```

Using the Notebook Field

In this entry, the Notebook field is filled in:

```
 ====> NAMES (Mail panel)      File: SNOWHITE NAMES A0                  <====
Fill in the fields and press a PFkey to display and/or change your names file
Nickname: GORGEOUS               Notebook: PRIVATE
Userid: PRINCE
  Node: ATLARGE
                      Name: Prince Charming
                     Phone: Area 111 111-1111
                   Address:
                         :
                         :
                         :
           List of Names:
                         :
                         :
                         :
                         :
 Tag:              Value:
 Tag:              Value:

 1= Help     2= Add      3= Quit    4= Clear     5= Find      6= Change
 7= PrevNick 8= NextNick 9= Delete  10= PrevScrn 11= NextScrn  12= Cursor
                               ======> Screen 1 of 1 <======
 ====>
                                          Macro-read 1 File
```

Usually, a copy of all notes that you send and receive are saved automatically. They are saved in a file whose file identifier is ALL NOTEBOOK. Fill in the Notebook field only if you want a copy of all notes to and from this person to be saved in a separate file. The file name will be any name up to eight characters that you type in the Notebook field. Its file type will be NOTEBOOK.

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

Creating an Entry for a List of Names

If you often send messages, notes, or files to the same group of people, you can create an entry for a list of names (which is like a *distribution list*).

If you use Shared File System (SFS) commands to share files, you might also find such a list useful in granting or revoking authority on your files and directories. For example, if you want to share files with

all seven dwarfs, you will find it easier to issue one command (rather than seven) to grant all the dwarfs authority to a file.

The nickname you type in the Nickname field is the name of the list, not of just one person. You type the nickname of each person in the list in the List of Names field.

Here is an entry for a list:

```
 ====> NAMES (Mail panel)      File: SNOWHITE NAMES A0                  <====
Fill in the fields and press a PFkey to display and/or change your names file
Nickname: DWARFS              Notebook:
Userid:
  Node:
                    Name:
                   Phone:
                 Address:
                        :
                        :
                        :
          List of Names: snoozy dummy boss smiley grouchy sniffles wistful
                        :
                        :
                        :
                        :
 Tag:            Value:
 Tag:            Value:

 1= Help      2= Add       3= Quit     4= Clear      5= Find       6= Change
 7= PrevNick  8= NextNick  9= Delete  10= PrevScrn  11= NextScrn  12= Cursor
                                 ======> Screen 1 of 1 <======
 ====>
                                            Macro-read 1 File
```

The nickname of this list is DWARFS. The nickname of each dwarf is typed in the List of Names field. You can type the nicknames in either capital or lowercase letters. Do not use commas to separate the names in the list. Each dwarf has his own entry in the names file.

What if a person in the list does not have a nickname in the names file? You can still specify the names of people not in the names file as part of the list. If the person shares your computer, type that person's user ID in the list. If the person is at a different node than you, type that person's user ID and node in the form *userid* at *node* in the list. Remember to press PF2 to add the entry to the file.

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

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

Entering Chained Lists of Names

You may need to send a message to more than one List of Names. This is done using a chained List of Names. You enter nicknames of other lists in the List of Names field, and then give that names entry a nickname also.

For example, we have seen Snow White's list of dwarfs with a nickname of DWARFS. She also has a list of forest friends called ANIMALS containing a list of names shown in the following example.

```
 ====> NAMES (Mail panel)      File: SNOWHITE NAMES A0                 <====
Fill in the fields and press a PFkey to display and/or change your names file
Nickname: ANIMALS             Notebook:
Userid:
  Node:
                    Name:
                   Phone:
                 Address:
                        :
                        :
                        :
         List of Names: BEAR LION MONKEY
                        :
                        :
                        :
                        :
 Tag:              Value:
 Tag:              Value:

 1= Help      2= Add       3= Quit     4= Clear      5= Find       6= Change
 7= PrevNick  8= NextNick  9= Delete  10= PrevScrn  11= NextScrn  12= Cursor
                               ======> Screen 1 of 1 <======
 ====> _
                                            Macro-read 1 File
```

If Snowhite wants to send a message to the DWARFS list of names and to the ANIMALS list of names, she can create another entry in the names file with the nickname FRIENDS. Under the List of Names field she would type DWARFS and ANIMALS like this:

```
 ====> NAMES (Mail panel)      File: SNOWHITE NAMES A0                 <====
Fill in the fields and press a PFkey to display and/or change your names file
Nickname: FRIENDS             Notebook:
Userid:
  Node:
                    Name:
                   Phone:
                 Address:
                        :
                        :
         List of Names: DWARFS ANIMALS
                        :
                        :
                        :
                        :
 Tag:              Value:
 Tag:              Value:

 1= Help      2= Add       3= Quit     4= Clear      5= Find       6= Change
 7= PrevNick  8= NextNick  9= Delete  10= PrevScrn  11= NextScrn  12= Cursor
                               ======> Screen 1 of 1 <======
 ====> _
                                            Macro-read 1 File
```

If a message is sent to FRIENDS, the following receive it:

To Nickname	Chained List of Names	Actual Recipients
FRIENDS	DWARFS	SNOOZY
		DUMMY
		BOSS
		SMILEY
		GROUCHY
		SNIFFLES
		WISTFUL
	ANIMALS	BEAR

To Nickname	Chained List of Names	Actual Recipients
		LION
		MONKEY

Keeping Your Names File up to Date

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

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

How to Change an Entry

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

```
NAMES nickname
```

For example,

```
names snoozy
```

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

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

How to Delete an Entry

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

```
NAMES nickname
```

Press PF9. The entry is deleted from the names file.

How to Find Information

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

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

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

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

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

2. Fill in one or more fields you can remember.

3. **Press PF5.** Your names file is searched for all entries that exactly match all fields containing information on the NAMES panel.

   If one or more matching entries are found, the panel enters "find mode":

   - All the entries that matched the search criteria form a "find ring." The first entry in the names file that matched the search fields (entry 1 in the find ring) is displayed.

   - A find mode status message (Find *n* of *m*) is displayed at the bottom of the panel where:

- *n* is the number of the matched entry that is currently displayed in the panel
- *m* is the number of matched entries in the find ring.

For example, if 7 entries are found that match the search criteria, the NAMES screen in find mode would appear like:

```
         List of Names:
                       :
                       :
                       :
                       :
                       :
 Tag:             Value:
 Tag:             Value:

 1= Help      2= Add       3= Quit     4= Clear      5= FindQuit   6= Change
 7= PrevNick  8= NextNick  9= Delete  10= PrevScrn  11= NextScrn  12= Cursor
 ====> Find 1 of 7 <====              ======> Screen 1 of 1 <======
 ====>
                                                    Macro-read 1 File
```

If your names file does not contain any entries that match the fields you typed, you will get the message:

```
 Entry not found
```

In find mode, there are also slight changes to the functions of some of the PF keys. See "Summary of PF Keys on the Names Panel in Find Mode" on page 105 for more information.

There is another way for you to quickly locate information in your names file. If you know any of the fields in a names file, use this field with the NAMEFIND command.

For example, if you know you have an entry in your names file with the nickname of SNOOZY, but you do not remember anything else about the entry, type:

```
 namefind :nick snoozy
```

NAMEFIND displays:

```
 :nick SNOOZY
 :userid SNOOZY
 :node COTTAGE
 :name I. M. Dozing
 :phone 777-7777
 :addr Dwarf Cottage; Forest
 Ready;
```

You can ask for any of the above field names (like :name) to get information. For example, if you only remember the name you could quickly locate information about Snoozy by typing:

```
 namefind :name I. M. Dozing
```

The NAMEFIND command can be entered on any command line, without having to go into the names file. But, if you need to make any changes to the file, then you must get into the names file and use the PF6 key to change the entry.

See *z/VM: CMS Commands and Utilities Reference* for more information on the NAMEFIND command.

Summary of PF Keys on the Names Screen

PF1 Help
Display a brief description and example of the NAMES command. (For more detailed information, press PF1 again.)

PF2 Add
Add this entry to the names file.

PF3 Quit
Immediately leaves the NAMES panel processing.

PF4 Clear
Blank out the fields, so you can create a new entry or find an entry.

PF5 Find
Fill in fields on a cleared screen, and press PF5. Locates all entries that match fields in the panel. If more than one entry matches the search criteria, PF5 will be redefined to "FindQuit".

PF6 Change
Change an existing entry. For example, if someone's address or phone number changes, type the new data on their screen and press this key.

PF7 PrevNick
Display the previous entry.

PF8 NextNick
Display the next entry.

PF9 Delete
Remove this entry.

PF10 PrevScrn
Scroll to previous panel for this entry.

PF11 NextScrn
Scroll to next panel for this entry.

PF12 Cursor
Move the cursor to the command line or back to its previous location on the screen.

Summary of PF Keys on the Names Panel in Find Mode

PF1 Help
Display a brief description and example of the NAMES command. (For more detailed information, press PF1 again.)

PF2 Add
Adds an entry to the names file immediately following the last displayed entry. The entry is also added to the find ring regardless of whether the entry matches the find search fields.

Note: The find mode status variable, m, is updated accordingly. If an error occurs that prevents an entry from being added, the find ring is unchanged.

PF3 Quit
Immediately leaves the NAMES panel processing.

PF4 Clear
Exits find mode and clears all panel fields.

PF5 Find or FindQuit
If more than one entry matches the search criteria, PF5 is set to "FindQuit". "Findquit" exits find mode, returns to the NAMES panel, and redisplays the find search fields.

PF6 Change
Changes the entry in the names file that corresponds to the current panel entry. The changed entry remains in the find ring regardless of whether it still matches the find search fields.

PF7 PrevNick
Scrolls to the previous entry in the find ring.

PF8 Next Nick
Scrolls to the next entry in the find ring.

PF9 Delete
Deletes the entry from the names file and the find ring. The entry will continue to be displayed until you scroll to the previous or next entry in the find ring.

> **Note:** The find mode status variable, *m*, is updated accordingly. If an error occurs that prevents an entry from being deleted, the find ring is unchanged. If the last entry in the find ring is deleted, the panel exits find mode and continues to display the last deleted entry in the NAMES panel.

PF10 PrevScrn
Scrolls to the previous panel of the current entry.

Pf11 NextScrn
Scrolls to the next panel of the current entry.

PF12 Cursor
Move the cursor to the command line or back to its previous location on the screen.

Creating Your Own Names File

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

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

After you have a names file, you can use the commands discussed in the rest of this chapter to send messages, notes, and files to these people just by using their nicknames.

You can also use nicknames in place of a user ID to share your files with other users. The Shared File System (SFS) commands GRANT AUTHORITY, REVOKE AUTHORITY, and CREATE ALIAS will accept a nickname.

Sending and Receiving Messages

What Is a Message?

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

TELL Command

The TELL command looks like this:

```
TELL name message
```

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

```
tell snoozy Wake up! It's lunch time!
```

After you type your message, press Enter.

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

```
 11:56:03  * MSG FROM SNOWHITE:  Wake up! It's lunch time!
```

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

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

```
SNOOZY not logged on
```

The message would not be saved and given to Snoozy when he does log on. You have to send it again when he does.

If you need to send a message to someone at a different node than yours, and you do not have an entry for this person in your names file, you must use the form:

```
TELL userid AT node message
```

For example:

```
tell smiley at cottage How are you doing?
```

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

I Do Not Want to Receive Messages

If you do not want to be interrupted by messages while you are working at your terminal, enter this command:

```
SET MSG OFF
```

This command can be entered on the CMS command line, the FILELIST large arrow command line, or the XEDIT large arrow command line. Now, you will not receive messages from other users.

If you decide later that you do want to receive messages, enter this command:

```
SET MSG ON
```

The SET MSG OFF command is good for one terminal session only. When you log off, it is canceled automatically.

Note: If you try to send a message to someone who has used a SET MSG OFF command, you will receive a message telling you that person is not receiving messages.

For Practice

If you would like to practice sending messages, send some to yourself. Type:

```
TELL name anything-you-want
```

For the *name*, substitute your user ID. If you have a names file with an entry for yourself, use your nickname.

Sending Notes

What Is a Note?

A note is the kind of communication you usually do by letter. It is generally longer than a message. A more important difference between messages and notes is that a person does ***not have to be logged on*** to receive a note, as is true with messages. A note is stored by the system until the person is ready to read it.

The NOTE Command

You use the NOTE command to prepare and send a note. A note can be sent to one person or to a list of people. You can even specify that certain people receive a **complimentary copy** of a note. All this happens with no copies printed on paper unless you want them to be.

The format of the NOTE command is:

```
NOTE name
```

where *name* can be one or more nicknames in the names file, or one or more user IDs of people who share your computer, or one or more *userids* at *node* of people who use computers connected to yours.

For example:

```
note snoozy
```

or,

```
note snoozy smiley boss
```

If the person is at a different node, use the form *userid at node* like this:

```
note queen at castle
```

You **must specify a name** when you begin a new note.

When you enter the NOTE command, a special screen is displayed. The following note screen was created by entering the command:

```
note snoozy
```

```
 SNOWHITE NOTE     A0  V 132  Trunc=132 Size=9 Line=9 Col=1 Alt=0

* * * Top of File * * *
OPTIONS: NOACK    LOG    SHORT    NOTEBOOK ALL



Date: 11 February 1992, 11:04:52 EST
From: Snow White                 555-5555             SNOWHITE at FOREST
To:   SNOOZY at COTTAGE

* * * End of File * * *




1= Help      2= Add line  3= Quit     4= Tab      5= Send     6= ?
7= Backward  8= Forward   9= =        10= Rgtleft 11= Spltjoin 12= Power input

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

The note screen has headings that show the date and time, and information about the sender (you) and the recipient(s). You do not have to type these headings. The information is collected from your names file and is placed in the note automatically. (You can ignore the line that begins with OPTIONS.)

If you want the headings to contain more information than is shown in the previous headings, you can enter this form of the NOTE command:

```
NOTE nickname (LONG
```

You must type the left parenthesis before the word LONG, exactly as shown here.

These headings were produced by entering the following command:

```
NOTE snoozy (LONG
```

```
Date: 11 May 1990, 11:04:52 EST
From: Snow White                 555-5555              SNOWHITE at FOREST
      201 Redwood Lane
      Forest Primeval, California
To:   I. M. Dozing               777-7777              SNOOZY   at COTTAGE
```

The names and addresses are collected from your names file. If you use the LONG option and someone's name or phone number is not in the names file entry, it will not appear in the note heading.

Typing the Body of the Note

After entering NOTE, you are in edit mode, as you are when you XEDIT a file. However, the screen looks different from the usual XEDIT screen layout. There is no prefix area or scale.

Before you start typing the note, enter the INPUT command on the command line (next to the large arrow at the bottom of the screen). Then you can type the body of the note, in the space provided after the headings.

```
 SNOWHITE NOTE      A0  V 132  Trunc=132 Size=9 Line=9 Col=1 Alt=0

* * * Top of File * * *
OPTIONS: NOACK    LOG    SHORT    NOTEBOOK ALL



Date: 11 February 1992, 11:04:52 EST
From: Snow White                 555-5555              SNOWHITE at FOREST
To:   SNOOZY at COTTAGE

Dear Sir:
   Would you be interested in hiring a domestic supervisor?
I understand the cottage needs help, what with your having
to work in the mines and sing "Heigh-Ho" and all that.
I am a hard worker and am seeking employment.
                                   Thank you,
                                   S. White
1= Help      2= Add line  3= Quit    4= Tab      5= Send      6= ?
7= Backward  8= Forward   9= =       10= Rgtleft  11= Spltjoin 12= Power input
Input mode:
====> * * * Input Zone * * *
                                                  Input-mode  1 File
```

All the PF keys but PF5 are set to commands that help you edit your note. If they look familiar, it is because they are XEDIT commands. For example, if your note does not fit on one screen, you press PF8 (FORWARD) to continue typing on a new screen.

If you want to use XEDIT commands that are not assigned to PF keys, you must go back to edit mode. Press Enter twice, just as you do in a regular file. Then you can use both the PF keys and enter XEDIT commands from the command line while composing your note. If you do not need to use any XEDIT commands, you can stay in input mode until you finish typing.

If you want to use insert mode, just press the PA2 key to unlock the line, press the insert mode key, insert the characters or words, and press RESET to resume normal typing.

Sending the Note

When you are finished typing your note, **_press PF5_** to send it.

Sending Notes to People on a List

The nickname you specify in the NOTE command can be the nickname assigned to a list in your names file. A copy of the note is sent to each person in the list.

For example, Snow White's names file contains an entry for a list whose nickname is DWARFS. The List of Names field has the nicknames of the seven dwarfs. (You can find this entry under "Creating an Entry for a List of Names" on page 100.) The command

```
note dwarfs
```

sends a copy of the note to each of the seven dwarfs.

The note screen would look like this:

```
 SNOWHITE NOTE     A0  V 132  Trunc=132 Size=9 Line=9 Col=1 Alt=0

* * * Top of File * * *
OPTIONS: NOACK    LOG    SHORT    NOTEBOOK ALL



Date: 11 February 1992, 11:04:52 EST
From: Snow White              555-5555           SNOWHITE at FOREST
To:   SNOOZY at COTTAGE, DUMMY at COTTAGE, BOSS at COTTAGE, SMILEY at COTTAGE
      GROUCHY at COTTAGE, SNIFFLES at COTTAGE, WISTFUL at COTTAGE

Dear Sir,
   Would you be interested in hiring a domestic supervisor?
I understand the cottage needs help, what with your having
to work in the mines and sing "Heigh-Ho" and all that.
I am a hard worker and am seeking employment.
                            Thank you,
1= Help      2= Add line  3= Quit     4= Tab       5= Send      6= ?
7= Backward  8= Forward   9= =         10= Rgtleft  11= Spltjoin 12= Power input

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

If you want the long headings, you would enter:

```
note dwarfs (long
```

Then the headings would look like this:

```
From: Snow White              555-5555           SNOWHITE at FOREST
      201 Redwood Lane
      Forest Primeval, California
To:   I. M. Dozing           777-7777           SNOOZY   at COTTAGE
      S. A. What             777-7777           DUMMY    at COTTAGE
      T.O.P. Banana          777-7777           BOSS     at COTTAGE
      H. A. Haas             777-7777           SMILEY   at COTTAGE
      E. B. Scrooge          777-7777           GROUCHY  at COTTAGE
      A. H. Choo             777-7777           SNIFFLES at COTTAGE
      R. U. Shy              777-7777           WISTFUL  at COTTAGE
```

Sending Complimentary Copies

Sometimes you may want someone to see a copy of a note, but you want that person to know that he or she is not the main recipient. If you were using paper, you would send the person a carbon copy. With the computer, you can send a *complimentary copy*. To send a complimentary copy, you could enter the NOTE command like this:

```
note boss cc: sniffles smiley
```

cc stands for complimentary copy. The name following *cc:* can be one or more nicknames of people who have entries in the names file, user IDs of people who share your computer, or user IDs and nodes of people who use computers connected to yours (in the form *userid* at *node.*)

In this example, BOSS is the main recipient. SNIFFLES and SMILEY are nicknames of people who have entries in the names file. The note header will show that they are complimentary copy recipients.

```
Date: 11 May 1990, 11:04:52 EST
From: Snow White              555-5555           SNOWHITE at FOREST
```

```
To:   BOSS at COTTAGE
cc:   SNIFFLES at COTTAGE, SMILEY at COTTAGE
```

The name following **cc:** can also be the nickname of a list.

```
note witch cc: dwarfs
```

DWARFS is the nickname of an entry that contains the names of the people in the list. (You can find this entry under "Creating an Entry for a List of Names" on page 100.) Each person in the list will be a complimentary copy recipient.

Keeping Copies of Notes

A copy of every note you send is saved automatically in a file called ALL NOTEBOOK. Each time a note is sent, a copy is added to the end of this file. A line of equal signs separates each note. If you ever need to look up a note, you can use the editor to examine this file. You would enter XEDIT ALL NOTEBOOK.

Sometimes you may want to keep all the notes you send to one person in a separate file. To do this, you can specify a file name in the Notebook field of that person's names file entry. A copy of each note you send to that person is saved automatically in a file whose file name is the name you specified and whose file type is NOTEBOOK. Also, all notes that you **receive** from that person are also kept in this file. You will see how to receive notes in "Receiving Notes and Files" on page 117.

If you ever need to look up a note that you sent, you can use the editor to examine that file. For example, suppose you typed snoozy in the Notebook field of Snoozy's names file entry. You could enter XEDIT SNOOZY NOTEBOOK to look at the notes you sent to Snoozy.

Continuing Notes

Suppose you are typing a note and have to stop before you are finished. You can save the note and finish it later by following these steps:

- Enter FILE from the XEDIT command line at the bottom of the note screen. Your note is saved under the file identifier *userid* NOTE, where *userid* is your user ID.
- Later, when you want to finish the note, enter the NOTE command, but **do not specify a name.** Your unfinished note appears on the screen. You can then complete the note and send it.

Not Continuing a Note

If you decide not to continue with a note that you previously saved, you must get rid of it before you can send another note. You must replace the old note when you are sending a new note. For example, suppose you ended a note without sending it to Snow White. Next, you decided to send a note to Grouchy instead. You would enter:

```
note grouchy (replace
```

This completely replaces the note you typed to Snow White and starts a new one addressed to Grouchy.

Summary of PF Keys on the NOTE Screen

PF1 Help
Display a brief description and example of the NOTE command. (For more detailed information, press PF1 again.)

PF2 Add line
Add a blank line after the line containing the cursor.

PF3 Quit
End the note session without saving any changes.

PF4 Tab
Use this key to tab the cursor.

PF5 Send
> Send the note.

PF6 ?
> Display the last command that was processed.

PF7 Backward
> Scroll back one screen.

PF8 Forward
> Scroll forward one screen.

PF9 =
> Execute again the last command that was processed.

PF10 Rgtleft
> View data to the right of the screen. Press PF10 again to return to original view.

PF11 Spltjoin
> Split a line or join two lines at the cursor.

PF12 Power input
> Enter power typing mode.

For more information on the various XEDIT commands, see Chapter 2, "Editing Files," on page 7.

Summary of How to Send a Note

Follow these steps to prepare and send a note.

- Enter the command NOTE in one of its forms:

```
NOTE name

NOTE name CC: name

NOTE name (LONG

NOTE name CC: name (LONG
```

In each form, *name* can be one or more names. A name can be a nickname of an entry in the names file, or the user ID of someone who uses the same computer system that you use, or the user ID and node of someone who uses a computer connected to yours, or it can be a combination of nicknames and user IDs.

- Type the body of the note.
- Press PF5 to send the note.

Sending Files

Suppose you have a file that you want other people to see. If you use Shared File System (SFS) commands, you can grant the other users authority to the file so that they can share it. (See Chapter 3, "Managing Your Files," on page 53 for details.)

If your files are stored on a minidisk (and you cannot use SFS commands), or if, for some reason, you do not want to grant other people authority to your file, send them a copy of the file. This section shows you how to send people a copy of a file for their own personal use. Each recipient can then place the copy in permanent storage so it will be available when it is needed.

SENDFILE Command

When you enter the SENDFILE command, a special screen for data entry is displayed, like the one that follows. You type the name of the file to be sent and the name of the recipient directly on this screen.

An abbreviation for SENDFILE is SF.

```
                --------------- SENDFILE ---------------
File(s) to be sent    (use * for Filename, Filetype and Filemode
                            to select from a list of files)
Enter filename : _
      filetype :
      filemode :

Send files to  :

Type over 1 for YES or 0 for NO to change the options:

    0       Request acknowledgement when the file has been received?

    1       Make a log entry when the file has been sent?

    1       Display the file name when the file has been sent?

    0       This file is actually a list of files to be sent?

   1= Help           3= Quit           5= Send             12= Cursor

====>
                                              Macro-read 1 File
```

Sending One File

To send one file, follow these steps:

- Type the file name in the space provided.
- Type the file type in the space provided.
- Type the file mode if it is not **A**. You can leave this field blank if the file is on file mode A (the directory or minidisk accessed with a file mode of A). If the file is on a different file mode, you ***must*** specify that file mode letter, or the file will not be found.
- Type the name(s) of the recipient(s). You can use nicknames from your names file. The nickname can represent a single user.

  For example:

  ```
  Send files to: snoozy
  ```

  Or you can type more than one name on this line. (Do not separate the names with commas.) For example:

  ```
  Send files to: snoozy grouchy prince
  ```

  The nickname can also represent a list. The file is sent to each person in the list. For example:

  ```
  Send files to: dwarfs
  ```

  If the recipient uses the same computer system that you use, you can type the user ID. If the recipient uses a computer system connected to yours, you can type the user ID and node.
- Choose the ***0/1*** options.

  You can select certain options from the list on the screen. They are described in the following section.
- Press PF5 to send the file and exit from the sendfile screen. Pressing Enter sends the file but keeps the screen (so that you can use it to send another file).

Choosing the Options

Each option is in the form of a question, with either 0 or 1 displayed to its left. A 0 represents NO, while a 1 represents YES. If you are satisfied with the answer shown, do not do anything. If you want a different answer, type 1 over the 0 (or 0 over the 1).

The options are as follows:

```
0    Request acknowledgement when the file has been received?
```

Type 1 only if you want to get an acknowledgment when the person receives (or discards) your file. The acknowledgment shows the date and time you sent the file, and the recipient's user ID and node.

When you get an acknowledgement, it appears in your reader and a message appears on your screen. The message informs you that you have been sent a file. (You will learn how to handle these files in "Receiving Notes and Files" on page 117.) If you choose to receive the acknowledgment, an entry is made in a ***userid NETLOG*** file.

```
1    Make a log entry when the file has been sent?
```

Each time you send a file, an entry is automatically made in a special log file. Its file identifier is ***userid NETLOG***, where the user ID is your user ID.

A typical entry might look like this:

```
File MY DATA A1 sent to SNOOZY at COTTAGE on 10/10/88 11:30:25
```

If you specified 1 (YES) on the first option (acknowledgment), an entry is also made when you receive the acknowledgment.

Type 0 if you do not want an entry made in the log file.

```
1     Display the file name when the file has been sent?
```

The names of the file(s) and the user ID(s) and node(s) of the recipients are displayed on a cleared screen. Type 0 for NO if you do not want this information displayed.

```
0     This file is actually a list of files to be sent?
```

This option provides one way to send many files at once. In order to use it, you need to learn how to use the FILELIST command, which is explained in Chapter 3, "Managing Your Files," on page 53. Type 1 for YES if you have saved a list of files produced by the FILELIST command. This option is discussed more fully in the following section.

Sending More than One File

What if you want to send a lot of files to someone? One way to do this is by filling in sendfile screens, one for each file you want to send. But there are easier ways. One way is to save a list of the files you want to send using the FILELIST command. Another way is to select the files you want to send from a list that contains a subset of your files. Both ways allow you to send all the files at once.

Using FILELIST to Save a List of Files to be Sent

You can use the FILELIST command to create a list of the files you want to send. Then you FILE that list. (If you are not familiar with the FILELIST command, read Chapter 3, "Managing Your Files," on page 53.)

For example, if you want to send all files on file mode A with a file type of ***payroll,*** you would enter:

```
filelist * payroll
```

The list might look like this:

```
 SNOWHITE FILELIST A0  V 108  Trunc=108 Size=7 Line=1 Col=1 Alt=0
Directory = VMSYSU:yourid.DWARFS
Cmd   Filename Filetype Fm Format Lrecl Records Blocks   Date     Time
      WISTFUL  PAYROLL  A1 V         95     187      7  5/04/90 21:12:04
      BOSS     PAYROLL  A1 V         95      99      4  5/04/90 20:58:07
      DUMMY    PAYROLL  A1 V        107      75      3  5/04/90 17:59:00
      WITCH    PAYROLL  A2 V         98      96      4  5/03/90 15:17:01
      GROUCHY  PAYROLL  A1 V         92      60      2  5/02/90 15:33:05
      PRINCE   PAYROLL  A2 V         75     112      5  4/25/90 12:10:03
      SNOOZY   PAYROLL  A2 V        120      58      2  4/24/90  9:14:02




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

====> file my list
                                                 X E D I T  1 File
```

Remember that the list is a file whose file identifier is *yourid FILELIST*. If you want to send only these files, you must give the file a new file identifier. Enter a FILE command on the large arrow command line, changing the file identifier. Change either the file name or file type (or both) in your FILE command. In the previous screen, the FILE command changes the identifier to MY LIST when you press Enter. The list would now be on file mode A. Its file identifier would be MY LIST A. (If you do not specify a file type, it will be FILELIST.)

Enter the SENDFILE command.

```
                ---------------  SENDFILE  ---------------
File(s) to be sent    (use * for Filename, Filetype and Filemode
                         to select from a list of files)
Enter filename : my
      filetype : list
      filemode :

Send files to  :

Type over 1 for YES or 0 for NO to change the options:

   0      Request acknowledgement when the file has been received?

   1      Make a log entry when the file has been sent?

   1      Display the file name when the file has been sent?

   1_     This file is actually a list of files to be sent?

   1= Help          3= Quit          5= Send          12= Cursor

====>
                                           Macro-read 1 File
```

On the sendfile screen, you

- Type the file name and file type of the FILELIST display you saved.
- Type the name of the recipient(s).
- Type *1* over the *0* for the list option.
- Press PF5 to send the files.

All the files in MY LIST, and the MY LIST file itself, are sent to the user you specified.

Selecting Files from a List

Another way to send many files at once is to select the files from a list of some or all of your files. You would probably choose this method if you wanted to send only some, but not all, files with the same file name or file type. Or perhaps you would like to browse through a list before deciding which files to send.

First enter the SENDFILE command to display a sendfile screen.

```
                ----------------  SENDFILE  ---------------

File(s) to be sent     (use * for Filename, Filetype and Filemode
                            to select from a list of files)
Enter filename : *
      filetype : data
      filemode :

Send files to  : sleepy_

Type over 1 for YES or 0 for NO to change the options:

    0       Request acknowledgement when the file has been received?

    1       Make a log entry when the file has been sent?

    1       Display the file name when the file has been sent?

    0       This file is actually a list of files to be sent?

    1= Help           3= Quit              5= Send              12= Cursor

====>
                                                    Macro-read 1 File
```

On the sendfile screen, you type an asterisk (*) next to the file name, file type, or file mode fields, as shown here. An asterisk in any of these fields means that you want the list to contain **all** files with a certain file name, file type, or file mode. Of course, if you wanted the list to contain **all** files on file mode A, you would type an asterisk for file name **and** file type.

In the previous screen, an asterisk is typed in the file name field and **data** is typed in the file type field. This means that the list will contain all files on Snow White's file mode A with a file type of DATA. To display the list, first finish filling out the sendfile menu, and then press PF5.

```
 SNOWHITE FILELIST A0  V 108  Trunc=108 Size=9 Line=1 Col=1 Alt=0
Directory = VMSYSU:yourid.DWARFS
Cmd    Filename Filetype Fm Format Lrecl Records Blocks   Date     Time
 s     WISTFUL  DATA     A1 V          95     187      7  5/04/90 21:12:04
       BOSS     DATA     A1 V          95      99      4  5/04/90 20:58:07
 s     DUMMY    DATA     A1 V         107      75      3  5/04/90 17:59:00
       MISC     DATA     A2 V          98      96      4  5/03/90 15:17:01
 s     GROUCHY  DATA     A1 V          92      60      2  5/02/90 15:33:05
       PRINCE   DATA     A2 V          75     112      5  4/25/90 12:10:03
       SNOOZY   DATA     A2 V         120      58      2  4/24/90  9:14:02
       SNIFFLES DATA     A1 V          26       7      1  4/23/90  8:50:06
       WITCH    DATA     A1 V          80       3      1  3/26/90  8:05:08




1= Help      2= Refresh   3= Quit    4= Cancel     5= Sendfile    6= Sort(size)
7= Backward  8= Forward   9= FL /n   10=           11= XEDIT/LIST 12= Cursor
Type 'S' in front of each file to be sent and press Enter
====>
                                                    X E D I T  1 File
```

To mark the files you want sent, type the letter **s** before the file name of each file you want sent. Then press Enter to send the files. In this example, three files would be sent to Sleepy (the recipient specified on the sendfile screen). They are the files marked with an **s**. When the files are sent, an asterisk (*) appears in the Cmd space next to each file sent.

If your FILELIST display extends for more than one screen and you also want to send some of these files, first scroll the screen (using PF8). Then type an **s** next to the files you want to send and press Enter.

When you are finished sending files, press PF3 to exit from FILELIST (and sendfile) screens.

PF Keys on the FILELIST Screen

The PF keys are set to commands that help you find information or rearrange the files in the list. Except for PF5, they are the same as those on the regular FILELIST screen. (You can use PF5 to send a file by placing the cursor on that line and then pressing PF5.) For more information on these PF keys, see the FILELIST command, in Chapter 3, "Managing Your Files," on page 53.

Summary

Here are the steps to follow when sending many files by selecting them from a list:

- Type SF and press Enter to display the sendfile screen.
- Type * for file name, file type, or file mode to define the subset of all your files you want displayed in the list.
- Type the name of the recipient(s).
- Type over the 0|1 options, if you want something other than what is shown on the screen.
- Press PF5 to display the list.
- Type **S** before each file name you want sent.
- Press Enter to send the file(s).
- Press PF3 to quit.

For Practice

For practice, you can send yourself a file. After entering SF, fill in the sendfile screen. Type in the file name and file type of a file that you already have on file mode A (the directory or minidisk accessed with a file mode of A). Type in your nickname (if you have an entry for yourself in a names file), or your user ID. Then press PF5.

The next section shows you how to receive files and notes that are sent to you. If you send yourself a file, you will have a real example to follow.

Receiving Notes and Files

All notes and files that other people send you are delivered to your **reader**. Your reader is a temporary place where incoming notes and files are kept until you decide what you want to do with them. In this section, both notes and files are called files.

After you log on, a message may appear on your screen, notifying you that a file is in your reader. It might be similar to the following message:

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

The part of the message that reads FILES: 003 RDR means that one or more people sent you a total of three files, and they are currently in your reader.

You might also receive a message that files have arrived in your reader while you are editing a file. The message appears on a cleared screen. When you press CLEAR, your file returns.

You can use the RDRLIST command to display information about the files in your reader. Then you can decide if you want to keep the files or discard them.

RDRLIST Command

The RDRLIST command displays information about files in your reader. An abbreviation for RDRLIST is RL.

If you sent yourself a file, as suggested in the previous section, you can enter RL now.

A sample rdrlist screen follows.

```
 SNOWHITE RDRLIST  A0  V 164  Trunc=164 Size=3 Line=1 Col=1 Alt=2
Cmd    Filename Filetype Class User  at Node     Hold  Records  Date     Time
_      MINING   NEWS     PUN A SNOOZY   COTTAGE   NONE       3 07/07   10:41:38
       SNOOZY   NOTE     PUN A SNOOZY   COTTAGE   NONE       5 07/07   10:3:11
       MIRROR   MIRROR   PUN A QUEEN    CASTLE    NONE       1 07/07   10:2:50




















1= Help      2= Refresh   3= Quit    4= Sort(type) 5= Sort(date) 6= Sort(user)
7= Backward  8= Forward   9= Receive 10=           11= Peek      12= Cursor

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

As you can see, the rdrlist screen lists information about the files in Snow White's reader:

Cmd is an area where you can enter commands. We will discuss this later.
Filename is the file name.
Filetype is its file type.
Class is not important to you right now.
User tells you the user ID of the person who sent you the file.
At Node is the person's node.
Hold is not important to you right now.
Records tells you how many records are in the file.
Date is the date the file was sent.
Time is the time it was sent.

What Should I Do with the Files in My Reader?

You can do any of the following things with each of the files in your reader:

• *Receive* the file, that is, move it from the reader to permanent storage.

• *Discard* the file.

• *Peek* at the file, that is, look at its contents without moving it to permanent storage. Then decide if you want to receive it or discard it.

Receiving a File

To receive a file, move the *cursor* to the line that describes the file you want to receive. Then *press PF9* or enter the RECEIVE command. A notice will appear on that line, telling you that the file has been received, as follows:

```
 SNOWHITE RDRLIST  A0  V 164  Trunc=164 Size=3 Line=1 Col=1 Alt=2
Cmd    Filename Filetype Class User  at Node     Hold  Records  Date     Time
*      MINING   NEWS     received from SNOOZY at COTTAGE.
       SNOOZY   NOTE     PUN A SNOOZY   COTTAGE   NONE       5 07/07   10:3:11
       MIRROR   MIRROR   PUN A QUEEN    CASTLE    NONE       1 07/07   10:2:50
```

Notice that an asterisk appears in the Cmd space next to the received file. It is just a signal to remind you that you did something with this file.

When you receive files with PF9, they are by default placed on file mode A (your top directory or minidisk accessed with a file mode of A).

You can also choose to receive files by using the RECEIVE command. To do so, move the cursor to the line that describes the file you want to receive. Then enter:

```
receive
```

in the Cmd area and press Enter to place the file on file mode A.

If more than one file is in your reader, move the cursor to the appropriate line and press PF9 or enter RECEIVE for each file you want to receive.

A quick way to receive several files from your reader is to press PF9 (or enter the RECEIVE command) for the first file you want to receive. Then move your cursor to the Cmd area for each of the remaining files you want to receive and place an equal sign (=) in front of each one. Press Enter and all of these files are placed on your file mode A.

If you wish to receive a file onto a file mode other than A (another SFS directory or minidisk), you can do so by using the RECEIVE command. For example, to receive the file MIRROR MIRROR onto an SFS directory accessed with a file mode of B, type the following command on the line for the MIRROR MIRROR file. Be sure to leave a space before and after the slash:

```
receive / = = b
```

Your screen will look like this:

```
 SNOWHITE RDRLIST  A0  V 164  Trunc=164 Size=3 Line=1 Col=1 Alt=2
Cmd   Filename Filetype Class User  at Node    Hold  Records  Date     Time
*     MINING   NEWS     received from SNOOZY at COTTAGE.
      SNOOZY   NOTE     PUN A SNOOZY   COTTAGE  NONE        5 07/07   10:3:11
receive / = = bMIRROR   PUN A QUEEN    CASTLE   NONE        1 07/07   10:2:50
```

When you press Enter, this command will receive the file in your reader. The file name and file type will remain the same. The file mode will be changed to B. That is, the file will be received onto the directory or minidisk accessed with a file mode of B.

Replacing or Renaming a File

What happens if the file in your reader has the same file identifier as one you already have? If the files are stored in an SFS file pool and the files will be kept in different directories, you can receive the file.

However, if the files will be in the same SFS directory, you must replace or rename one of them. The same is true if your files are stored on a minidisk. Two files in the same directory or on the same minidisk cannot have the same file identifier.

For example, if you already have a file called MINING NEWS in your top directory or on your minidisk, you would not be able to receive another file called MINING NEWS into the same directory or onto the same minidisk.

When you press PF9, the following message appears at the bottom of your rdrlist screen:

```
File fn ft fm already exists; specify REPLACE option
```

This means that the computer will not move the file to permanent storage.

There are two ways to handle this.

If you want the file in your reader to replace the file you already have, type the following command in the Cmd space next to the file you want to receive. Then press Enter.

```
receive / (replace
```

The slash (/) is a kind of shorthand. It stands for the number that z/VM assigned the file in your reader.

You would probably want to replace the file you have if, for example, someone sent you an updated copy of it.

However, if you want to keep the file currently on file mode A (your directory or minidisk accessed with a file mode of A), you can give the file in your reader a different file identifier. Type the following command in the Cmd space. Then press Enter.

```
receive / newfilename newfiletype
```

You need to specify either a new file name **or** a new file type for the file identifier to be unique. For example, if a file identifier is MINING NEWS, you might specify either RECEIVE / MINING1 NEWS or RECEIVE / MINING TIDBITS.

Note: If you have any authorities granted on the file you are replacing, these authorities are not lost when you use RECEIVE with the REPLACE option. However, if you were to erase the original file, and RECEIVE the new version, all authorities granted on the file would be lost, even though the file name is the same.

Receiving Notes

You can recognize a note because its file type is NOTE. The second file in Snow White's reader is a note. You receive a note file the same way you do other kinds of files. Move the **cursor** to the line that describes the note and **press PF9** or enter the RECEIVE command.

```
 SNOWHITE RDRLIST  A0  V 164  Trunc=164 Size=3 Line=1 Col=1 Alt=2
Cmd    Filename Filetype Class User  at Node    Hold  Records  Date    Time
*      MINING   NEWS     received from SNOOZY at COTTAGE.
*      SNOOZY   NOTE     added to ALL NOTEBOOK A0.
       MIRROR   MIRROR   PUN A QUEEN    CASTLE   NONE      1 07/07  10:2:50
```

This note file was added to the file ALL NOTEBOOK, as the notice shows. Of course, if you specified a different NOTEBOOK file name in your names file entry for that person, the note is added to that file.

Discarding a File

To discard a file, type DISCARD in the Cmd space next to the file name. Then press Enter. The notice **has been discarded** appears in that line.

Sometimes you cannot decide whether you want to receive or discard a file unless you **peek** at its contents. Let us peek at the last file in Snow White's reader and then decide if we want to keep it or discard it.

Peeking at a File

By placing the **cursor** on the appropriate line and **pressing PF11**, you can **peek** at the file. This means that you can see what is in the file without first moving it to your top directory or minidisk. Let us peek at the last file in Snow White's reader.

It is displayed on a peek screen, as shown:

```
 3001     PEEK      A0  V 255  Trunc=255 Size=7 Line=0 Col=1 Alt=0
FILE MIRROR MIRROR from QUEEN  at CASTLE.  Format is DISK-DUMP.
* * * Top of File * * *
             Apples, Apples, Apples

  I have recently developed a new variety of apple.  It is
crunchier, sweeter, and larger than the run-of-the-mill varieties.
I am leaving samples on doorsteps throughout the forest, free
of charge.  I guarantee you will like these apples.  After just one
bite, you will feel tranquil, totally at peace.
* * * End of File * * *




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

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

PF Keys on the Peek Screen

You can locate information by typing the XEDIT command:

```
 ===> CLOCATE /data/
```

at the bottom of the screen and pressing PF5. Each time you press PF5, the next occurrence of the string is located. This means that the cursor moves under the string, and the line that contains it is highlighted.

If the file takes up more than one screen, you can use PF7 and PF8 to scroll through the file.

If you decide you want to receive the file, press PF9. You will then go back to the rdrlist screen automatically.

If you decide to discard the file, type DISCARD in the large arrow command line on the peek screen, and press Enter.

If you are not sure whether you want to receive or discard the file, you can choose to leave it in your reader for now. You can press PF3 to quit the PEEK environment. The file will remain in your reader.

Snow White decides that no news from the Bad Queen can be good news, so she discards the file.

Now her rdrlist screen looks like this:

```
 SNOWHITE RDRLIST     A0  V 108  Trunc=108 Size=3 Line=1 Col=1 Alt=9
Cmd    Filename Filetype Class User  at Node Hold  Records  Date  Time
*      MINING   NEWS   received from SNOOZY at COTTAGE.
*      SNOOZY   NOTE   added to ALL NOTEBOOK A0.
*      MIRROR   MIRROR has been discarded.
```

Using PF Keys to Scroll through or Rearrange the List

Snow White's reader contains only three files. However, your reader can contain many files, and the rdrlist display can occupy more than one screen.

By using PF keys, you can scroll through the rdrlist screens and rearrange the list of files. PF7 and PF8 scroll the screen backward and forward, respectively. PF4, PF5, and PF6 sort the list according to the file type, date, and the user id from whom the file was sent.

When you are finished using the rdrlist screen, press PF3 to quit.

Is There a File in Your Reader?

If you sent yourself a file, as suggested in the last section ("Sending Files" on page 112), enter the command RL (if you have not already). If you sent yourself a file that you already have in permanent storage, you can discard the one in your reader. Type DISCARD in the Cmd area on the rdrlist screen, and press Enter. Then press PF3 to quit.

Summary

You learned the following commands in this chapter:

Command	Function
NAMES	Displays names file data entry screen.
TELL	Sends a message to another person who is logged on.
SET MSG OFF	Prevents messages from being displayed.
SET MSG ON	Resumes message display.
NOTE	Prepares a note for another user, who may or may not be logged on.
SENDFile or SF	Displays data entry screen for sending a file.
RDRList or RL	Displays information about files in your reader.
RECEIVE	Receives a file that is in your reader, that is, moves it from your reader to permanent storage.
DISCARD	Discards a file that is in your reader.

For more information on the commands discussed in this section, see *z/VM: CMS Commands and Utilities Reference*.

Chapter 6. Asking Questions

This section teaches you how to ask z/VM:

- Who am I, where am I, and what is the time and date?
- Who is using the computer?
- How much space do I have left for my files?
- Which command should I use?

Where Do I Enter These Commands?

The commands discussed in this chapter can be entered on:

- The CMS command line
- The FILELIST large arrow command line
- The RDRLIST large arrow command line
- The XEDIT large arrow command line.

When you enter a command from the FILELIST, RDRLIST, or XEDIT command line, the answer to your question will be displayed on a cleared screen, with MORE... displayed in the status area. You can press CLEAR to get the FILELIST display, reader list, or file back on the screen.

Who Am I, Where Am I, What Is the Date and Time?

When you are using z/VM, you will never have an identity crisis, and you will not need a calendar or a watch! To find out your user ID and node, and the date, time, and day of the week, use this command:

```
IDentify
```

You will get a message like this one:

```
SNOWHITE AT FOREST  VIA RSCS  06/10/09  12:01:00 EDT  WEDNESDAY
```

Who Is Using the Computer?

You can use three different QUERY commands to find out who else is using your computer.

If you want to know how many people are using your computer, which can affect the response time (how long it takes z/VM to do the various things you ask it to do), enter:

```
Query Users
```

The number of people currently logged on is displayed.

If you want to know which of your fellow users are currently logged on, enter:

```
Query Names
```

You will see a list of the user IDs of all the people who are currently logged on.

If you want to know if a particular person is logged on and you do not want to see the whole list of users, enter the QUERY command with that person's user ID. For example:

```
q johndoe
```

z/VM responds with a message. If JOHNDOE is logged on, z/VM gives the address of the terminal he is using. If he is not logged on, you see this message:

```
JOHNDOE not logged on
```

How Much Space Do I Have Left for My Files?

Whether your files are stored in a Shared File System (SFS) file pool or on a minidisk, you can enter commands to determine how much space you have used and how much space is left for your files. If your files are stored in an SFS file pool, see the section that follows. If your files are stored on a minidisk, see

How Much of My File Space Have I Used?

No user has an infinite amount of file space. How can you find out how much of your file space is full? With the command:

```
Query LIMITS
```

The response is similar to this:

```
Userid     Storage Group    4K Block Limit  4K Blocks Committed   Threshold
yourid              3                 1000              820-82%        90%
```

The first column shows your user ID. The second column, labeled `Storage Group`, shows the storage group within the file pool to which your system administrator has assigned your file space.

The third and fourth columns contain the information regarding the size of your file space. The third column, `4K Block Limit`, shows that you have been allocated 1000 4K block units. 1K equals 1024 characters; 4K equals 4096 characters. That is approximately the equivalent of one typewritten page of information (about two screens full on your terminal). Therefore, 1000 4K blocks would give you enough room to store about 1000 pages of information (2000 screens full).

The fourth column, `4K Blocks Committed`, shows of the 1000 4K blocks you were allocated, you have used 820, which is 82% of the total. Therefore, you have 180 4K blocks left, or enough storage for approximately 180 typewritten pages of information.

The fifth column, `Threshold`, shows when you will receive a warning from the system informing you that your file space is almost full. The default threshold is 90%. That is, when your file space is 90% full, you will receive a warning. You will continue to receive warnings until you erase some files or get your file space increased by your system administrator. You can change the threshold to find out sooner (or wait longer to find out) that your file space is almost full. To change the threshold, use the SET THRESHOLD command.

How Much Minidisk Space Have I Used?

Minidisks are identified by letters, known as file modes. The file mode that you usually use when you create a file is A. That means that your file is put on the minidisk accessed with a file mode of A. How can you find out how much minidisk space is left? With this command:

```
Query DISK A
```

The response is a message similar to this:

```
LABEL    VDEV M STAT CYL TYPE BLKSIZE FILES BLKS USED-(%) BLKS LEFT BLK TOTAL
DAB191   191  A  R/W   5 3390 4096       180    2473-61        1577     4050
```

The minidisk label is DAB191. The address of the minidisk (where the minidisk is) is 191. The file mode letter is A. R/W means that you can read and write on this minidisk. Some minidisks are R/O, which means read only. That is, you cannot write a file on them. The fields labeled CYL, TYPE, and BLKSIZE

give information about the amount of space on this particular minidisk and how that space is organized. FILES tells you how many files you have on the minidisk. BLKS USED shows the percentage of the space you have used. Here, you see that 43% of this minidisk is used. Therefore, 57% of this minidisk is still available. The rest of the fields are not important to you now.

Which Command Should I Use?

Have you ever forgotten the name of a command? If you know what you want to do, but cannot remember how to do it, use the HELP command. You can enter the HELP command from any large arrow command line.

To use the HELP command, type:

```
====> help
```

Then press ENTER.

When you type HELP from the XEDIT, RDRLIST, or FILELIST command line, you will see a **_menu_**. The menu lists all the XEDIT commands. (HELP in XEDIT was described in Chapter 2, "Editing Files," on page 7.)

When you type HELP from the CMS command line, your screen should look like this:

```
HELP TASKS                   Task Help Information        line  1 of  39
(c) Copyright IBM Corporation 1990, 2009

z/VM HELP, main panel

The help panels listed below provide information about various
z/VM functions, topics, and tasks.
To view a help panel, move the cursor to any character of the name
or description and press the ENTER key or the PF1 key.


HELP      - Using the z/VM HELP Facility
MENUS     - z/VM help menus
TASKS     - Basic z/VM tasks
AVS       - AVS commands
CMS       - CMS commands
CP        - CP commands
DIRMAINT  - DirMaint commands
DIRMAINT  - DirMaint topics
PF1= Help    2= Top     3= Quit    4= Return    5= Clocate   6= ?
PF7= Backward 8= Forward  9= PFkeys  10=          11=          12= Cursor

====>
                                              Macro-read 1 File
```

This is a **_menu_** of the different types of help you can get. Note that this is only the first screen of the main HELP menu. If the task or command you want is not listed on the first screen of any menu, you can press PF8 to go to the next screen and PF7 to return to the first screen.

If you move the cursor under TASKS and press ENTER, you will see:

```
TASK TASK                    Task Help Information        line 1 of 22
(c) Copyright IBM Corporation 1990, 2009

Help for basic z/VM tasks

To view a Help panel, move the cursor to any character of the description
and press the ENTER key or the PF1 key.


Create or change (edit) files
Manage files (copy, rename, erase, etc.)
Work with windows in fullscreen CMS
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
```

Now move the cursor so it is under the task you want to do. Press ENTER. The next screen you see will provide more information about the task. Just follow the directions on the screen.

To leave the HELP screens at any time, press PF4. Enter:

```
help help self
```

to find out more about using the HELP Facility.

Summary

You learned the following commands in this chapter:

Command	Function
Help	Displays information about commands.
IDentify	Displays user ID, node, time, and date.
Query Users	Displays how many people are logged on.
Query Names	Displays user IDs of people who are logged on.
Query *userid*	Tells you if a particular user ID is logged on.
Query LIMITS	Displays the size of your file space and shows how much of the file space you have used.
Query DISK A	Displays the size of your minidisk and shows how much of the space on the minidisk has been used.

For more information on the commands discussed in this chapter, see *z/VM: CP Commands and Utilities Reference* and *z/VM: CMS Commands and Utilities Reference*.

Appendix A. Keyboard Key/Function Conversions

IBM 3270 terminal keyboards (on which the z/VM user interface is based) and current "PC" (workstation and laptop) keyboards are somewhat different. The following chart shows the main differences between key appearances and key combinations required to perform common functions.

Table 1. Keyboard Key/Function Conversions
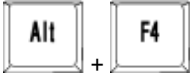
Function	3270	PC
ENTER INPUT	Enter	⏎
CLEAR SCREEN	Clear	Esc
FUNCTION KEYS 11-20	PF11 to PF20	↑ + F1 to ↑ + F10
FUNCTION KEYS 21-24	PF21 to PF24	Ctrl + F1 to Ctrl + F4
LOCK UPPERCASE	🔒	Caps Lock
UNLOCK UPPERCASE	↑	Caps Lock
RESET CONDITION	Reset	Alt + Scroll Lock
INSERT MODE	⬆	Ins
UNLOCK INSERT MODE	Reset	Ins
DELETE CHARACTER	a̶	Del

Table 1. Keyboard Key/Function Conversions (continued)		
Function	*3270*	*PC*
ERASE TO END OF FIELD	Erase EOF	Alt + F4

Notices

This information was developed for products and services offered in the US. This material might be available from IBM in other languages. However, you may be required to own a copy of the product or product version in that language in order to access it.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
US

For license inquiries regarding double-byte character set (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you provide in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
US

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

The performance data and client examples cited are presented for illustrative purposes only. Actual performance results may vary depending on specific configurations and operating conditions.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

Statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information may contain examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to actual people or business enterprises is entirely coincidental.

COPYRIGHT LICENSE:

This information may contain sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Trademarks

IBM, the IBM logo, and ibm.com® are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the web at IBM copyright and trademark information - United States (www.ibm.com/legal/us/en/copytrade.shtml).

Adobe is either a registered trademark or a trademark of Adobe Systems Incorporated in the United States, and/or other countries.

The registered trademark Linux® is used pursuant to a sublicense from the Linux Foundation, the exclusive licensee of Linus Torvalds, owner of the mark on a world-wide basis.

Terms and Conditions for Product Documentation

Permissions for the use of these publications are granted subject to the following terms and conditions.

Applicability

These terms and conditions are in addition to any terms of use for the IBM website.

Personal Use

You may reproduce these publications for your personal, noncommercial use provided that all proprietary notices are preserved. You may not distribute, display or make derivative work of these publications, or any portion thereof, without the express consent of IBM.

Commercial Use

You may reproduce, distribute and display these publications solely within your enterprise provided that all proprietary notices are preserved. You may not make derivative works of these publications, or reproduce, distribute or display these publications or any portion thereof outside your enterprise, without the express consent of IBM.

Rights

Except as expressly granted in this permission, no other permissions, licenses or rights are granted, either express or implied, to the publications or any information, data, software or other intellectual property contained therein.

IBM reserves the right to withdraw the permissions granted herein whenever, in its discretion, the use of the publications is detrimental to its interest or, as determined by IBM, the above instructions are not being properly followed.

You may not download, export or re-export this information except in full compliance with all applicable laws and regulations, including all United States export laws and regulations.

IBM MAKES NO GUARANTEE ABOUT THE CONTENT OF THESE PUBLICATIONS. THE PUBLICATIONS ARE PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, AND FITNESS FOR A PARTICULAR PURPOSE.

IBM Online Privacy Statement

IBM Software products, including software as a service solutions, ("Software Offerings") may use cookies or other technologies to collect product usage information, to help improve the end user experience, to tailor interactions with the end user, or for other purposes. In many cases no personally identifiable information is collected by the Software Offerings. Some of our Software Offerings can help enable you to collect personally identifiable information. If this Software Offering uses cookies to collect personally identifiable information, specific information about this offering's use of cookies is set forth below.

This Software Offering does not use cookies or other technologies to collect personally identifiable information.

If the configurations deployed for this Software Offering provide you as customer the ability to collect personally identifiable information from end users via cookies and other technologies, you should seek your own legal advice about any laws applicable to such data collection, including any requirements for notice and consent.

For more information about the use of various technologies, including cookies, for these purposes, see:

- IBM Privacy Statement at http://www.ibm.com/privacy/us/en/
- IBM Online Privacy Statement at http://www.ibm.com/privacy/details/us/en/ in the section entitled "Cookies, Web Beacons and Other Technologies"

Bibliography

This topic lists the publications in the z/VM library. For abstracts of the z/VM publications, see *z/VM: General Information*.

Where to Get z/VM Information

The current z/VM product documentation is available in IBM Documentation - z/VM (www.ibm.com/docs/en/zvm).

z/VM Base Library

Overview

- *z/VM: License Information*, GI13-4377
- *z/VM: General Information*, GC24-6286

Installation, Migration, and Service

- *z/VM: Installation Guide*, GC24-6292
- *z/VM: Migration Guide*, GC24-6294
- *z/VM: Service Guide*, GC24-6325
- *z/VM: VMSES/E Introduction and Reference*, GC24-6336

Planning and Administration

- *z/VM: CMS File Pool Planning, Administration, and Operation*, SC24-6261
- *z/VM: CMS Planning and Administration*, SC24-6264
- *z/VM: Connectivity*, SC24-6267
- *z/VM: CP Planning and Administration*, SC24-6271
- *z/VM: Getting Started with Linux on IBM Z*, SC24-6287
- *z/VM: Group Control System*, SC24-6289
- *z/VM: I/O Configuration*, SC24-6291
- *z/VM: Running Guest Operating Systems*, SC24-6321
- *z/VM: Saved Segments Planning and Administration*, SC24-6322
- *z/VM: Secure Configuration Guide*, SC24-6323
- *z/VM: TCP/IP LDAP Administration Guide*, SC24-6329
- *z/VM: TCP/IP Planning and Customization*, SC24-6331
- z/OS and z/VM: Hardware Configuration Manager User's Guide (www.ibm.com/servers/resourcelink/svc00100.nsf/pages/zosv2r4sc342670/$file/eequ100_v2r4.pdf), SC34-2670

Customization and Tuning

- *z/VM: CP Exit Customization*, SC24-6269
- *z/VM: Performance*, SC24-6301

Operation and Use

- *z/VM: CMS Commands and Utilities Reference*, SC24-6260

- *z/VM: CMS Primer*, SC24-6265
- *z/VM: CMS User's Guide*, SC24-6266
- *z/VM: CP Commands and Utilities Reference*, SC24-6268
- *z/VM: System Operation*, SC24-6326
- *z/VM: TCP/IP User's Guide*, SC24-6333
- *z/VM: Virtual Machine Operation*, SC24-6334
- *z/VM: XEDIT Commands and Macros Reference*, SC24-6337
- *z/VM: XEDIT User's Guide*, SC24-6338

Application Programming

- *z/VM: CMS Application Development Guide*, SC24-6256
- *z/VM: CMS Application Development Guide for Assembler*, SC24-6257
- *z/VM: CMS Application Multitasking*, SC24-6258
- *z/VM: CMS Callable Services Reference*, SC24-6259
- *z/VM: CMS Macros and Functions Reference*, SC24-6262
- *z/VM: CMS Pipelines User's Guide and Reference*, SC24-6252
- *z/VM: CP Programming Services*, SC24-6272
- *z/VM: CPI Communications User's Guide*, SC24-6273
- *z/VM: ESA/XC Principles of Operation*, SC24-6285
- *z/VM: Language Environment User's Guide*, SC24-6293
- *z/VM: OpenExtensions Advanced Application Programming Tools*, SC24-6295
- *z/VM: OpenExtensions Callable Services Reference*, SC24-6296
- *z/VM: OpenExtensions Commands Reference*, SC24-6297
- *z/VM: OpenExtensions POSIX Conformance Document*, GC24-6298
- *z/VM: OpenExtensions User's Guide*, SC24-6299
- *z/VM: Program Management Binder for CMS*, SC24-6304
- *z/VM: Reusable Server Kernel Programmer's Guide and Reference*, SC24-6313
- *z/VM: REXX/VM Reference*, SC24-6314
- *z/VM: REXX/VM User's Guide*, SC24-6315
- *z/VM: Systems Management Application Programming*, SC24-6327
- *z/VM: TCP/IP Programmer's Reference*, SC24-6332
- *z/VM: z/Architecture Extended Configuration (z/XC) Principles of Operation*, SC27-4940
- *CPI Communications Reference*, SC26-4399
- *Common Programming Interface Resource Recovery Reference*, SC31-6821
- z/OS: IBM Tivoli Directory Server Plug-in Reference for z/OS (www.ibm.com/servers/resourcelink/svc00100.nsf/pages/zosv2r4sa760169/$file/glpa300_v2r4.pdf), SA76-0169
- z/OS: Language Environment Concepts Guide (www.ibm.com/servers/resourcelink/svc00100.nsf/pages/zosv2r4sa380687/$file/ceea800_v2r4.pdf), SA38-0687
- z/OS: Language Environment Debugging Guide (www.ibm.com/servers/resourcelink/svc00100.nsf/pages/zosv2r4ga320908/$file/ceea100_v2r4.pdf), GA32-0908
- z/OS: Language Environment Programming Guide (www.ibm.com/servers/resourcelink/svc00100.nsf/pages/zosv2r4sa380682/$file/ceea200_v2r4.pdf), SA38-0682
- z/OS: Language Environment Programming Reference (www.ibm.com/servers/resourcelink/svc00100.nsf/pages/zosv2r4sa380683/$file/ceea300_v2r4.pdf), SA38-0683

- z/OS: Language Environment Runtime Messages (www.ibm.com/servers/resourcelink/svc00100.nsf/pages/zosv2r4sa380686/$file/ceea900_v2r4.pdf), SA38-0686
- z/OS: Language Environment Writing Interlanguage Communication Applications (www.ibm.com/servers/resourcelink/svc00100.nsf/pages/zosv2r4sa380684/$file/ceea400_v2r4.pdf), SA38-0684
- z/OS: MVS Program Management Advanced Facilities (www.ibm.com/servers/resourcelink/svc00100.nsf/pages/zosv2r4sa231392/$file/ieab200_v2r4.pdf), SA23-1392
- z/OS: MVS Program Management User's Guide and Reference (www.ibm.com/servers/resourcelink/svc00100.nsf/pages/zosv2r4sa231393/$file/ieab100_v2r4.pdf), SA23-1393

Diagnosis

- *z/VM: CMS and REXX/VM Messages and Codes*, GC24-6255
- *z/VM: CP Messages and Codes*, GC24-6270
- *z/VM: Diagnosis Guide*, GC24-6280
- *z/VM: Dump Viewing Facility*, GC24-6284
- *z/VM: Other Components Messages and Codes*, GC24-6300
- *z/VM: TCP/IP Diagnosis Guide*, GC24-6328
- *z/VM: TCP/IP Messages and Codes*, GC24-6330
- *z/VM: VM Dump Tool*, GC24-6335
- z/OS and z/VM: Hardware Configuration Definition Messages (www.ibm.com/servers/resourcelink/svc00100.nsf/pages/zosv2r4sc342668/$file/cbdm100_v2r4.pdf), SC34-2668

z/VM Facilities and Features

Data Facility Storage Management Subsystem for VM

- *z/VM: DFSMS/VM Customization*, SC24-6274
- *z/VM: DFSMS/VM Diagnosis Guide*, GC24-6275
- *z/VM: DFSMS/VM Messages and Codes*, GC24-6276
- *z/VM: DFSMS/VM Planning Guide*, SC24-6277
- *z/VM: DFSMS/VM Removable Media Services*, SC24-6278
- *z/VM: DFSMS/VM Storage Administration*, SC24-6279

Directory Maintenance Facility for z/VM

- *z/VM: Directory Maintenance Facility Commands Reference*, SC24-6281
- *z/VM: Directory Maintenance Facility Messages*, GC24-6282
- *z/VM: Directory Maintenance Facility Tailoring and Administration Guide*, SC24-6283

Open Systems Adapter

- Open Systems Adapter-Express Customer's Guide and Reference (www.ibm.com/support/pages/node/6019492), SA22-7935
- Open Systems Adapter-Express Integrated Console Controller User's Guide (www.ibm.com/support/pages/node/6019810), SC27-9003
- Open Systems Adapter-Express Integrated Console Controller 3215 Support (www.ibm.com/docs/en/SSLTBW_2.1.0/com.ibm.zos.v2r1.ioa/ioa.htm), SA23-2247
- Open Systems Adapter/Support Facility on the Hardware Management Console (www.ibm.com/docs/en/SSLTBW_2.1.0/com.ibm.zos.v2r1.ioa/ioa.htm), SC14-7580

Performance Toolkit for z/VM

- *z/VM: Performance Toolkit Guide*, SC24-6302
- *z/VM: Performance Toolkit Reference*, SC24-6303

RACF® Security Server for z/VM

- *z/VM: RACF Security Server Auditor's Guide*, SC24-6305
- *z/VM: RACF Security Server Command Language Reference*, SC24-6306
- *z/VM: RACF Security Server Diagnosis Guide*, GC24-6307
- *z/VM: RACF Security Server General User's Guide*, SC24-6308
- *z/VM: RACF Security Server Macros and Interfaces*, SC24-6309
- *z/VM: RACF Security Server Messages and Codes*, GC24-6310
- *z/VM: RACF Security Server Security Administrator's Guide*, SC24-6311
- *z/VM: RACF Security Server System Programmer's Guide*, SC24-6312
- *z/VM: Security Server RACROUTE Macro Reference*, SC24-6324

Remote Spooling Communications Subsystem Networking for z/VM

- *z/VM: RSCS Networking Diagnosis*, GC24-6316
- *z/VM: RSCS Networking Exit Customization*, SC24-6317
- *z/VM: RSCS Networking Messages and Codes*, GC24-6318
- *z/VM: RSCS Networking Operation and Use*, SC24-6319
- *z/VM: RSCS Networking Planning and Configuration*, SC24-6320
- z/OS: Network Job Entry (NJE) Formats and Protocols (www.ibm.com/servers/resourcelink/svc00100.nsf/pages/zosv2r4sa320988/$file/hasa600_v2r4.pdf), SA32-0988

Prerequisite Products

Device Support Facilities

- Device Support Facilities (ICKDSF): User's Guide and Reference (www.ibm.com/servers/resourcelink/svc00100.nsf/pages/zosv2r4gc350033/$file/ickug00_v2r4.pdf), GC35-0033

Environmental Record Editing and Printing Program

- Environmental Record Editing and Printing Program (EREP): Reference (www.ibm.com/servers/resourcelink/svc00100.nsf/pages/zosv2r4gc350152/$file/ifc2000_v2r4.pdf), GC35-0152
- Environmental Record Editing and Printing Program (EREP): User's Guide (www.ibm.com/servers/resourcelink/svc00100.nsf/pages/zosv2r4gc350151/$file/ifc1000_v2r4.pdf), GC35-0151

Index

IBM®

Product Number: 5741-A09

Printed in USA