
HP 3000 Guide for the New User



**Manual Part No.
32033-90009**

Notice

The information contained in this document is subject to change without notice.

Copyright 1986
by Hewlett-Packard Co.

Hewlett-Packard makes no warranty of any kind with regard to this material, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. Hewlett-Packard shall not be liable for errors contained herein or for incidental or consequential damages in connection with the furnishing, performance, or use of this material.

Hewlett-Packard assumes no responsibility for the use or reliability of its software on equipment that is not furnished by Hewlett-Packard.

Printing History

First Edition—January, 1986

Printed in U.S.A.

FCC

Federal Communications Commission Radio Frequency Interference Statement

Warning: This equipment has been certified to comply with the limits for a Class B computing device, pursuant to Subpart J of Part 15 of FCC Rules. Only peripherals (computer input/output devices, terminals, printers, etc.) certified to comply with the Class B limits may be attached to this computer. Operation with non-certified peripherals is likely to result in interference to radio and TV reception.

HP Computer Museum
www.hpmuseum.net

For research and education purposes only.

Table of Contents

Chapter 1:

First Session

Turning It On.....	1-1
Getting Started.....	1-4
Making Corrections.....	1-6
Helpful Messages.....	1-8
Identifying Yourself (Logging On).....	1-9
Logging Off.....	1-11
Logging On Again.....	1-13
Typing and Correcting.....	1-13
The SHOWTIME Command.....	1-14
Correcting With REDO.....	1-15
The SHOWME Command.....	1-20
Commands.....	1-21
Looking Back.....	1-23

Chapter 2:

Word Processing

What It Does.....	2-1
The EDITOR Command "Family".....	2-2
Starting to Write.....	2-2
The ADD And // Commands.....	2-3
Saving With KEEP.....	2-5
Using END To Leave.....	2-6
LISTF Lists Your Files.....	2-7
Retrieving Files Using Text and List.....	2-8
Adding More To Your File.....	2-10
Defining Editor's Commands.....	2-12
Editing & Rearranging.....	2-15
Text Changes; D, I & R.....	2-15
Adding More Text.....	2-18
Locking A File.....	2-20
Inserting Lines.....	2-21
Adding File To File.....	2-23
Deleting Lines.....	2-25
Renumbering With GATHER.....	2-26
Moving Lines With GATHER.....	2-27

Moving Sets With GATHER.....	2-28
Coping A Line.....	2-30
Copying Many Lines.....	2-32
Shortcuts.....	2-34
Command Initials.....	2-35
Defaults And Answers.....	2-36
Additional Commands.....	2-37
Looking Back.....	2-38

Chapter 3:

The Keyboard

Meet the Keyboard Again.....	3-1
The Terminal Controls.....	3-3
Soft Reset.....	3-3
Hard Reset.....	3-4
The Function Keys.....	3-6
Switching Off And On.....	3-7
Testing Block Mode.....	3-8
Function Controls.....	3-10
The Display Controls.....	3-12
The Arrow Repeater Keys.....	3-12
Moving The Work Window In Jumps.....	3-13
And Line By Line.....	3-14
Sending Cursor To File Ends.....	3-16
The Editing Keys.....	3-17
Character Manipulation.....	3-17
Line Manipulation.....	3-19
Numeric Keys.....	3-22
Graphics Mode.....	3-24
The Other Keys.....	3-26
Caps Key.....	3-26
The CTRL Key.....	3-29
The DEL/ESC "Escape" Key.....	3-30
The Shift Keys.....	3-32
The Print/Enter Key.....	3-32
Looking Back.....	3-33

Chapter 4:

Using Files

About Computer Files.....	4-1
Naming A File.....	4-3
Creating A File.....	4-4
Checking File Contents.....	4-6
Copying A File (FCOPY).....	4-7
Checking The Copy.....	4-10
Renaming A File.....	4-10
Removing A File.....	4-12
Printing a File.....	4-14
The Printing Request.....	4-15
The Printing Process.....	4-16
Help With MPE Commands.....	4-19
Controlling Scroll.....	4-20
Continuing HELP.....	4-22
Using Tape Storage.....	4-24
The STORE Procedure.....	4-25
The RESTORE Procedure.....	4-26
Shuffling File Data.....	4-27
Random Data File.....	4-28
Sorting By Name.....	4-30
Sorting By Date.....	4-33
Sorting Applications.....	4-35
Looking Back.....	4-37

Chapter 5:

Your Account

USER.ACCOUNT.GROUP.....	5-1
Everybody Has Files.....	5-3
Checking the Neighbors.....	5-5
The SHOWJOB Listing.....	5-6
Sessions And Jobs.....	5-8
The Operator And Console.....	5-10
Security for Your Work.....	5-12
Passwords When Logging On.....	5-13
Lockwords On Files.....	5-15
Remembering Lockwords.....	5-16

The Pub(lic) Groups.....	5-18
Creating A PUB File.....	5-20
File Addresses.....	5-22
Default File Addresses.....	5-23
Getting Files From A Distance.....	5-23
Using Your PUB File.....	5-25
UDC's — User-Defined Commands.....	5-27
The Available UDC's.....	5-28
User-Level UDC's.....	5-29
Creating A UDC File.....	5-30
Creating User UDC's.....	5-31
Testing Your UDC's.....	5-33
Nesting UDC's.....	5-35
Account-Level UDC's.....	5-38
System-Level UDC's.....	5-39
Looking Back.....	5-40

Chapter 6:

Messages

Your Communication Network.....	6-1
Messages to the Operator.....	6-2
A Message to Another User.....	6-4
Refusing Messages.....	6-6
Status Messages.....	6-8
The SHOW- Status Commands.....	6-9
Non-SHOW- Status Commands.....	6-10
“ERROR” Messages.....	6-12
“FAILURE” Messages.....	6-14
Looking Back.....	6-17

Index

Appendix A

1

First Session

Turning It On.....	1-1
Getting Started.....	1-4
Making Corrections.....	1-6
Helpful Messages.....	1-8
Identifying Yourself (Logging On).....	1-9
Logging Off.....	1-11
Logging On Again.....	1-13
Typing and Correcting.....	1-13
The SHOWTIME Command.....	1-14
Correcting With REDO.....	1-15
The SHOWME Command.....	1-20
Commands.....	1-21
Looking Back.....	1-23

First Session

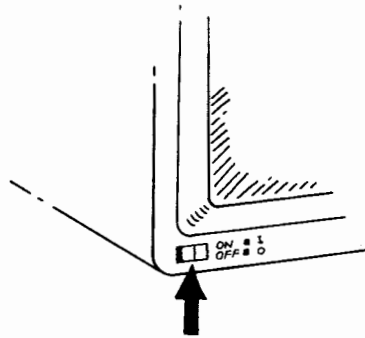
This book is for beginners. It's a guide to using an HP 3000 computer terminal. (For operating systems MPE V G.01.00 or later.)

Turning It On

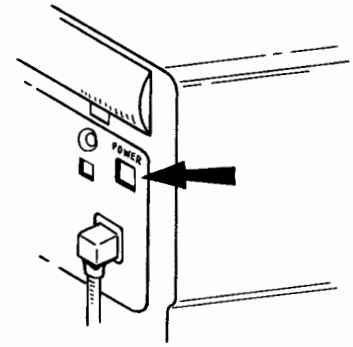
Step 1. Turn it on.

You may have to look around a bit to find the POWER or ON/OFF switch. It might be almost anywhere, front or back... as shown here.

HP 2392A (Front)



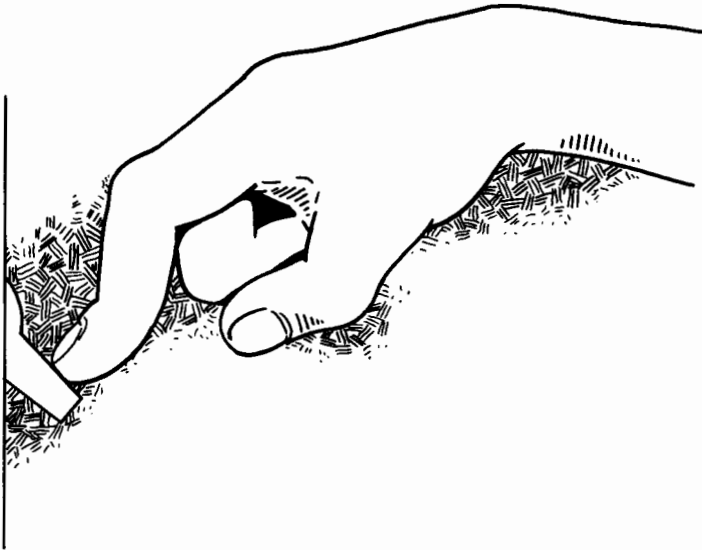
HP 150 (Back)



Step 2. Let it warm up.

This can take a minute or two, so be patient. Also, it may beep several times, and make other noises, so don't let that bother you.

Step 3. Now, turn it off, again.

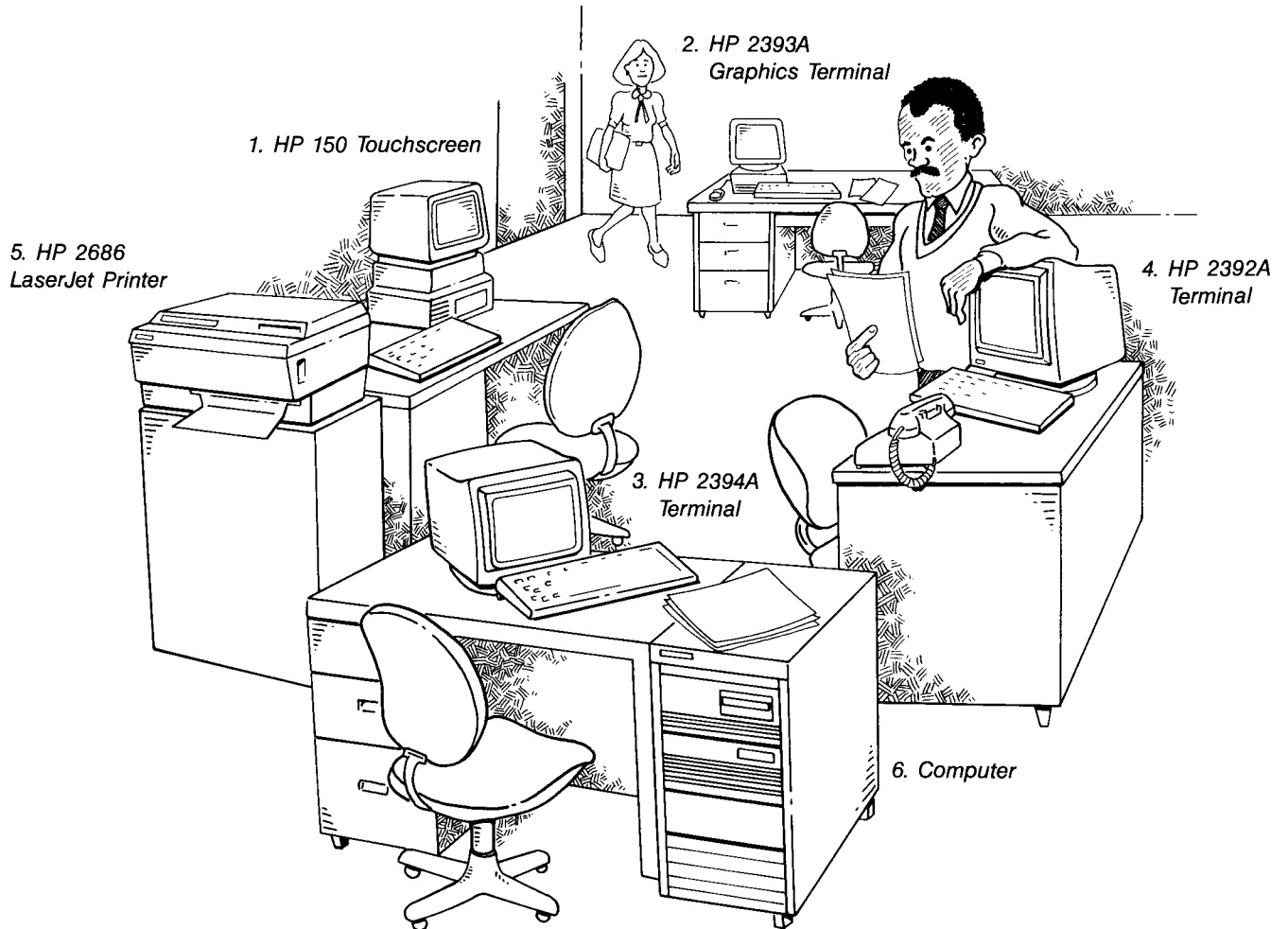


IMPORTANT:

Notice that you CAN turn it off whenever you want to. This doesn't harm it physically, and the other users aren't interrupted.

"It", by the way, is NOT the system computer. This thing you're getting ready to use is called a "terminal". It's like an extension of your computer; it gives you the ability to "talk" to the computer, to send it information, and to have it respond.

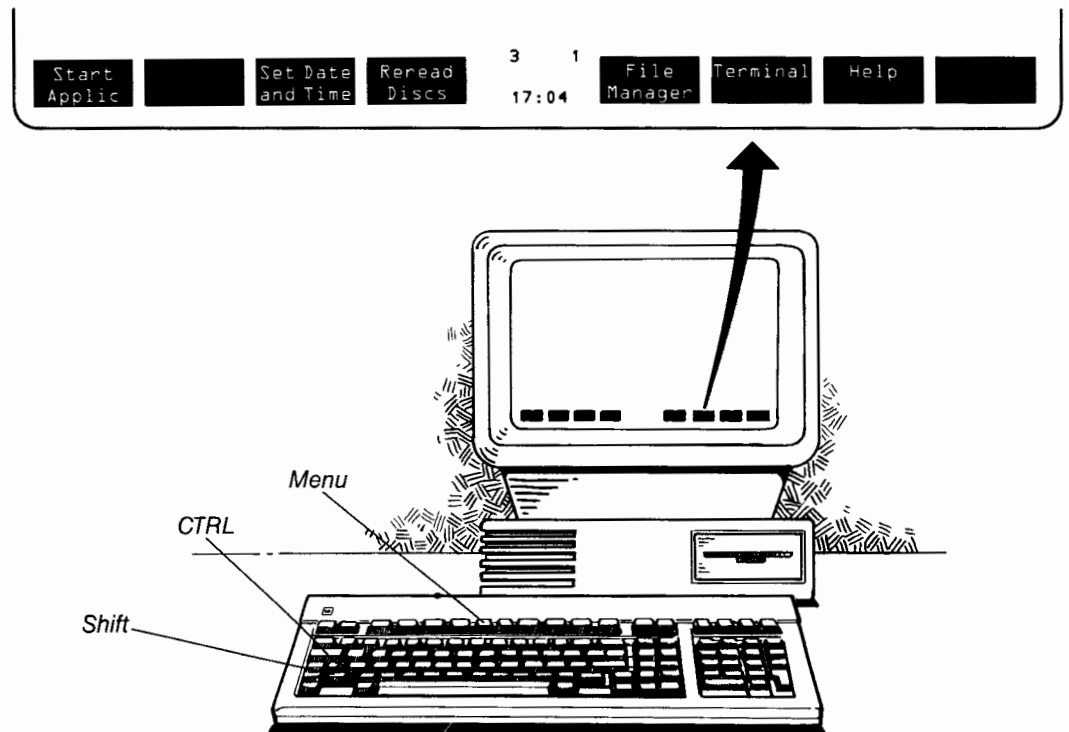
Most system computers will have at least a few terminals. In this picture, items 1, 2, 3 and 4 are different kinds of terminals, 5 is a printer, and 6 is the actual computer, itself. They're all connected to each other, as well as being plugged into electrical sockets.



Getting Started

Turn your terminal back on, and let it warm up again.

HP 150 USERS! If your terminal is a "Touchscreen", you may see some title blocks at the top of your screen, and the name "Personal Applications Manager" (P.A.M.) in one of them, and eight little message windows, like this, across the bottom. If so, touch the window that says "Terminal" with your fingertip, and wait for the screen to empty.



HP 150 USERS! Next, it's VERY important to turn OFF the "touchscreen" feature. It's simply too distracting while you're learning. To turn it off, SIMULTANEOUSLY press the two keys: **CTRL** and **Shift** and, while you're holding them down, also press: **Menu**. The picture shows where, on the keyboard, you should find those keys.

If you were successful, you should see the message "TouchOff" down in the lower right-hand corner of the screen. If you don't see it, press the keys again, until it does appear.

You can turn "Touchscreen" back on again at any time, by pressing those same keys. It will also come back on again each time you turn the HP 150 off, then on again.

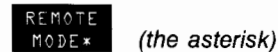
As you can see, your screen isn't entirely empty. You'll probably see at least these three kinds of things on it.

1. A blinking hyphen, or block, or other figure, usually in the upper left-hand corner. This is the "cursor". Things that you type, later, will appear here, at the cursor, one letter at a time.
2. A row of eight message windows, like this:



IMPORTANT:

There should be a little star, or asterisk, in the window where you see the words "REMOTE MODE". If there isn't, find the key marked "f4." Press it until the window looks like this:



3. There may also be other numbers and messages on your screen. They can be ignored for the moment.

Making Corrections

The first thing you do is press this key: **Return**. Immediately, you should see, on the screen:

```
:_ (a colon, in addition to your blinking cursor)
```

IMPORTANT:

If no colon appears, press **Return** again, sharply, once or twice. . . or even more often, if necessary. If there's still no colon, ask your Operator to check your terminal.

You now have the computer's attention, and it's time to type your first message. Use the terminal's keyboard just as though you're using a typewriter.

Misspell "hello" by pressing the five keys: **H E L K P**.

IMPORTANT:

Don't type **H E L K P** in capital letters. Those "keycaps" (or letters in squares) just show the proper sequence of keys, but not their case. For now, type all letters in lowercase.

In this book, after you've typed something, you'll usually be shown what it should look like on your screen. The book shows an outline of the screen's lower edge, as on previous pages, or its upper edge, as seen below:

```
:helkp_
```

Correct the spelling. Press: **Backspace** **Backspace** **L** **Q**

IMPORTANT:

Did the screen go “dead” after just a minute or so? Don’t worry! That’s normal! Colons only stay “fresh” for that long. Whenever you’re ready for another fresh one, just press **Return again.**

If you pressed those keys, and **ONLY** those keys, the letters on your screen should now look like this:

```
:hello_
```

IMPORTANT:

But, if the screen shows this, instead:

```
:hello
HELLQ
EXPECTED [SESSION NAME,] USER.ACCT (etc.)
```

it probably means you pressed the Return key after you made the correction. It’s just a helpful message (see below). It’s all right to continue on.

If all went well, you see that you can use “Backspace” to back the cursor up, for any distance, and then retype over your mistakes.

Try experimenting! For instance, type your name and address in lower case letters, then backspace, and add capitals. Freshen the colon as necessary. (Remember, use only the space bar and Backspace key to move the cursor back and forth.)

Helpful Messages

The computer tries to give you as much help as it can. If it notices that you're unsure what to do next, or have forgotten a procedure, it'll try reminding you. To see such a message, try this.

Press: **Return** again, for a fresh colon. Then, misspell "hello" again, plus a another Return: **HELK P Return**

```
:helkp
EXPECTED HELLO, :JOB, :DATA, OR (CMD) AS LOGON.
(CIERR 1402)
:
```

EXPLANATION: the computer EXPECTED you to say HELLO, or to start a JOB, to give it some DATA, or to use a COMMAND in PARENTHESES. These are all alternative ways to do a thing called LOGON... a contraction of "log on". "Logging on" is a computer term. It means identifying yourself to the computer so it'll accept your commands.

IMPORTANT:

Messages like this are the computer's way of trying to help you. Some are easy to understand, others are obscure. And, if you don't understand one, DON'T WORRY! It's safe to continue.

NOTE: Some are incorrectly called "error" messages. They aren't YOUR errors, however. They usually just mean that someone expected you to type something different, right there. So, try to make allowances.

Identifying Yourself (Logging On)

You're a new user. The computer probably doesn't know your name. To temporarily identify yourself (computer people say: to log on), you can use this generalized three-part "name": USER.ACCOUNT,GROUP.



For even better identification, you can add your own name (if it has fewer than nine letters) like this: PAT,USER.ACCOUNT,GROUP.

IMPORTANT:

Do NOT use the name PAT. Use your own, or none. Also, remember to type every single keycap, including the space, the commas, and the period.

For a fresh colon, press: **Return**

Briskly, before the minute or so “freshness” ends, type:

```
HELLO PAT,USER.ACCOUNT,GROUP Return
```

↑ (remember, use your own name, not PAT)

Your screen should look sort of like the example shown below... although there can be variations from computer to computer.

```
:hello pat,user.account,group
HP3000 / MPE (ID numbers/letters)(day, date & time)
(followed by other identification messages)
(wait, and eventually you'll see another colon and cursor)

:_ (the new colon... computer people call it a "prompt")
```

IMPORTANT:

If your screen stayed “dead”, press the **Return** key again for a fresh colon. Then, log on again within the one minute time limit.

IMPORTANT:

Do you see this, instead?

```
:hello pat,user.account,group
HELLO PAT,USER.ACCOUNT,GROUP
NON-EXISTENT ACCOUNT (CIERR 1437)
```

This means you’re using an early version that’ll need a minor change before you can continue. Ask your System Manager to turn to the last page of this book (Appendix A) and use it to install the generic identity.

IMPORTANT:

If you got yet another result, check: did you type a space after HELLO; a period after USER; commas after your name and ACCOUNT? Did you spell it all correctly? Did you add spaces between the words where they shouldn't be? Or... was there a different reason completely, and can you tell what it was from the computer's message?

You can't really hurt anything, so try things that might work.

If all went well, you're "logged on".

Congratulations! You're now an actual computer user! You've started what is called a "session", in which you and the computer can now "talk" to each other. And you've learned some useful things, like making changes, understanding some of the computer's messages, and logging on.

A helpful reminder: later, if you want to find this part of the book again, to refresh your memory, you can look in the Index under many headings, like: HELLO, logging on, identifying yourself, and others. Try using the Index right now, just for practice.

Logging Off

Logging on used the easy-to-remember command HELLO. Logging off uses the equally well-chosen command BYE. Both are simple to use.

Try it, just for practice. Type: **B Y E** **Return**

The actual numbers on your screen may be different, but it should look something like this.

```
:bye
CPU=2. CONNECT=12. (plus day, date & time)
— (the cursor alone; no colon, or "prompt")
```

You've logged off. The terminal is still on, still doing everything it was doing before you logged on, but the computer has stopped "listening" for messages from you.

It leaves you with the report that, so far, your session has used 2 seconds of the Central Processing Unit's time, (CPU) and has lasted 12 minutes (CONNECT).

IMPORTANT:

You can log off any time you want to during sessions!

You can also take breaks when YOU want to take them... stand up, leave the terminal, get coffee, have lunch... it'll wait.



IMPORTANT:

In logging off, you'll lose any current work, unless you "save" it. More about this later.

Logging On Again

Now, log back on again, to continue this first session.

Again, press **Return** for your colon prompt, then type:

```
HELLO PAT.USER.ACCOUNT.GROUP Return
      ↑ (remember, use your own name)
```

```
:hello pat.user.account.group
HP3000 / MPE (ID numbers/letters). (day, date & time)
(standard identification and messages)

:_      (colon prompt & cursor)
```

This identity that you're using, USER.ACCOUNT,GROUP, is actually a "generic" identity. The computer accepts this while you're learning to use the system. You're logging on as a generic user named USER, in a generic account called ACCOUNT, and in a generic group named GROUP. Later on you'll have a real identity, like PAT.JONES,MARKETING. This is all explained in the section "Your Account", but for the time being, you'll use these generics.

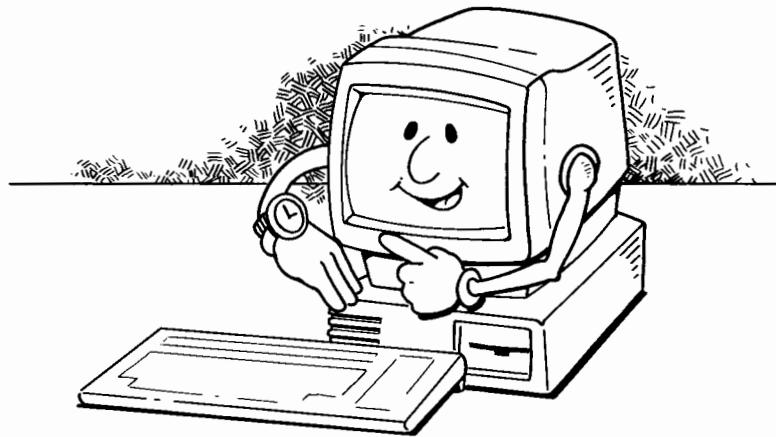
Typing and Correcting

You've seen how much like a typewriter's keyboard this one is. Perhaps a bit different in feeling, but certainly the same layout in the central keys. And, by now, you've probably even gotten used to seeing things appearing on screen instead of on paper. But it may still be a little strange to have it "answer back" the way it does.

Try exploring that a bit.

The SHOWTIME Command

You're going to ask the computer what time it is, using the SHOWTIME command.



IMPORTANT:

This time you'll notice that it isn't necessary to start by pressing Return for a new colon prompt. The reason is that once you're logged on, you don't need fresh colons every time. The one you have will stay fresh all day long.

Type: `SHOWTIME` Return

Immediately, the computer "answers":

```
:showtime  
TUE, JAN 8, 1986, 9:37 AM  
(or whatever the date and time are)
```

IMPORTANT:

If something else happened, check your spelling, or the helpful computer message.

Is the information correct? Is that really the correct day and time of day? If it isn't, and the Operator is handy, you might want to ask whether your computer's clock has been "set".

Wait a few minutes, then try it again, once or twice. Do you see the time changes happen?

Correcting With REDO

You've learned how to make corrections to the commands you type, as long as you catch the error BEFORE pressing Return. But, if you've slipped up, and pressed it before you've made the corrections, you've already seen the message: "UNKNOWN COMMAND NAME".



To make corrections AFTER pressing Return, you use the REDO command. You'll position your cursor by using the backspace key and space bar ONLY, and then correct with the letter subcommands.

To see how it works, deliberately misspell the command name "showme" with a couple of incorrect letters by typing: **C H Ø W E M** Return

```
:chowem
^
UNKNOWN COMMAND NAME. (CIERR 975)
—
```

IMPORTANT:

If your work isn't exactly like the example above, don't worry. Just take your new errors into consideration, and figure out how to correct them, too.

Type: **R E D Ø** Return

```
:redo
CHØWEM      (the computer retrieves the misspelled
—           command for rework; puts it in caps)
```

IMPORTANT:

REDO has 4 subcommands.

- **D** Deletes the character above it.
- **I** Inserts character(s).
- **R** Replaces character with new.
- **U** Undoes earlier D, I, or R.

They're put in place with the help of the cursor, in this step-by-step process.

First, replace the C with an S. You **MUST** place the subcommand letter directly under the letter to be changed. Fortunately, the cursor's already positioned under C, so you'll just use subcommand "R" (Replace) followed by the desired letter S. Watch what happens!

Type: **R S Return**

```
CHOWEM      (the misspelling)
rs
SHOWEM      (s replaces C)
—
```

The rest of it is a two-step process that inserts a new M between the W and E, then deletes the final M. Use the space bar to put the cursor under the E, then use subcommand "I" (Insert) to put the M in place.

Type: **I M Return**

The result should be:

```
SHOWEM
  im
SHOWmEM      (m is inserted; E & M move to the right)
—
```

You'd normally finish by placing the cursor under the last M, and using sub-command "D" (Delete) to remove it. But, first, deliberately make this mistake: delete the E instead of the M, by moving the cursor only five spaces, instead of six.

Watch carefully as this happens.

Type: `D` `Return`

```
SHOWmEM
  d
SHOWmM    (E is deleted; M moves to the left)
```

Cancel this error with "U" (Undo). Type: `U` `Return`

```
SHOWmM
u
SHOWmEM    (the E returns)
```

IMPORTANT:

If this all seems to be getting out of control, just press `Return` for a colon prompt. You can either start over, or skip ahead to the next part: The SHOWME Command. It's all right.

Or, continue the correction to "Showme". Move the cursor under the last M, and use subcommand "D" (Delete).

Type: ■■■■■D Return

```
sHOWmEM
      d
sHOWmE      (the last M goes away)
```

PLEASE... don't press Return yet.

Now, check what you've done. The word "sHOWmE" should be on your screen, properly spelled. Any original letters will still be in caps, the new letters in lower case.

Remember: the REDO command, and its four subcommands, let you make step-by-step changes in any command that you misspell, AFTER you press Return.

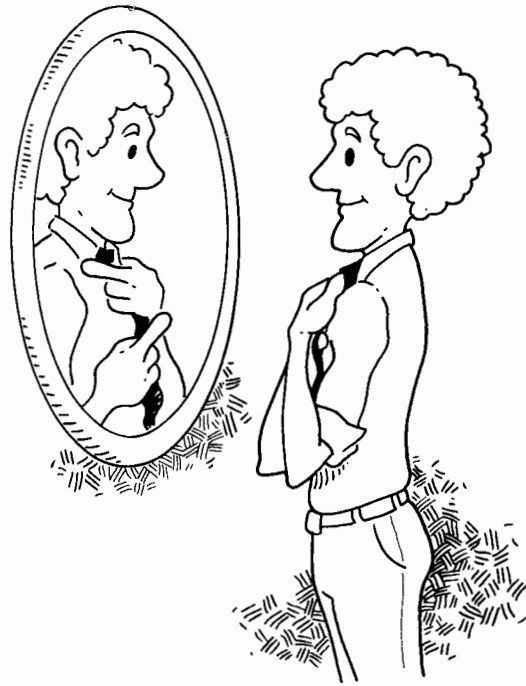
IMPORTANT:

If the REDO command process seems too complicated to bother with, you can accomplish the same thing by pressing Return again, and simply retyping the command.

The SHOWME Command

If you completed the last couple of pages, you'll have the word "sHOWmE" on your screen. You can now "give the computer this command" by pressing Return.

Press: `Return`



If, on the other hand, you DIDN'T finish the earlier steps, and would prefer to start over, type: `Return S H O W M E Return`

```

:SHOWME (or showme)
USER: #S(numbers),USER.ACCOUNT,GROUP (NOT IN BREAK)
MPE VERSION: HP (ID letters/numbers)
CURRENT: (today's date and time)
LOGON: (the date and time you logged on)
CPU SECONDS: (numbers) CONNECT MINUTES: (numbers)
#STDIN LDEV: (numbers) #STDLIST LDEV: (numbers)

```

It's a "status report". The computer is responding to the command "Showme" with a report, to you, on the status of this current session that you're conducting. It's really quite complete.

It does NOT include your name, if you used it (PAT,USER.ACCOUNT,GROUP). Your name will be more important to other status reports that you'll meet later, but it doesn't appear in this one.

You can ask for this status report, or the other (SHOWTIME) whenever you have a colon prompt.

Commands

You're heard the term "computer commands"? Well, congratulations... you've already used five!

- | | |
|------------|-------------------------------------|
| ■ HELLO | to log on, and start a session |
| ■ BYE | to log off, and end a session |
| ■ SHOWTIME | to tell you the date and time |
| ■ REDO | to correct misspelled commands |
| ■ SHOWME | to learn the status of your session |

A command is your demand that the computer do this thing for you. “This thing” can be as simple as telling you the time, or as complex as helping you make a group of text corrections.

There are over 100 commands that you can use. They’ll help you do just about everything you may have to do with this computer. There are even ways, later on, to make up your OWN commands.

As you learn more commands, these rules will help you.

- Understand their limitations.
- Remember when each is most appropriate to use.
- Spell them accurately.

Whenever you find yourself with terminal time, experiment with the commands you’ve learned so far.

Go ahead and end the session by logging off, using the BYE command, the way you learned earlier.

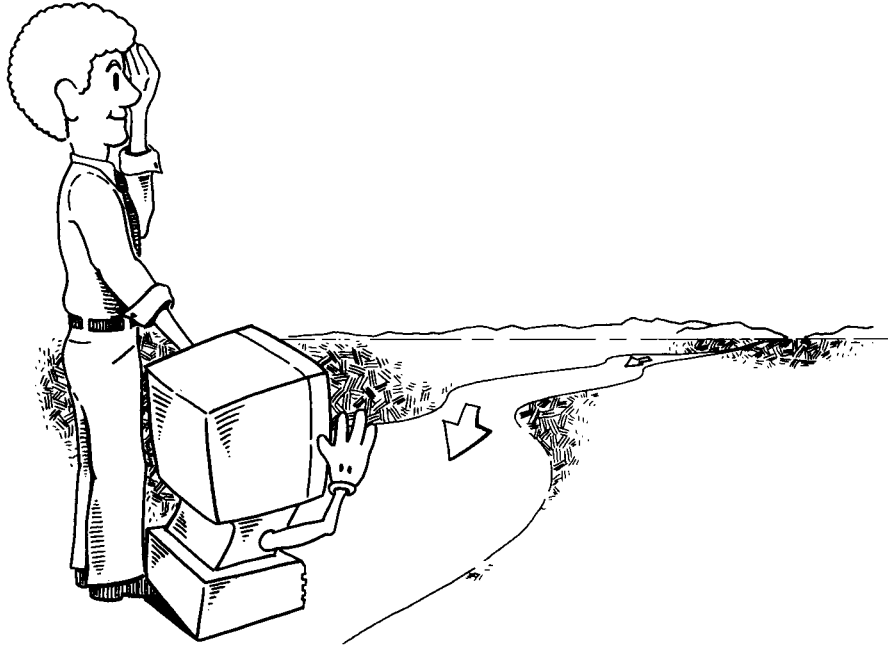
Press: **B** **Y** **E** **Return**

This marks an important point in the progress of any computer newcomer: you’ve started and successfully finished your first complete session! (A pleased smile wouldn’t be out of place.)

Looking Back

If you'd like to check yourself on some of the things you covered during this first session, you might enjoy this review.

1. Is ACCOUNT a command? (Yes/No)
2. Using REDO, which actual keys do you press to change SHOWWTME to SHOWTIME?
3. What actual sequence of keys do you press in order to log on?
4. Can you log off in the middle of a session? (Yes/No)



ANSWERS

1. No
2. `REDO` Return, then: `II` Return
3. Return HELLO USER.ACCOUNT.GROUP Return
Did you remember the first Return, for the prompt?
4. Yes

What It Does.....	2-1
The EDITOR Command "Family".....	2-2
Starting to Write.....	2-2
The ADD And // Commands.....	2-3
Saving With KEEP.....	2-5
Using END To Leave.....	2-6
LISTF Lists Your Files.....	2-7
Retrieving Files Using Text and List.....	2-8
Adding More To Your File.....	2-10
Defining Editor's Commands.....	2-12
Editing & Rearranging.....	2-15
Text Changes; D, I & R.....	2-15
Adding More Text.....	2-18
Locking A File.....	2-20
Inserting Lines.....	2-21
Adding File To File.....	2-23
Deleting Lines.....	2-25
Renumbering With GATHER.....	2-26
Moving Lines With GATHER.....	2-27
Moving Sets With GATHER.....	2-28
Coping A Line.....	2-30
Copying Many Lines.....	2-32
Shortcuts.....	2-34
Command Initials.....	2-35
Defaults And Answers.....	2-36
Additional Commands.....	2-37
Looking Back.....	2-38

Word Processing

What It Does

Word Processing means using your terminal to:

- write things... letters, memos, reports, and others;
- edit them... make changes to the written things;
- store the written work... file it someplace;
- retrieve it from those files; and
- put the final work on paper, in order to use it.

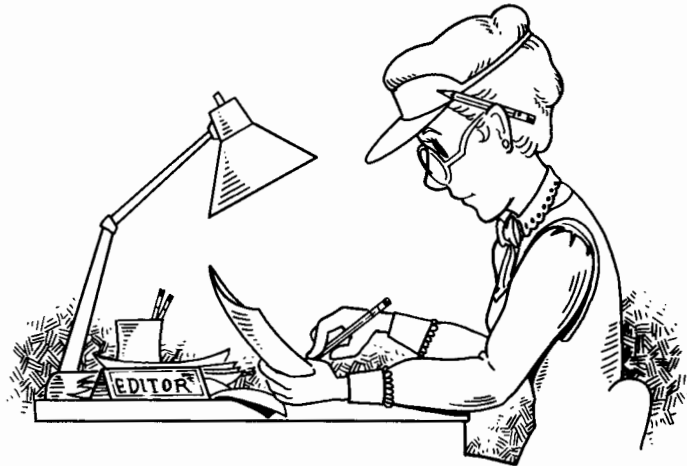
These are certainly very normal, familiar tasks. You'll just be doing them in less familiar ways.

The least familiar part of it is having your work appear on a screen instead of on paper. The nice part of it is that it's so easy to change things on the screen until you're happy with them, all without wasting paper.

Remember the REDO command? It's like that, but much more capable. It's called the EDITOR command.

The EDITOR Command “Family”

EDITOR is a very hardworking command. You might even call it a command “family”. Your Operator would call it an entire subsystem of your computer Operating System (My goodness!). It has dozens of its own commands, some quite elaborate. It even uses three of the four REDO subcommands (so they WERE worth learning).



Starting to Write

Did you log off, or turn your terminal off after the first chapter? If so, get things back to normal in the following ways.

HP 150 USERS! If you turned your terminal off, you probably reactivated the touchscreen feature. Please get rid of it by, again, simultaneously holding down the **CTRL** and **Shift** keys, then pressing the **Menu** key. Check that you can see the “TouchOff” message in the lower righthand corner of your screen.

If necessary, log back on by typing:

Return H E L L O U S E R . A C C O U N T , G R O U P Return

After the messages, type: E D I T O R Return

IMPORTANT:

When you're using the EDITOR command family, a slash mark prompt (/) takes the place of the colon prompt(:).

```
HP32201A.#.## EDIT/3000 (day, date & time)
(C) HEWLETT-PACKARD CO. 1984
/ (this is the slash mark prompt)
```



The ADD And // Commands

To get started, type: A D D Return

This should look like this on your screen:

```
/add
1 _ (blinking cursor)
```

The computer's ready for you to start typing. Each line that you type will have its own line number. The "1", above, is the first. The cursor, as usual, is where your typing will appear.

In capital letters, type: T A B L E A Return Return

```
1 TABLE A
2
3 _
```

IMPORTANT:

If you make mistakes during the following exercise, don't worry. Just use the Return key to produce fresh lines, and go on from there. Keep the extra line(s) in mind and correct for them, or factor them in later.

IMPORTANT:

Soon, you'll learn to erase or purge any mistakes you may be making here, even after you've filed them away.

IMPORTANT:

If you've had other computer experience, don't try to use the arrow keys while in EDITOR. They just "seem" to be working, but aren't.

For this next part, use a mixture of capital and lower case letters. Type every keycap shown, including all the spaces. Use space bar and Backspace key for positioning.

type: ADD ADDS LINES Return
 then: KEEP SAVES YOUR WORK Return
 then: END LEAVES EDITOR Return
 and: / / CLOSES THE ADD Return

3	ADD	Adds Lines
4	KEEP	Saves Your Work
5	END	Leaves EDITOR
6	//	Closes The ADD
7	—	

Finally, finish the additions by typing: `// Return`

```

6      //      Closes The ADD
7      //      (and, sure enough, it does)
...
/_      (line numbers end; slash prompt returns)

```

Saving With KEEP

You've now actually used the ADD and // commands to start and end the adding process. Next, you'll practice the KEEP command. This tucks your work away into the computer for later use. You'll use your own name as a label.

Type: `KEEP PAT Return`

↑ (please, use your own name, not PAT; name can have a maximum of 8 letters)

```

/keep pat
/_      (sometimes takes awhile to appear)

```

IMPORTANT:

Did you get the following message, instead?

```

/keep pat
PAT ALREADY EXISTS - RESPOND YES TO PURGE OLD AND KEEP NEW
PURGE OLD?

```

This is just a coincidence. An earlier person, doing this same thing, used your name after the KEEP command. To get a new slash prompt, type: `YES Return`

```
PURGE OLD?yes  
/_
```

You've just kept your work by filing it away under a label consisting of your own name. You'll prove that to yourself later. Now it's time to leave EDITOR.

Using END To Leave

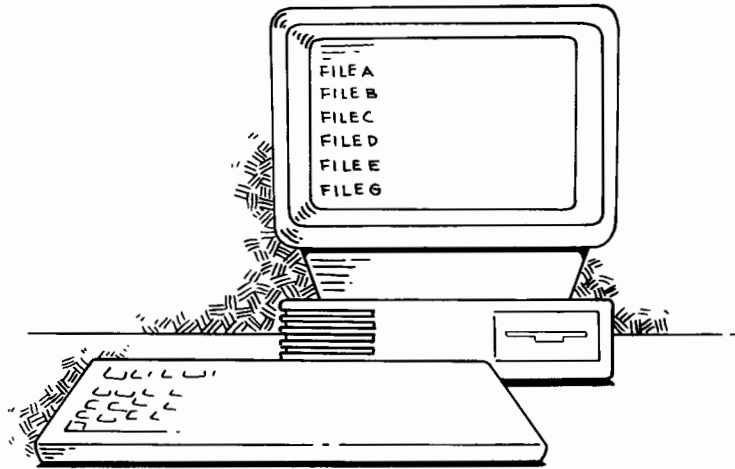
Type: **E N D** Return

```
/end  
END OF SUBSYSTEM (signoff)  
:_ (the main system's colon prompt returns)
```

Notice that it's easy to end things. END terminates a word processing session as simply as // finishes an ADD, or as the BYE command gets you out of a general session.

LISTF Lists Your Files

Now, it's time to prove that using KEEP did tuck your work safely away. If so, it's now actually a file with your name on it (or the name you used). It's filed with other files that earlier new users have stored.



Ask the computer to show you a list of them.

Type: **L I S T F** Return

```
:listf
FILENAME
PAT      (your name; and probably many others)
```

IMPORTANT:

If there **ARE** other names, yours won't be first, unless it happens to be there alphabetically.

IMPORTANT:

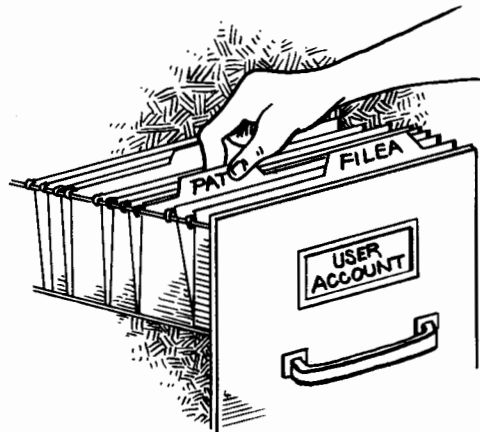
If you didn't get any FILENAME list at all, check back: did you type LISTF after a slash prompt (/)? If so, use END to leave EDITOR, then try again after a colon prompt (:).

LISTF is your command to the computer, asking it to list all the files in the "GROUP" part of your "ACCOUNT" that your "USER" identity is allowed to know about. (You'll recognize all THAT as your log-on identification.)

As mentioned, if there's more than one file listed, they'll be arranged alphabetically.

Retrieving Files, Using TEXT and LIST

You've seen that your work file is in the computer's memory, where it's supposed to be. Now, can you get it back onto your screen? Yes, with the TEXT command.



Reenter EDITOR, then ask for the file with your name on it, which was referred to as PAT, by

typing: **E D I T O R** Return

then: **T E X T P A T** Return

↑ (your name, not PAT)

```
:editor
HP32201A.#.## EDIT/3000 (day, date and time)
(C) HEWLETT-PACKARD CO. 1984
/text pat (this asks for file PAT)
/ (this appears when the file's at hand)
```

Then, type: **L I S T A L L** Return

```
/list all (this asks that every line be shown)
1 TABLE A
2
3 ADD Adds Lines
4 KEEP Saves Your Work
5 END Leaves EDITOR
6 // Closes The ADD
/_
```

The TEXT command gets your file, and brings it up to be worked on. The LIST command lets you define how much of the file you want shown:

- LIST 6 shows only a single line... line 6
- LIST 3/5 shows three lines... lines 3, 4 & 5
- LIST ALL shows every line in the file

Try them, and see. Type: **L I S T** **6** Return

and then: **L I S T** **3 / 5** Return

```

/list 6
    6      //      Closes The ADD
/list 3/5
    3      ADD      Adds Lines
    4      KEEP     Saves Your Work
    5      END      Leaves EDITOR
/_

```

Experiment with this a bit more until you're comfortable with it. Order every combination you can think of. You'll probably discover that you can't list things backwards, like LIST 5/3, and that just typing LIST by itself gives you whatever line is next.

Adding More To Your File

If you want to add things to the end of what you already have, it's simple enough. Use ADD, of course.

Type: **A D D** Return

```

/add
    7      (a brand new line)

```

Then, you can put in the new commands you're using: TEXT, LIST and MODIFY. Use caps and lower case, as shown.

Type: `TEXT HAS A FILE` Return

then: `LIST PRESENTS THE FILE` Return

then: `MODIFY MAKES CHANGES` Return

and finish with: `//` Return

```

7      TEXT  Has A File
8      LIST  Presents The File
9      MODIFY Makes Changes
10     //
...
/_

```

NOTE: To check if a thing really did get included, type:

`LIST ALL` Return

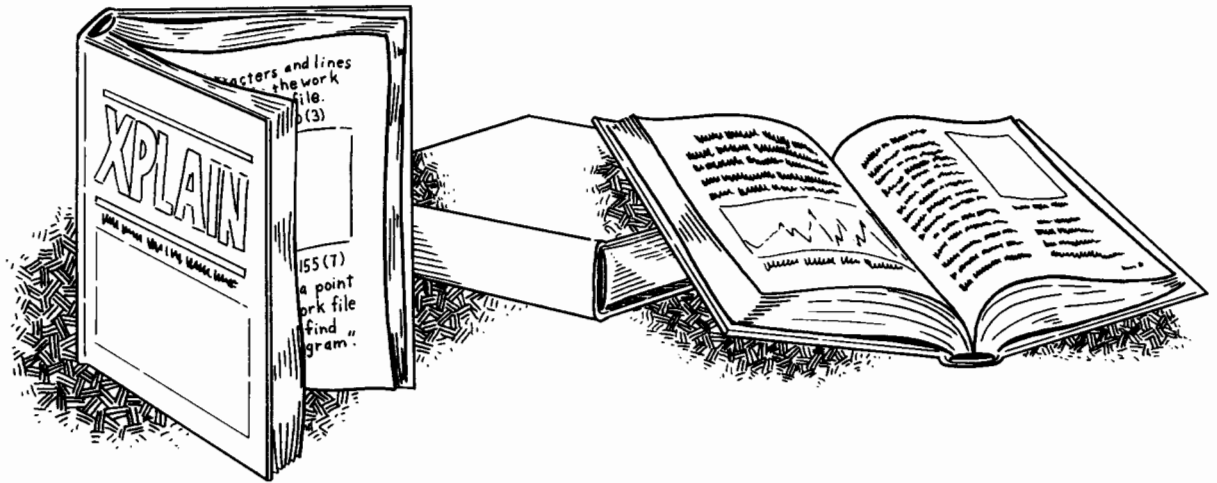
IMPORTANT:

Whenever you have to leave EDITOR in the middle of a project, save the unfinished work with KEEP. You can return later to finish it.

Defining Editor's Commands

You've already seen, and used, a half dozen of EDITOR's commands. And, thanks to your entries in TABLE A, you know, sort of, what each one does.

To help understand, there's a special command that does nothing but define the rest of EDITOR's commands. The name of this special command is XPLAIN.



Let's try it, to see if it really works.

Type: **X P L A I N K E E P** Return

```
/xplain keep
KEEP: TO SAVE ALL OR PART OF THE WORK FILE IN A USER FILE
EXAMPLE: KEEP EDIT02
```

IMPORTANT:

**Did you get this message on your screen: *1*INVALID
COMMAND NAME? Was that because you spelled it EXPLAIN
(with an E)? Try it again, but with an X.**

The KEEP definition seems pretty self-evident. Let's try another one... END.

Type: **X P L A I N E N D** Return

```
/xplain end
END: TO TERMINATE EXECUTION OF EDIT/3000 OR, WHEN USED
WITH A MATCHING BEGIN COMMAND, TO TERMINATE AN ITERATIVE
OPERATION
EXAMPLE: END (ALSO SEE "BEGIN" EXAMPLE)
```

Again, sort of understandable. END lets you leave EDIT/3000 (which is EDITOR's formal name) or, along with a BEGIN command, helps end a repetitive task.

Go ahead and check out the others you know, including XPLAIN, itself. Some, like these, will be clear. Others may have words or terms that are new to you.

IMPORTANT:

This only defines EDITOR commands. It won't work with MPE commands: HELLO, SHOWME, and others. Try it.

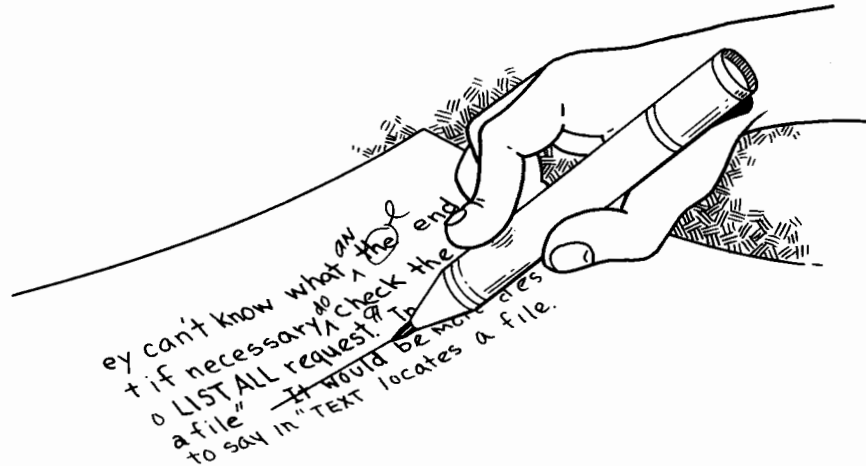
Remember, you DON'T HAVE TO UNDERSTAND these definitions. They're just meant to be helpful, but if they aren't, they aren't. Anyway, you almost certainly know the central meanings of the commands, just through your good common sense!

NOTE: You don't have to spell out XPLAIN. Just use an X. Example: X ADD. Go ahead, try it. Type : `X ADD Return`

Editing & Rearranging

If necessary, check the status of your work file with a LIST ALL request.

In line 7, it's not quite accurate to say "TEXT Has A File". It would be more descriptive to say "TEXT Locates A File". As the first editing task, then, change it to read that way.



Text Changes; D, I & R

Use the 1-letter subcommands from REDO to make the correction. Remember them? This makes use of them, but by typing MODIFY first, instead of REDO.

- D Deletes character(s) above
- I Inserts character(s)
- R Replaces character with new

Line 7 is the one you want to change. Tell the computer you want to **MODIFY** it, like this.

Type: **M O D I F Y** **?** Return

```
...
/modify ?
MODIFY ?
TEXT Has A File
```

IMPORTANT:

Remember that corrections have to be made one step at a time. After each step you can see what you accomplished by pressing Return. This also clears the way for the next step. When you're done, leave the MODIFY step by pressing Return twice.

So, use the space bar to move the cursor under the H. Put d's under the beginning and ending points of the part you want to delete.

Do this by typing: `■■■■■■■■D■■■D Return`

TEXT Has A File
 d d (*d's appear under H and the space*)
TEXT A File (*Has, and its following space, are gone*)

Next, use the space bar to move the cursor under A, and Insert "Locatis", (which is "Locates", but deliberately misspelled) by typing:

```

I L O C A T I S Return

```

```

TEXT    A File
        iLocatis
TEXT    Locatis A File

```

IMPORTANT:

Did the words run together, like: LocatisA File? Did you remember to add a space after Locates? If not, you can Insert one by moving the cursor under the A, then typing: **I** **Return**

Now, correct the misspelling by moving the cursor under the first "i", then using the R subcommand. Replace it with an "e" by typing:

```

R E Return

```

It should now look more or less like this:

```

TEXT    Locatis A File
        re
TEXT    Locates A File

```

Make any other changes that are needed. When you're happy with the accuracy of the line, press: **Return** again.

The slash prompt should return.

You might type LIST ALL, to be sure everything happened, but that should do it. If there's any question, go back over the steps, and see if you can find the reason.

If there are any errors, misspellings or that sort of thing, use your new knowledge of the D, I & R subcommands to correct them before going on.

Adding More Text

Save the work by typing: **K E E P P A T Return**
 ↑ *(please! your own name)*

Answer the computer's question with: **Y E S Return**

Clear your work area with: **D E L E T E A L L Return**

Start the new project with the familiar: **A D D Return**

IMPORTANT:

Once you've used KEEP, DELETE ALL does not purge the file from the computer's memory. It just erases whatever's left in the work area. It clears the decks before starting something new.

```
/keep pat
PAT ALREADY EXISTS - RESPOND YES TO PURGE OLD AND KEEP NEW
PURGE OLD?yes
/delete all
/add
  1  _
```

Next, type five lines of things. Stay roughly in the left-hand two-thirds of the screen. Press **Return** after every line. If you can't think of anything else, copy a book or a magazine.

At the end, on what should be line 6, add another command, by typing:

JOIN ADDS A FILE Return

This would look like:

```
/add
1      Now is the time for all good folks to come
2      to the aid of the split in their bathing suit.
3      Throughout this period, historians will later
4      say, everything seemed to come together in
5      support of this new technology, and the things
6      JOIN Adds A File
7      —
```

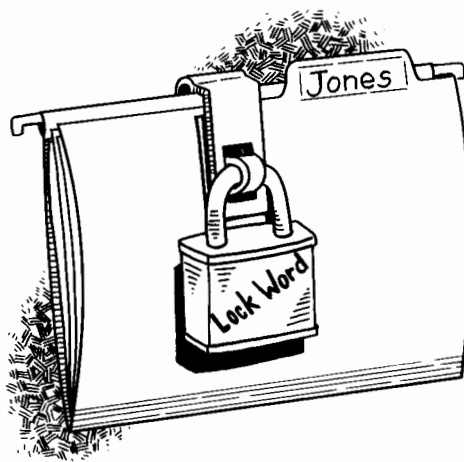
Finish, as usual, with: **// Return**

Use MODIFY and the letter subcommands to correct spelling errors. Also, practice editing so that the words read more smoothly.



Locking A File

Keep this file, giving it a different name. Use your LAST name, if it has eight letters or less. (Jones, for example. But don't actually USE Jones!)



Also, you're going to put a "lockword" on this file. This will keep anyone from using the file unless they know the right word. For your lockword, use one you remember easily, like your nickname, if it has eight letters or less. (Let's use "Arkie" to represent yours.)

This time, type: **K E E P J O N E S / A R K I E** Return
 ↑ (your last name & nickname)

IMPORTANT:

Never use a slash as part of your file name. The computer sees everything to the right of the slash as a lockword. This results in lockwords that you don't remember assigning.

```

      7      //
...
/keep jones/arkie
/_

```

Inserting Lines

Next, let's say you want to add another two commands, DELETE and GATHER, back in the file we're calling Pat. Let's also say you want to squeeze them into TABLE A between the // and TEXT commands, (lines 6 and 7).

For this clever trick, you'll use ADD again, but in a new way that allows incremental additions. You can squeeze in as many as 27, automatically, or as many as 999 by using the information in "Copying Many Lines", later in this chapter.

To relocate File Pat, type: **T E X T P A T** Return
 then: **A D D 6 . 1** Return

IMPORTANT:

Did you get this message from the computer: *14*INVALID LINE NUMBER? This probably means that, in typing 6.1, you used a lower-case L for the 1. That works on typewriters, but not computers. Try it again using a real number 1.

Type: **D E L E T E R E M O V E S L I N E S** Return

and: **G A T H E R M O V E S & M E L D S L I N E S** Return

End the ADD with: **//** Return

Finally, check what you've done by: **L I S T 6 / 7** Return

```
/text pat
/add 6.1
    6.1    DELETE Removes Lines
    6.2    GATHER Moves & Melds Lines
    6.3    //
...
/list 6/7
    6      //      Closes The ADD
    6.1    DELETE Removes Lines
    6.2    GATHER Moves & Melds Lines
    7      TEXT   Locates A File
/_
```

See how the computer accepts the direction "ADD 6.1" as an order to squeeze the following lines in between line 6 and line 7? You can use the same technique to add between any two line numbers.

Adding File To File

You know what the entire File Pat looks like by now. If not, use LIST ALL, and check.

Then, add File Jones onto the end of File Pat. To do that you'll need the JOIN command.

Type: `J O I N J O N E S` Return
 ↑ *(or whatever name you used)*

```
/join jones
LOCKWORD: JONES.GROUP.ACCOUNT?
```

Type: `A R K I E` Return
 ↑ *(or whatever nickname you used)*

IMPORTANT:

Your lockword WILL NOT APPEAR ON THE SCREEN. It will remain INVISIBLE as you type it. This helps keep it secret as you use it. If you mistype, don't worry. Repeat it until you get it right. You can also backspace to correct. At the end you should see this:

```
NUMBER OF LINES JOINED = 6
/_
```

Check the Pat File. Type: **L I S T A L L** Return

```

/list all
1      TABLE A
2
3      ADD      Adds Lines
4      KEEP     Saves Your Work
5      END      Leaves EDITOR
6      //       Closes The ADD
6.1    DELETE   Removes Lines
6.2    GATHER   Moves & Melds Lines
7      TEXT     Locates A File
8      LIST     Presents The File
9      MODIFY   Makes Changes
10     Now is the time for all good folks to come
11     to the aid of the split in their bathing suit.
12     Throughout this period, historians will later
13     say, everything seemed to come together in the
14     support of this new technology, and the things
15     JOIN      Adds A File
/_

```

If your Pat File DOESN'T look like this, for reasons of wear and tear and unexpected experimenting, this might be a good time to reconstruct it. Use this as a guide.

Deleting Lines

Removing entire lines of text or material from your work is done with the DELETE command. It works on individual lines, groups of lines, or even your entire project.

Following is a list of four different ways to specify what you want to delete.

IMPORTANT:

Please do NOT type the following examples. Just read them.

- DELETE 4 Removes a single line... line 4.
- DELETE 2,7 Removes two lines... 2 and 7.
- DELETE 3/6 Removes four lines... 3, 4, 5 and 6.
- DELETE ALL Removes your whole project.

Now, here are some real ones to type. At each step you'll be shown what's being taken away.

IMPORTANT:

Remember to use the number key for the 1's, NOT the lower-case L.

Take one line away by typing: `DELETE 11 Return`

two lines, with: `DELETE 12,14 Return`

and a range, with: `DELETE 10/13 Return`

Finally, check the results with: `LIST ALL Return`

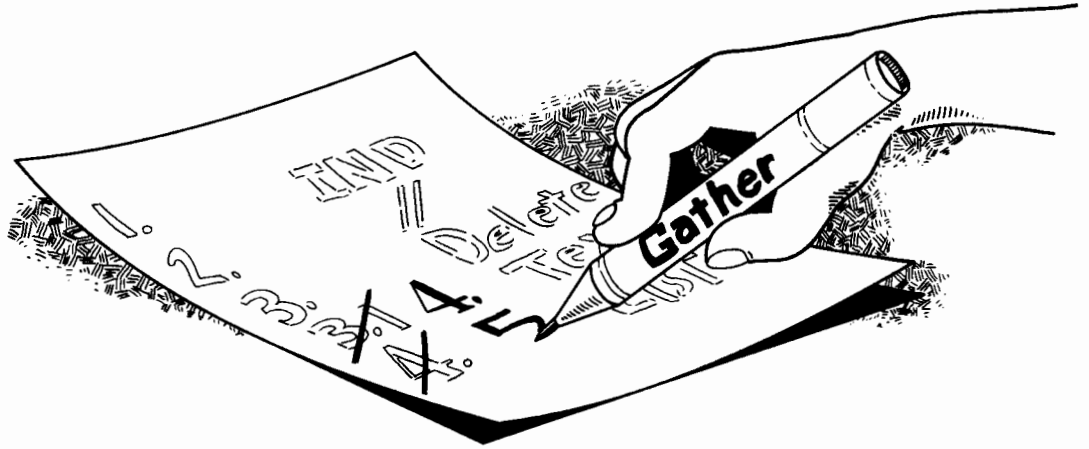
When you check, you find lines 10 through 14 are gone, and line 15, JOIN, is right under line 9, MODIFY.

Renumbering With GATHER

One thing the command GATHER does is to renumber the lines in your work file in their proper order. To see how this works,

type: `GATHER ALL` Return

and: `LIST ALL` Return



The result is effective:

```
/gather all
/list all
1      TABLE A
2
3      ADD      Adds Lines
4      KEEP     Saves Your Work
5      END      Leaves EDITOR
6      //       Closes The ADD
7      DELETE   Removes Lines
8      GATHER   Moves & Melds Lines
9      TEXT     Locates A File
10     LIST     Presents The File
11     MODIFY   Makes Changes
12     JOIN     Adds A File
```

See how everything smooths out? Each of the lines has been properly renumbered, and the “bumps” are gone.

Moving Lines With GATHER

As a matter of fact, about the only thing that’s still disorganized about TABLE A is its sequence. It might be nice to rearrange the commands alphabetically, starting with ADD and DELETE, and ending with TEXT and the // command.

The GATHER command lets you move lines around within your work! It’s a bit trickier than just typing GATHER ALL, but it’s so handy that it’s well worth learning.

This is the correct sequence for making it work:

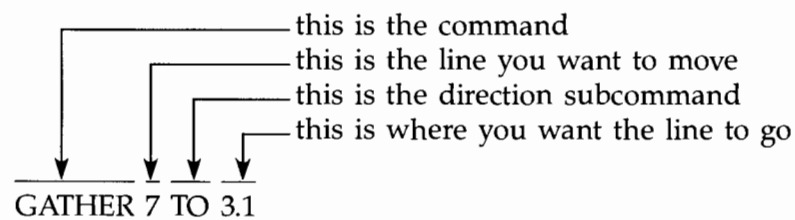


Diagram illustrating the syntax of the GATHER command:

- this is the command (points to GATHER)
- this is the line you want to move (points to 7)
- this is the direction subcommand (points to TO)
- this is where you want the line to go (points to 3.1)

GATHER 7 TO 3.1

If, as your first step in alphabetizing the table, you wanted to move DELETE (line 7) up and tuck it in under ADD (line 3), you’d do it just that way.

So, do it! Type: **G A T H E R ? T O 3 . 1 Return**

```
/gather ? to 3.1
?      =>    3.1    (which shows 7 did move)
/_
```

It isn’t necessary, but you could check it out with LIST ALL, noting the new line 3.1, and the missing line 7.

Moving Sets With GATHER

Alphabetically, END should follow DELETE. You can make it happen by removing KEEP (line 4) from between them, and dropping it down under JOIN, making it the new line 13.

IMPORTANT:

This will ONLY work if line 13 is still vacant. If your experimenting has added lines beyond 12, use the DELETE command, first, to get rid of them.

When it's all right, type: `GATHER 4 TO 13 Return`

You also know that LIST (line 10) and MODIFY (line 11) will follow KEEP, alphabetically. They're already on adjoining lines, so they can be moved together, with just

one request: `GATHER 10/11 TO 14 Return`

Then check it out with: `LIST ALL Return`

```
/gather 4 to 13
  4      =>   13
/gather 10/11 to 14
 10      =>   14
 11      =>   15
/list all
 1      TABLE A
 2
 3      ADD      Adds Lines
 3.1    DELETE  Removes Lines
 5      END      Leaves EDITOR
 6      //       Closes The ADD
 8      GATHER  Moves & Melds Lines
 9      TEXT     Locates A File
 12     JOIN     Adds A File
 13     KEEP     Saves Your Work
 14     LIST     Presents The File
 15     MODIFY  Makes Changes
/_
```

Do you see how it can be finished in just two more moves? Right. Move TEXT (line 9) down to line 16, and // (line 6) down to line 17.

But, there's a splashier way to do it, also in two moves. Drop // (line 6) down to line 10, and then insert the four-line group "12, 13, 14 & 15" in at 8.1! Try that, and see if it works.

IMPORTANT:

Notice that you CAN insert lines freely any place there's an opening in the sequence, like at 10, here. Otherwise you have to use increments.

Type: `G A T H E R 6 T O 10` Return

and: `G A T H E R 12/15 T O 8.1` Return

then: `L I S T A L L` Return

```

/gather 6 to 10
 6      =>      10
/gather 12/15 to 8.1
12      =>      8.1
13      =>      8.2
14      =>      8.3
15      =>      8.4
/list all
1      TABLE A
2
3      ADD      AddS Lines
3.1    DELETE  Removes Lines
5      END      Leaves EDITOR
8      GATHER  Moves & Melds Lines
8.1    JOIN     Adds A File
8.2    KEEP     Saves Your Work
8.3    LIST     Presents The File
8.4    MODIFY   Makes Changes
9      TEXT     Locates A File
10     //       Closes The ADD
/_

```

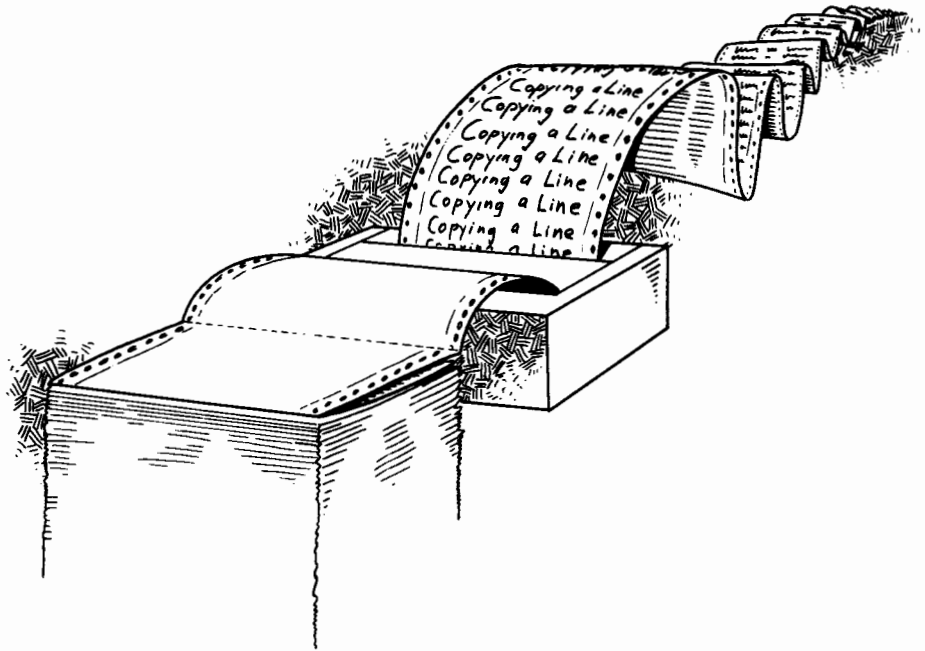
There they are, alphabetically arranged. The only step left is to renumber the lines, using the method you learned earlier.

Type: **G A T H E R A L L** Return

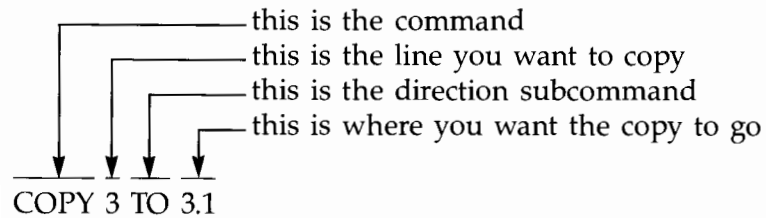
Then use LIST ALL to make sure the line numbers resequenced cleanly, from 1 to 12. Go ahead and do it.

Copying A Line

You haven't yet explored the COPY command. This copies everything from single lines to entire work areas, and puts them wherever you ask.



Like GATHER, it's complex. It needs certain information, in the right sequence, to do what you want. Here's the structure:



This is a convenience. For instance, if you copy line 3 (ADD), you can edit, or make other changes to that copy without forgetting what the original looked like.

To make the copy, type: **C O P Y 3 T O 3 . 1 Return**

and, to check: **L I S T A L L Return**

Your screen will look like this.

```

/copy 3 to 3.1
      3      =>      3.1      (the record of what happened)
/list all
1      TABLE A
2
3      ADD      Adds Lines
3.1    ADD      Adds Lines      (the copied line)
4      DELETE  Removes Lines
  
```

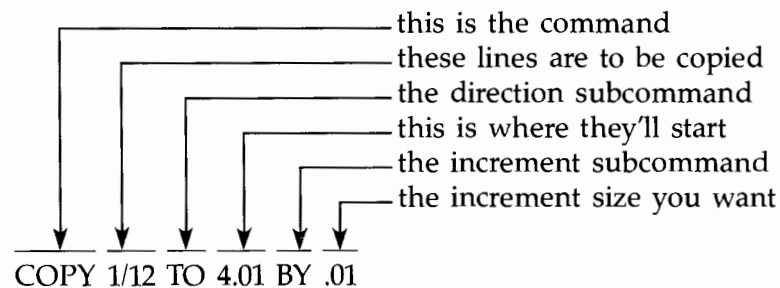
Copying Many Lines

You know you can insert lines between, let's say, 3 and 4 by having them numbered 3.1, 3.2, 3.3, etc. But this only works up to 3.9. To insert more than nine lines, you have to ask for increments of .01. For even larger numbers, you have to ask for increments of .001... or:

- .1 increments for: 1 to 9 more lines
- .01 increments for: 1 to 99 more lines
- .001 increments for: 1 to 999 more lines

The first one (.1) is the so-called "default" increment: the one you get automatically if you don't specify one of the others.

You add this item to the end of the command sequence, like this:



In the example shown, you're requesting a copy of the entire TABLE A, and you're asking that it be inserted after line 4, and that the line numbers be in increments of .01. Try it, watching your screen carefully to see what happens.

Type: `COPY 1/12 TO 4.01 BY .01` Return

then, to check on it: `LIST ALL` Return

```

/copy 1/12 to 4.01 by .01
 1      =>    4.01
 2      =>    4.02
 3      =>    4.03
 3.1    =>    4.04
 4      =>    4.05
 5      =>    4.06
 6      =>    4.07
 7      =>    4.08
 8      =>    4.09
 9      =>    4.1
10      =>    4.11
11      =>    4.12
12      =>    4.13

/list all
 1      TABLE A
 2
 3      ADD    Adds Lines
 3.1    ADD    Adds Lines
 4      DELETE Removes Lines
 4.01    TABLE A
 4.02
 4.03    ADD    Adds Lines
 4.04    ADD    Adds Lines
 4.05    DELETE Removes Lines
 4.06    END    Leaves EDITOR
 4.07    GATHER Moves & Melds Lines
                        (and so on, through:)
11      TEXT    Locates A File
12      //      Closes the ADD

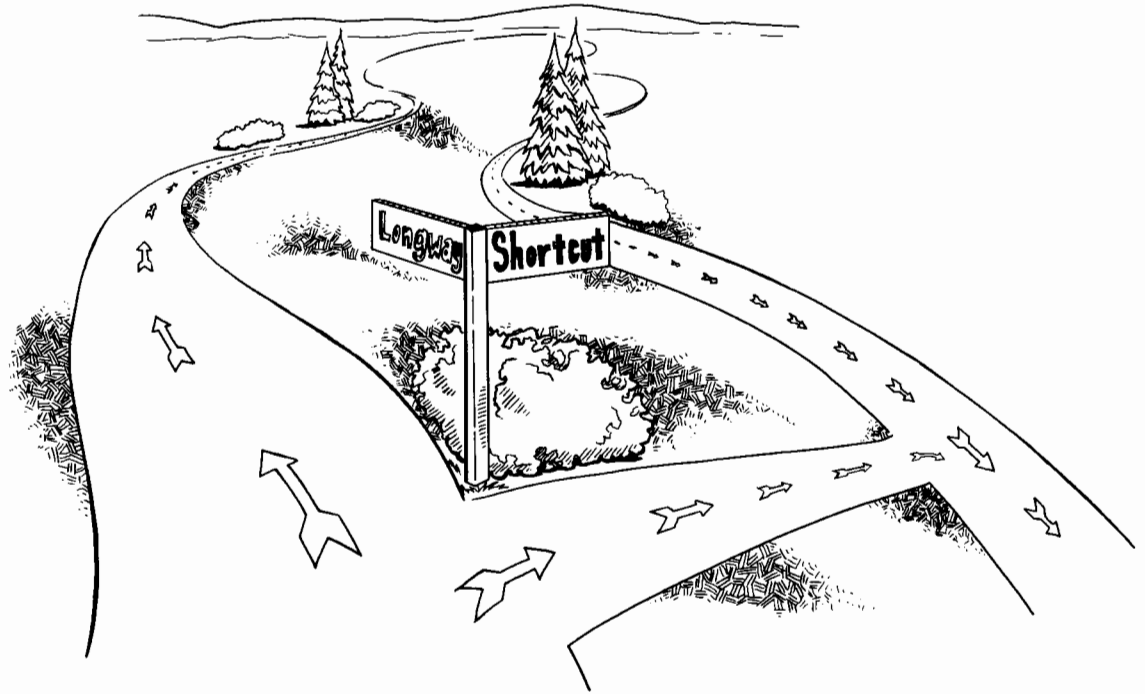
/_

```

A second way to do that same job would be to substitute "ALL" for the lines "1/12". This second way would look like this: `COPY ALL TO 4.01 BY .01`

Shortcuts

If you continue to use EDITOR for your word processing, you may appreciate learning some shortcuts.



These aren't mandatory. They're just to speed things up a bit, or take the edge off your impatience.

Command Initials

The computer will understand all but one of the EDITOR commands as soon as you type the first letter. Remember typing just "x" for XPLAIN? Well, all the commands but COPY work the same way. For COPY, you have to type two letters, "CO". The "C", by itself, is reserved for the command CHANGE, which you haven't met yet.

Let's try this, to get a feeling for it. Call up line 7 (JOIN) with the MODIFY command, so you can make changes in it. Normally, you'd type MODIFY 7 Return. This time just type: **M** **7** Return

```
/m 7
MODIFY      7
JOIN       Adds A File
```

It works! Press: **Return** to get the slash prompt back, and try another one: DELETE. Let's use it to get rid of the 12-line TABLE A that was inserted between lines 4 and 5, from 4.01 to 4.13. Normally you type DELETE 4.01/4.13 Return. This time, try: **D** **4.01/4.13** Return

NOTE: If you jumped the gun and did a GATHER ALL to renumber the file, just find out what the new line numbers are, and substitute them for the ones above.

Check with: **L** **A L L** Return (LIST ALL, right?)

Defaults And Answers

There are occasions when you press a command letter that there's only one possible action open to the computer. It will often figure that out for itself, and act.

An example. You're now working in File Pat (or whatever you named it). Try to KEEP it, but don't tell it what to keep. Type: **K Return**

```
/k
PAT          (it figured out what to keep)
PAT ALREADY EXISTS - RESPOND YES TO PURGE OLD AND
KEEP NEW PURGE OLD?_
```

Another minor effort-saver is abbreviated answers. In the computer's response, shown above, you're asked if you want to purge the previously-kept File Pat to make room for this new version. It wants a YES or NO answer. It will settle for a Y or N answer.

Try it. Type: **Y Return** (or, of course: **N Return**)

If YES, soon the slash prompt returns.

If NO, you are told: PURGE OF OLD FILE NOT CONFIRMED - TEXT NOT KEPT

Additional Commands

Can you believe it? You've already learned, and used, twelve EDITOR commands: ADD, COPY, DELETE, END, GATHER, JOIN, KEEP, LIST, MODIFY, TEXT, XPLAIN and //.

In this group you'll find most of the functions you need to do word processing. It helps to explore the additional abilities of each one, and it really helps to practice. And even if word processing isn't important in your work, this familiarity will help in other areas, like entering data, writing programs, and many other applications.

In addition to these commands, there are others...twenty-nine in all. As a matter of fact, there's one for every letter of the alphabet (isn't that neat), plus CO and //.

Why don't you use the X(PLAIN) command to check them out. And, remember: if you don't understand the explanations, don't worry; you can make do without them.

THE EDITOR COMMANDS

*A = ADD	*G = GATHER	N = NOT	U = USE
B = BEGIN	H = HOLD	O = OR	V = VERIFY
C = CHANGE	I = INSERT	P = PROCEDURE	W = WHILE
*CO = COPY	*J = JOIN	Q = Q	*X = XPLAIN
*D = DELETE	*K = KEEP	R = REPLACE	Y = YES
*E = END	*L = LIST	S = SET	Z = Z::=
F = FIND	*M = MODIFY	*T = TEXT	*//

* the commands you've learned

If you plan to do a lot of word processing with EDIT/3000 (the EDITOR subsystem), it'll pay to get a more detailed description of the commands and how to use them.

Looking Back

Another section completed. Another lot of useful information learned.
Another group of major milestones. It's not necessary, of course, but would you like to check what you learned?

1. Is ALL a command? (Yes/No)
2. Insert the COPY command (and definition: "Duplicates Lines") into TABLE A, using any appropriate shortcuts.
3. How do you lock a file "Payroll" with your nickname?
4. What sequence would you follow to change the MODIFY definition from "Makes Changes" to "Permits Changes"?
5. Is word processing used to write computer programs?

ANSWERS

- No.
- Type: A [] E . [] Return followed by:
C O P Y D U P L I C A T E S L I N E S Return / / Return
- K [] P A Y R O L L / A R K I E Return
 ↑ (*your nickname*)
- Start with: M [] I [] Return then place d's under the first and last letters of "Makes" then, one space before the C, insert:
I P E R M I T S Return
- It can be, yes.

Meet the Keyboard Again.....	3-1
The Terminal Controls.....	3-3
Soft Reset.....	3-3
Hard Reset.....	3-4
The Function Keys.....	3-6
Switching Off And On.....	3-7
Testing Block Mode.....	3-8
Function Controls.....	3-10
The Display Controls.....	3-12
The Arrow Repeater Keys.....	3-12
Moving The Work Window In Jumps.....	3-13
And Line By Line.....	3-14
Sending Cursor To File Ends.....	3-16
The Editing Keys.....	3-17
Character Manipulation.....	3-17
Line Manipulation.....	3-19
Numeric Keys.....	3-22
Graphics Mode.....	3-24
The Other Keys.....	3-26
Caps Key.....	3-26
The CTRL Key.....	3-29
The DEL/ESC "Escape" Key.....	3-30
The Shift Keys.....	3-32
The Print/Enter Key.....	3-32
Looking Back.....	3-33

The Keyboard

Meet the Keyboard, Again

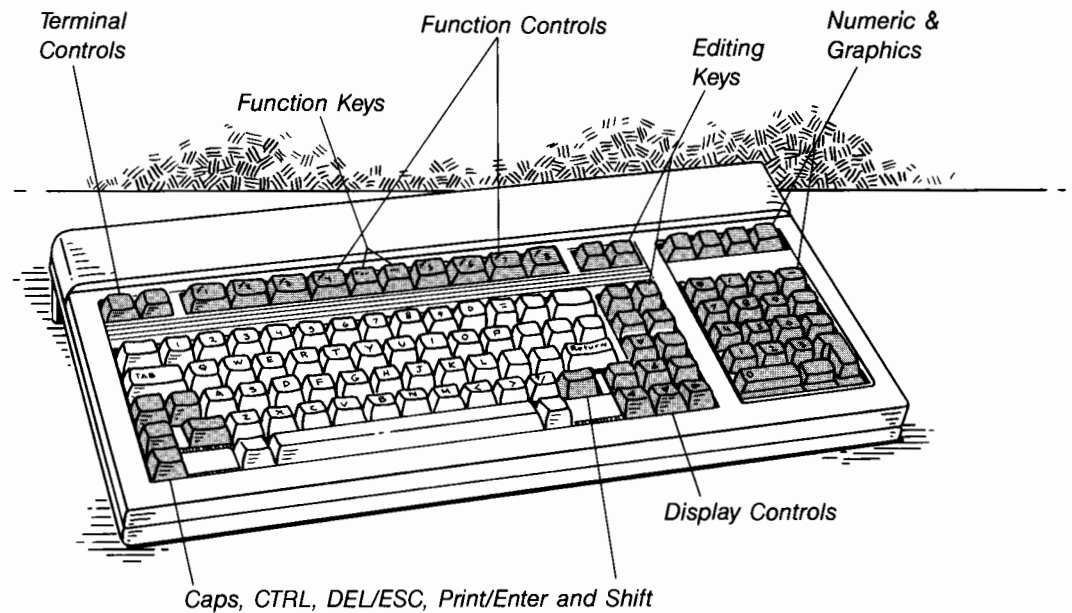
You've used the keyboard almost constantly, ever since you first met the terminal. You've gotten acquainted, and had a chance to see how much it's like a typewriter's keyboard. You've used it in a typewriter manner, so it probably HAS been a familiar and comfortable experience.

Of course, it's different, too. There are keys you've never seen on a typewriter keyboard. They remind you that this IS a computer terminal.

IMPORTANT:

This chapter describes the HP 46010A. This is the standard keyboard for HP 2392A terminals and HP 150 Touchscreens. If your keyboard is different, please make allowances.

3-2 The Keyboard



There are seven groups of keys shown here. Some, you may never need. Others are very helpful, and worth learning about. Each group is described in greater detail later.

- Terminal Controls reset the terminal
- Function Keys activate the message windows
- Function Controls label the function keys
- Editing Keys modify onscreen text
- Numeric Keys simplify number entry
- Display Controls relocate cursor & work "window"
- The Others Caps, CTRL, DEL/ESC, Print/Enter and Shift
 are dealt with individually

You'll have a chance to experiment with many.

The Terminal Controls

There are two, a “Stop” key, which you won’t use, and a “Reset/Break” key, which you might need.

Before starting this... is there anything on your screen? If not, go back into EDITOR, and bring your PAT file to the screen. Do you remember how?

Go ahead, do it.

Soft Reset

Next, simultaneously press the two keys:

Shift and **Reset/Break**

This is called “soft reset”. It’s supposed to unlock the keyboard, clear any error messages, turn off DISPLAY FUNCTIONS mode, stop any printer operation, and ring the terminal’s bell.

The things on your screen won’t look any different except that a colon prompt (:) will appear just below the latest slash prompt (/). If not, or if you missed the bell, try it again.

IMPORTANT:

When you’re having terminal troubles, this is often the first step you take.

To be sure that it worked, type: **S H O W M E** **Return**

The first line of the status report should show this:

```
/
:showme
USER: #S73,USER.ACCOUNT,GROUP (IN BREAK)
      ↑ (there it is!)
```

To get back to where you were, type: **E D I T O R** **Return**

When asked to abort, answer: **Y E S** **Return**

```
:editor
ABORT? (YES/NO)yes
PROGRAM ABORTED PER USER REQUEST. (CIERR 989)
(I.D.) EDIT/3000 (day,date,time)
(C) HEWLETT-PACKARD CO.
/      (and there you are, back in EDITOR)
```

Hard Reset

This next is a finger-stretcher. It's necessary to press "Reset/Break", "CTRL" and "Shift", in that order, and hold them all down simultaneously.

It's called "hard reset". It's supposed to return your terminal to the condition it would be in just after turning the power on.

Try it. Press: **Reset/Break** **CTRL** **Shift**

The screen should be empty except for a blinking cursor in the upper left-hand corner.

HP 150 USERS! Your screen may empty and, a bit later, show the P.A.M. selections, or however else it normally starts up. Go through your normal startup routine. You will also have to turn the Touchscreen feature back off again.

Use Return to get a colon prompt, then the SHOWME command for a status report.

Press: `Return` `S` `H` `O` `W` `M` `E` `Return`

```
:showme
SHOWME
USER: #s##,USER.ACCOUNT,PUB (IN BREAK)
MPE VERSION: HP32033G.##.##. (BASE G.##.##).
CURRENT: (day, date and time)
LOGON: (day, date and time)
CPU SECONDS: #          CONNECT MINUTES: #
#STDIN LDEV: ##          #STDLIST LDEV:  ## (etcetera)
```

To get back where you were, type: `E` `D` `I` `T` `O` `R` `Return`

When asked to abort, answer: `Y` `E` `S` `Return`

The screen should show this result:

```
:editor
ABORT? (YES/NO)yes

PROGRAM ABORTED PER USER REQUEST. (CIERR 989)
(I.D.) EDIT/3000 (day, date, time)
(C) HEWLETT-PACKARD CO.
/      (and there you are, back in EDITOR)
```

IMPORTANT:

This Hard Reset is another useful last resort, before turning the machine's power off, or pulling its plug.

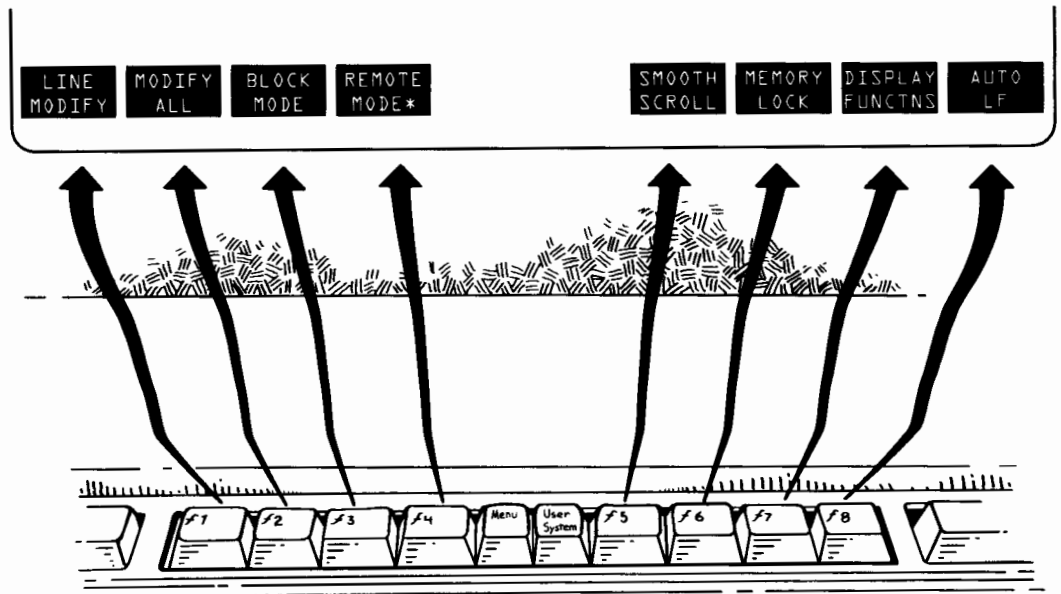
The Function Keys

These are among the more visible of the non-typewriter type keys. There are eight of them, in two sets of four, marked f1, f2, f3, and so on through f8. You will need these.

Remember when you first saw the lit screen... and noticed the row of message windows across the bottom? Eight of them, in two sets of four... as though they were related to the function keys? Well, they ARE related.

IMPORTANT:

Each function key **EXACTLY** corresponds to one of the eight "message windows" at the bottom of the screen, as shown below. Key f1 affects the lefthandmost window; f8, the righthandmost.



Those are the messages that are usually in the windows. This group mostly shows your terminal's "mode" settings. Think of them as toggle switches... or ON/OFF switches. They show whether each window is "on" or "off".

Switching Off & On

As you can see from the illustration, they are all in the off condition, except for the window with REMOTE MODE showing.

Following is a very brief description of what they are and what they do. (You don't really HAVE to know these things, but if you want more information about them, you'll find it in the Terminal User's Manual.)

- LINE MODIFY lets you edit a line in Character mode; it ends when line is sent to computer: press "Return" or "Print/Enter"
- MODIFY ALL is like Line Modify, but remains active after sending line
- BLOCK MODE sends nothing to computer until you press "Print/Enter"
- REMOTE MODE is for working with a computer; if off, it's in Local mode
- SMOOTH SCROLL flows work smoothly, vertically
- MEMORY LOCK this freezes every line from the cursor up, and lets you enter new lines below; gives Overflow protection when off
- DISPLAY FUNCTNS shows functions as graphics, without their action
- AUTO LF advances cursor to next line when you press Return

3-8 The Keyboard

While watching your own windows, try pressing each of the four left-hand function keys: `f1` `f2` `f3` `f4`

See? Every one you pressed has reversed its reading... or been toggled; the first three turned on (the asterisks appeared), the fourth turned off (the asterisk disappeared). To return them to the correct settings again, press: `f1` `f2` `f3` `f4`

Check to be sure everything is back where it belongs. If any of these settings are wrong for what you're doing, the terminal may not work, or will do strange, unexpected things.

Testing Block Mode

As an example, call up EDITOR again, and start a new file with the ADD command.

Type: `E` `D` `I` `T` `O` `R` `Return` and `A` `Return`

When line number 1 appears, turn BLOCK MODE on, type meaningless words, then press Return. Under normal circumstances this should cause line 2 to appear, ready to be typed on. Watch your screen as you do this.

Type: **f3**

then: **M E A N I N G L E S S W O R D S** Return

```
:editor
HP32201A.#.## EDIT/3000 (day, date and time)
(C) HEWLETT-PACKARD CO. 1984
/a
- 1 Meaningless words
```

You saw what happened. The cursor stayed on line 1, and just returned to the far left end of the line. Line 2 never appeared.

The terminal, which started out in Line mode, is now in Block mode. This means that pressing Return no longer sends the line you've been working on to the computer. The computer now accepts only blocks of stuff from your terminal.

Turn BLOCK MODE off by pressing: **f3**

Exit EDITOR by typing: **Return**

then: **/ /** Return

then: **E** Return

and then: **Y** Return

Function Controls

These are a pair... "Menu" and "User/System"... that are squeezed in between the two sets of function keys. Both may be helpful.

"Menu" turns the message windows off and back on. Watch.

Press: **Menu**

and, again: **Menu**

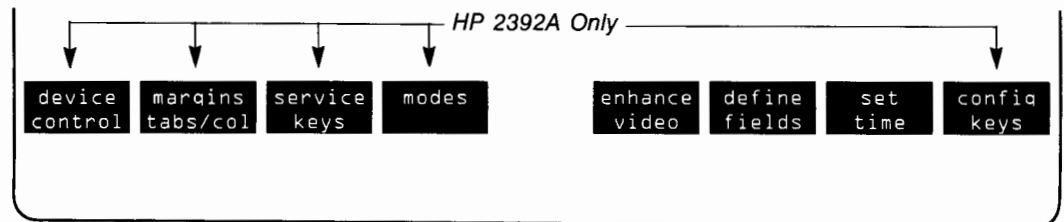
If you don't need the information in the message windows, and they're a distraction, you can turn them off.

- You can't use the screen space for anything else.
- They switch on again when you restart the terminal.

"User/System" changes the labels in the message windows to a so-called "basic" set. again, watch.

Press: **User/System**

If your terminal is an HP 150 Touchscreen, your message window contents should be as shown. If you have an HP 2392A, you'll only have the five indicated. If you have some other terminal you may have to refer to its manual for what's normal.



Each of these, when picked, leads to one or more levels of set-up toggle switches that can be turned on or off.

- device control selects a device to receive your work, and a route
- margins/tabs/col sets up margins, tabs & column
- service keys helps test the terminal or pick Monitor mode
- modes returns to the first label set: LINE MODIFY/MODIFY
ALL/BLOCK MODE/REMOTE MODE etc.
- enhance video sets up the screen's appearance
- define fields makes field definitions
- set time sets the time display
- config keys lets you pick your set-ups

IMPORTANT:

You can poke around in this maze of set-up toggles as long as you don't make any permanent commitments without knowing what you're doing.

If you want more detailed information about these toggles and what they do, check in the Terminal User's Manual.

To return the windows to normal, be sure that "modes" is showing in window f4, then press key: **f4**

The Display Controls

These are seven keys in the keyboard's lower right-hand corner. They include the "arrow" keys. They let you move the cursor around and, also, relocate your screen's work "window".

To experiment with them, it's easier to have something on the screen, so call up EDITOR and ask for your TEXT Pat File again, by typing:

E D I T O R **Return**

then: **T** **P A T** **Return**

and: **L** **A L L** **Return**

The Arrow Repeater Keys

When the file reappears, press the right-hand arrow: **▶** (the key in the lower right-handmost corner).

Press it again: **▶**

Notice that wherever your cursor was when you started, it moved one space to the right every time you pressed it. Which is just about what you'd expect from a right-hand key.

Next; press it, and HOLD it down: **▶**

See? It's a "repeater" key... the kind of thing you've probably seen on electric typewriters. The cursor just keeps moving to the right until it comes to the end of the line. Then, surprisingly, it appears in the upper left-hand corner of your screen. It then proceeds, line by line, downward.

You can stop its progress by releasing the key. Try it.

And, you can reverse the whole thing by holding down:  (the left-hand arrow key).

It, too, is a repeater key. The cursor moves from right to left... as you would expect from a left-hand key. In addition, when it finishes a line, it moves UP one!

Now, try pressing and holding the “up” key: .

Notice how it sends the cursor up the same column over and over again?

Finally, press and hold the “down” key: .

Sure enough, the cursor heads in the other direction.

Remember... the cursor marks the spot where you do your actual work. Now, using these arrow keys, you can put the cursor anyplace on the screen, and quickly.

Moving The Work Window In Jumps

You’ve already noticed that you work toward the bottom of the screen. After you’ve filled the screen, your work stays near the bottom. As each new line is added to the bottom, the whole “window” of your working area inches upward.

This is called “scrolling”: your work scrolls upward. And it’s normal to scroll one line at a time, this way.

Relocating the work window in chunks larger than a single line uses the keys “Prev” and “Next”.

"Prev", which means Prev(ious), moves the cursor back... or up... fifteen to twenty lines!

"Next" moves the cursor forward... or down... the same fifteen to twenty lines.

Try them. While watching the screen, press: **Prev**

and, again: **Prev**

then: **Next**

and, again: **Next**

Notice how the cursor moves to the top corner, then stays in the same spot on the screen, while the backgrounds shift?

Experiment with these keys until you feel comfortable with what they can do.

And Line By Line

You can move the work window in another helpful way, with a combination of two keys... the Shift key and either the up or down arrow key.

This results in a line-by-line scrolling, up or down.

Give it a try. Hold down: **Shift**

then press: **▲**

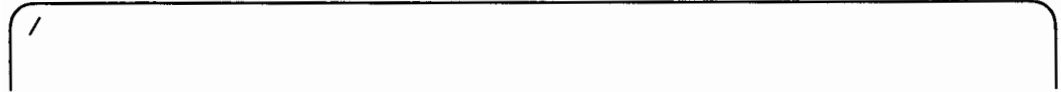
followed by: **▼**

If all went well, your cursor remained in the same spot while the background moved up one line, then down one line.

Now use its repeater quality.

While still holding down: **Shift**

also hold down: **▲**



The background moves up until everything has disappeared from sight except EDITOR's slash prompt, and whatever you may have added to that last line.

The down arrow works just as well, but in reverse. Your work should scroll slowly down, until it stops automatically. This marks the limit that your terminal can "remember" without the computer's help.

Try it. Experiment with them until you feel comfortable using this moving method, too.

Sending Cursor To File Ends

Finally, there's a technique for sending the cursor as far "back" and as far "forward" as possible with a single keystroke.

The key that looks like a hollow arrowhead, used alone, sends the cursor back as far as possible.

Try it. See how far back yours will go. Press: 

How far back it goes will depend upon what you're doing, how long your work file is, or how big your terminal's memory is. It might only go a few lines, or dozens of "pages".

Going as far forward as possible uses the hollow arrowhead plus the Shift key.

Try it. Hold down: **Shift**
then press: 

How far did your cursor go? It could have been as little as a line, or half a page, or to the end of a long file. This is really helpful if you're writing a LONG document of many pages. You can save it, then get back to your work area quickly.

The Editing Keys

This set of six keys is located at the upper right, as shown in the picture on page 3-2.

Their functions are clear from their labels:

- Clear line erases a line at the cursor.
- Clear display erases screen below the cursor.
- Insert line opens a new line at the cursor.
- Delete line deletes line and closes hole.
- Insert char(acter) turns "Insert" function off/on.
- Delete char(acter) removes a character at cursor.

Incidentally, except for "Insert char", all of these editing tools are repeaters.

IMPORTANT:

These keys only work partially if you're in EDITOR. They do what you'd expect them to while you're on screen, but you can't retain the changes that you make with the KEEP command.

Character Manipulation

Using the arrow keys, move the cursor to the left-hand edge of the screen, and down to one line from the bottom.

Press: `Delete char`

The action should have "eaten" whatever character was at the cursor's position, and the balance of the line should have shifted left by one character. Even if there was an empty space at the cursor, that space should have been deleted.

Hold down: **Delete char** (to make it repeat)

How about THAT! Character by character, it does away with the entire line, leaving an empty slot.

Refill the line by typing meaningless words, but DON'T press Return:

M E A N I N G L E S S W O R D S

Notice that you can make changes by typing right over the words, or characters, that you want to alter.

Turn the Insert char switch on by pressing: **Insert char**

```

      9      LIST  Presents The File
meaningless wordsakes Changes
                        ↑ (cursor location)
      12      //      Closes The ADD

```

LINE MODIFY	MODIFY ALL	BLOCK MODE	REMOTE MODE*		SMOOTH SCROLL	MEMORY LOCK	DISPLAY FUNCTNS	AUTO LF
					InsChar			

You can see it has been turned on by the indicator under the MEMORY LOCK (f6) window. You can now insert characters wherever the cursor is.

Type: **B E T W E E N**

You should see the word "between" appear, flanked by two spaces, pushing the rest of that line to the right, as shown.

```

  9      LIST  Presents The File
meaningless words between akes Changes

12      //      Closes The ADD

```

These are somewhat different than the commands in EDITOR, but you can accomplish about the same things with them.

Line Manipulation

Use the arrow keys to move the cursor into the left-hand edge of a "busy" part of the screen.

Press: **Insert line**

A one-line "hole" should have opened up at the cursor's location, pushing the other lines down to make room. It should be literally a new opening... with nothing in it.

Move the cursor up two lines.

Press: **Clear line**

A second one-line hole should have opened at the cursor's location. This one didn't shove the other lines around. It just blanked out the line it's on.

Move the cursor down a few lines.

Press: **Delete line**

This time, you should not only see a line disappear, but the lines below it should move up to close the slot.

Now, move the cursor to the third line above the end.

Press: **Clear display**

Everything below the cursor, which means the last two lines, should have disappeared.

To clear the entire screen with this key: move the cursor up to the upper left-hand corner, then press it.

Look at how many changes you've made to your Pat file. It should be almost unrecognizable by now. So, now it's time to try saving it with KEEP. Type:

K Return

and: **Y Return**

After the computer has done its job, check to see what happened by typing:

L A L L Return

Were you surprised? There should have been no change at all. File Pat should still look as it did before you started all this.

The editing keys COULD be useful to you later. They'll help you make changes to blocks of work that you'll give to the computer later, in a batch. They're also accepted by certain other non-EDITOR word processing systems.

You may want to do some final experimenting with the keys to be sure you know what they're all able to do.

Remember to turn off your Insert char toggle before going on.

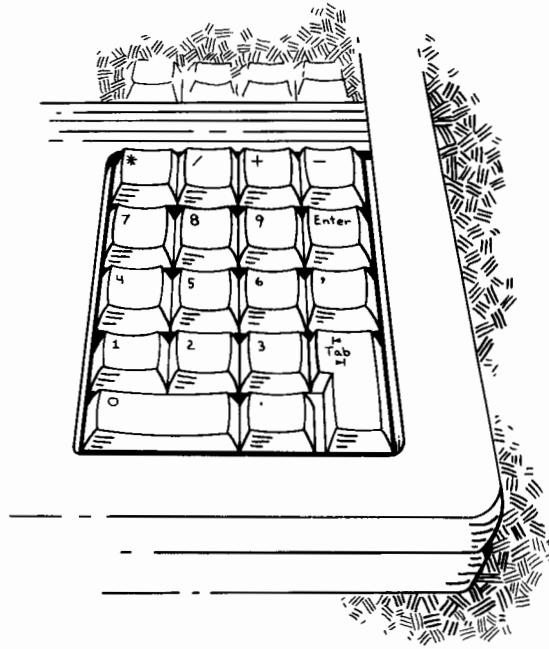
Press: **Insert char**

Check that "Ins Char" disappears from under the MEMORY LOCK (f6) message window.



Numeric Keys

At the far right end of the keyboard is a numeric key pad. It looks very much like the key pad on an office calculator, and can serve the same purposes.

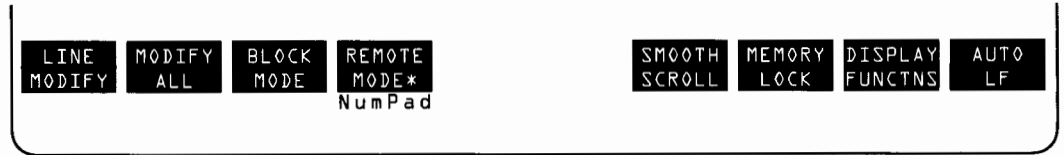


If you work with numbers... spreadsheets, tables, or simply numeric material... this is the easiest way to get them on the screen.

If you have any speed with a standard numeric pad, you may find yourself using this more than the whole rest of the keyboard.

HP 150 USERS! Your terminal will show you that it's in Numeric mode. Under the f4 message window (REMOTE MODE) you'll see: Num Pad, as shown below. If, instead, you see: Grph Pad, restore the Num Pad label. Press: **CTRL -**

↑ (numeric pad's minus)



The terminal lets you type with whatever mixture of keys is convenient. You can interleave:

- keyboard letters and symbols with pad numbers;
- keyboard numbers with pad numbers; and
- all three.

Try typing a line of mixed numbers using the numeric pad. Go ahead, do it.

Try another line. Mix letters with both sets of numbers.

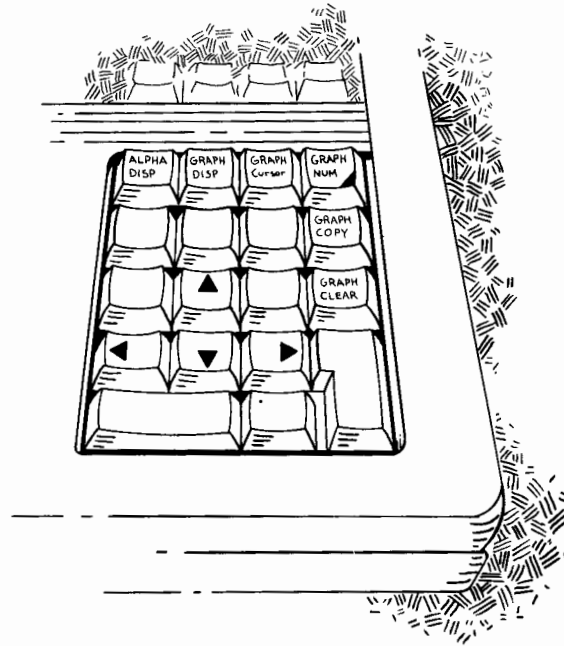
Can you think of any other combinations to try? Go ahead. Take time to practice and experiment.

Graphics Mode

If your terminal is an HP 2392A, you probably won't be able to use it in Graphics mode. If, on the other hand, it's an HP 150 Touchscreen, you MAY be able to use it in Graphics mode.

Change it (toggle it) over to Graphics mode by holding down: **CTRL** and then pressing: **-**

↑ (the numeric pad's minus key)



The indicator label should appear below the f4 screen, replacing Num Pad:

You're now in Graphics mode. The key pad no longer makes numbers on the screen. It now responds according to the diagram at the left.

If your computer has graphics capability these key functions will be true.

- ALPHA DISP toggles display off/on.
- GRAPH DISP toggles graphics off/on.
- GRAPH CURSOR toggles cursor off/on.
- GRAPH NUM toggles numeric/graphic.
- GRAPH COPY sends graphic data.
- GRAPH CLEAR clears out work area.
- FOUR ARROWS move cursor.
- CURSOR FAST speeds cursor movement.

Greater detail on the keys and their functions can be found in your terminal handbook, or in the special documents that describe using your particular graphics application.

Return your terminal to Numeric mode by holding down: **CTRL** **Shift**

and pressing: **-**

↑ *(the numeric pad's minus key)*

Confirm that you've returned to Numeric mode by observing a Num Pad label under the REMOTE MODE window. If this doesn't happen, repeat the steps above again, slowly.

The Other Keys

One way your keyboard differs from a typewriter's, is "the other keys". Some, like the Shifts, are similar, but have expanded uses. Others, like CTRL, are all new.

- Caps reverses Shift for alphabet keys
- CTRL sends a control signal to computer
- DEL/ESC extends the terminal's abilities
- Shift gives upper case characters, and extends some key functions
- Print/Enter sends work to the computer in chunks

If you'd like more information about any of these keys, it probably follows.

Caps Key

Are you still in EDITOR? If so, delete everything in the work area with:

D A L L Return

If you're not in EDITOR, return to it.

Type: **E D I T O R Return**

Start a new file by pressing: **A Return**

After your first numbered line appears, press: **Caps**

Be sure that the status note "CAPS" appears under message window f5

(SMOOTH SCROLL).

Now type "Some Meaningless Words". Use initial caps, and lowercase for the balance.

Type: **S O M E M E A N I N G L E S S W O R D S** Return

```
/a
  1  sOME mEANINGLESS wORDS
```

sEE? aLL THE LETTERS ARE REVERSED. tHE INITIAL CAPS ARE SMALL, AND THE LETTER THAT WERE LOWERCASE ARE NOW CAPS.

There are applications that call for typing all of the letters in upper case... or capitals. To avoid pressing "Shift" on every keystroke, the Caps key is available. While it's "on", every letter key will be capitalized UNLESS you press the Shift key... in which case, as you've already discovered, they appear in lowercase... or small. In other words, it's like the Shift key with its action reversed.

Experiment with it. Turn it on and off, type things, see what happens.

In your experimenting you may have already discovered that it doesn't work with all the keys on your keyboard. It doesn't work with:

- number keys
- punctuation keys
- symbol keys

3-28 The Keyboard

To check it out, make sure that Caps is "off". Then press the number keys:

1 2 3 4 5 Return

then, turn Caps back on: **Caps**

and retype the numbers: **1 2 3 4 5 Return**

then, hold down: **Shift**

and, again press the keys: **1 2 3 4 5 Return**

```
1    SOME mEANINGLESS wORDS
2    12345
3    12345
4    !@#%&
5    _
```

So, you were only able to reach the !@#\$% symbols through the Shift key!

Experiment more. Discover for yourself which keys work normally when CAPS is "on". With practice, you'll begin to use the key without even thinking about it.

The CTRL Key

This key sends special control messages to the computer. It can be used alone, but mostly it's used with other keys, sometimes more than one.

You've already used it once or twice if you've followed every previous step. Do you recall this combination: Reset/Break plus CTRL plus Shift (to do a "hard reset")? Or the combination: CTRL plus - (to switch between "Num Pad" and "Grph Pad")? Well, you can also:

- > ring the terminal's bell with: **CTRL** plus **G**;
- > toggle the HP 150's "touchscreen-edness" off and on with: **CTRL** plus **Shift** plus **Menu**;
- > stop "scrolling" with: **CTRL** plus **S**; and
(described better later)
- > start it up again with: **CTRL** plus **Q**.
(also described better later)

Other uses are described in your terminal's handbook.

Incidentally, this key is indicated in several different ways in the handbooks you'll be reading later:

- CTRL as in: CTRL G
- Control as in: Control G
- ^ as in: ^G

Since this IS a computer-type key, it's nice to have an idea what's going to happen before you use it.

The DEL/ESC "Escape" Key

Like CTRL, this also sends messages to the computer, or allows certain actions.

Do you remember, back in the Word Processing section of this book, putting a lockword (your nickname) on your "Jones" file? Later, the computer asked you for that lockword and, when you typed it, it was invisible?

Here's how to make things on your screen invisible:

- DEL/ESC plus ; (semicolon) turns visibility off
- DEL/ESC plus : (colon) restores it

If you're still in EDITOR, leave: **E Return**

Hold down: **DEL/ESC** then press: **;** **Return**

Notice that the semicolon did NOT appear on the screen, but the computer reacted to it as an "unknown command".

Now, type: **E D I T O R Return**

```
:
  ^
UNKNOWN COMMAND NAME. (CIERR 975)
:
HP32201A.#.## EDIT/3000 (day, date & time)
(C) HEWLETT-PACKARD CO. 1984
/
```

See... the command EDITOR also was invisible. Putting two and two together:

- the things YOU type are invisible
- the computer's answers are visible

With that in mind, you might start doing a little experimenting.

But, first; hold down: **Return**

See the slash prompts work their way to the right-hand side of the screen: /

/

/

/

/ until they disappear, and pick up again on the left-hand side.

Release: **Return**

You can type commands to EDITOR when the slash prompt is anywhere on the screen.

Go ahead... experiment!

When you're ready to make the commands visible again,

remember to hold down: **DEL/ESC** and press: **:**

There are other uses for DEL/ESC, which is also called the "Escape Key", to be found in the terminal's handbook.

The Shift Keys

You're familiar with Shift keys from your experience with typewriters. They change your typing from lowercase to uppercase.

They do that here, too. In addition, they offer a number of control functions, most of which you've already used:

- Shift plus Caps key reverses what is typed
- Shift plus Reset/Break + CTRL is hard reset
- Shift plus ▲ or ▼ (arrows) moves work window
- Shift plus ▷ (hollow arrow) moves cursor to end

The Print/Enter Key

Earlier in this chapter you tested Block mode. You turned BLOCK MODE on by pressing function key f3. Then you discovered that you couldn't get the computer to accept the line you'd typed: Meaningless words. You pressed Return, but nothing happened.

If you had pressed Print/Enter, the computer would have accepted, because you would be "Entering" a "Block" of information... whether it's a line or a whole page.

If you'd like to check this out, turn back to the part titled "Testing Block Mode". Go through the steps. Finally,

type: **M E A N I N G L E S S W O R D S Print/Enter**

You see what happens. The program accepts the line, this time, and indicates it by presenting the next line, ready for use.

When you finish experimenting, remember to return your

terminal to Line mode by pressing: **f3**

Looking Back

By now you have a much better understanding of your keyboard. You know that it's some different from your typewriter's, but not TOO different, really. So, if you'd like to check some of this new knowledge...

1. Can you KEEP the changes made with the Editing keys? (Yes/No)
2. How do you switch the message windows off and on?
3. Which keys do you press to make a "hard reset"?
4. How do you turn the message windows, as a group, off?
5. How do you scroll your work area up on your screen?
6. How do you make the commands you type invisible?

ANSWERS

1. Not in EDITOR.
2. With the eight function keys.
3. **Reset/Break** **CTRL** **Shift**
4. Press: **Menu**
5. Hold down both: **Shift** and: **▲** (up arrow)
6. Press: **DEL/ESC** **¶**

About Computer Files.....	4-1
Naming A File.....	4-3
Creating A File.....	4-4
Checking File Contents.....	4-6
Copying A File (FCOPY).....	4-7
Checking The Copy.....	4-10
Renaming A File.....	4-10
Removing A File.....	4-12
Printing a File.....	4-14
The Printing Request.....	4-15
The Printing Process.....	4-16
Help With MPE Commands.....	4-19
Controlling Scroll.....	4-20
Continuing HELP.....	4-22
Using Tape Storage.....	4-24
The STORE Procedure.....	4-25
The RESTORE Procedure.....	4-26
Shuffling File Data.....	4-27
Random Data File.....	4-28
Sorting By Name.....	4-30
Sorting By Date.....	4-33
Sorting Applications.....	4-35
Looking Back.....	4-37

Using Files

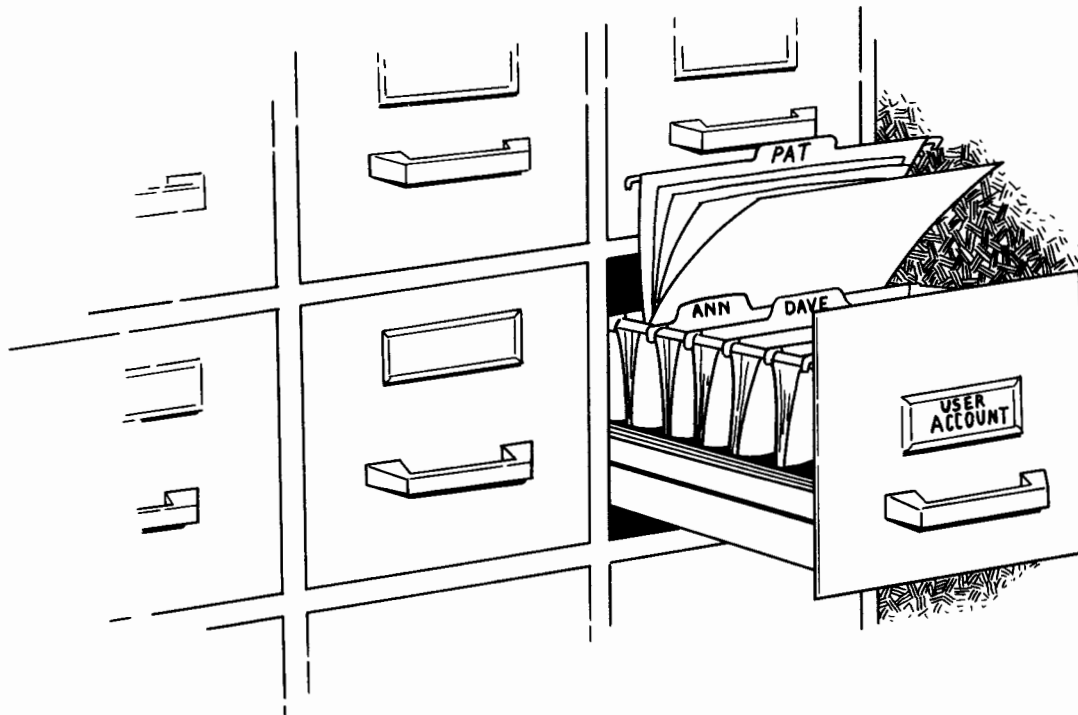
About Computer Files

Computer files are a lot like your familiar office files.

You know: a filing cabinet is a place where you can store lots of papers and other things. So, a computer has a place where you can store lots of pages of information, and also other things. It's electronic, not paper, but it's the same.

You know: most filing cabinets have their files stored in a logical, organized way. So do computers.

You know: a filing cabinet uses tabs and labels to identify its files. So, computer files are named and indexed for ease in relocating them.



4-2 Using Files

If you're still in EDITOR's ADD mode, after finishing the last section, leave it by typing: `// Return`

then: `E Return`

and: `Y Return`

If, on the other hand, you've just logged on again, you should have the computer's colon prompt on your screen.

Type: `L I S T F Return`

Remember this? It's the list of files in the USER.ACCOUNT that you have permission to use, because you logged on with that identity. You should be able to spot your Pat and Jones files (or whatever names you used).

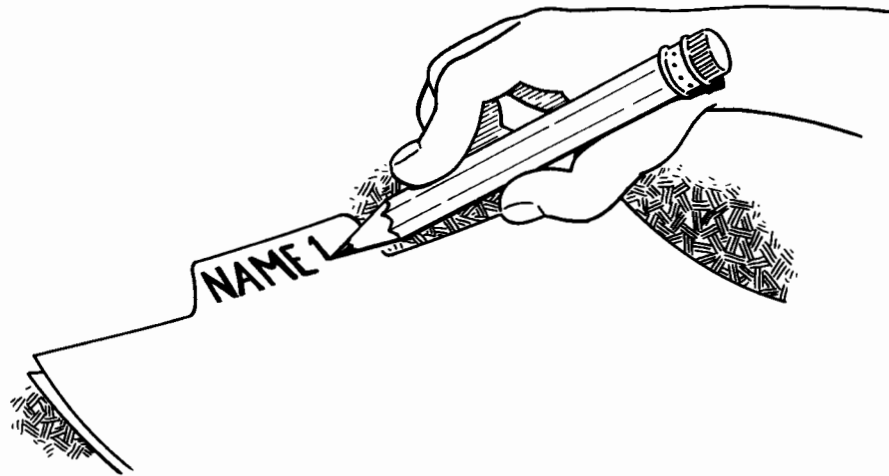
Notice again: the file names are arranged alphabetically, so it's easier to find the one you're looking for... just like file folders in a filing cabinet drawer. The names are the computer's version of the tabs and labels in that drawer.

IMPORTANT:

If the list of file names has become very long, you may have to scroll back to find them.

Naming A File

You'll probably be naming a lot of files, if you spend much time working with computers. So, it's worthwhile to learn the few rules involved.



A good file name:

- must start with a letter
- will have from 1 to 8 letters and/or numbers
- avoids symbols; they often have "computer" meanings
- avoids punctuation marks, for the same reason
- is new, not already in your LISTF list
- may helpfully describe the file's contents
- is short enough to type quickly, if it's used often

4-4 Using Files

Here are a handful of typical file names, to give you an idea of how it works.

A1	A2	A3	BUDGET87	BUDGET88	CLASS
COPYUDC	DISC	FEDTAX85	FIG1	FIG2	GRADES
MEMO1	MEMO2	MEMOA	MEMOB	OUTLINE	REPORT
REVIEW	SALESJAN	SALESFEB	SPEECH1	TAX85	ZOT71MM

Names like A3, Fig2 and Memo1 are quick to type, but you must remember what's in them... easy with two or three, but tough with a hundred.

With all that in mind, and without knowing what's going into the file, pick a name. Don't spend a long time on the project, but try to pick a good one. You'll be using it right away to name a newly created file.

Scan the LISTF list to be sure it's really new.

For convenience, it'll be referred to in this section as: "Name1".

Creating A File

You've created files by using the EDITOR's KEEP command. There's also another way: using the MPE command BUILD. This is followed by the file name you've chosen, and some specifications that tell the computer what kind of file you want: ;REC=-80,1,F,ASCII. These have to be typed very accurately.

NOTE: Computer people will recognize the specifications as REC(ord size=-80 bytes) F(ixed-length records), 1 (initially allocated extents), ASCII(vs BINARY) for text vs numeric material. (You don't really have to understand all this.)

So, to create the file, type very carefully:

```
B U I L D   N A M E 1 ; R E C = - 8 0 , 1 , F , A S C I I   R e t u r n
```

↑ (substituting your own choice, of course)

IMPORTANT:

If there was a mistake in the process, the computer will tell you. It will try to show you what was incorrect or missing. Its messages may not be too clear, but give it your best shot. If all else fails, remember to experiment.

To see if you were successful, type: `LISTF Return`

It should be in its proper place, alphabetically.

Pick another new file name. Use the same rules to pick it. Check the LISTF list to be sure it's really new. For the purposes of this book, it'll be called "Name2".

Then, being very accurate, use the BUILD command again to actually create the file.

Type: `BUILD NAME2;REC=-B D,1,F,A S C I I Return`

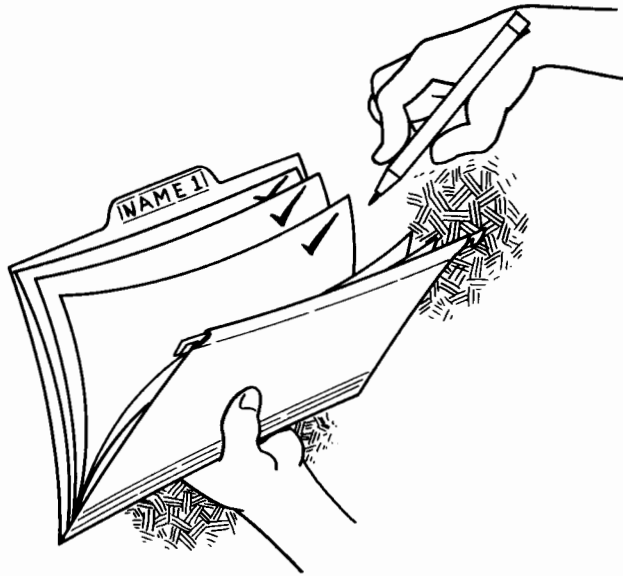
Check your success again with: `LISTF Return`

Incidentally, just for your convenience, it might be helpful for you to start a little list of the file names you're actually using, versus the names given to them in this book. It might look something like this:

```
Pat .....Chris
Jones .....Aruna
Name1 .....Charts
Name2 .....Report
```

Checking File Contents

Somewhere in that list are also the names of your Pat and Jones files (or whatever you called them). You remember what's in each of them? In Pat, a dozen EDITOR commands; in Jones, a half-dozen other lines, and a command.



What're in Name1 and Name2? Can you guess?

You can find out by going into EDITOR, then asking to see the TEXT in Name1.

Do it. Type: `EDITOR` Return

then: `T NAME1` Return

```
/t name1
*23*FAILURE TO OPEN TEXT FILE (0)
END OF FILE (FSERR 0)
/
```

This means that the computer can't open the file because there's nothing in it. It's empty... or, maybe it would be more accurate to describe it as "not yet started".

IMPORTANT:

All files that you create with BUILD are empty until you put something in them. Much like empty file folders, with proper labels on their tabs. So, not only is Name1 empty, so is Name2.

Leave EDITOR by typing: **E** **Return**

**Copying A File
(FCOPY)**

In Word Processing you learned how to copy lines, groups of lines, and even a whole work area inside a file. Here, you'll learn to copy an entire file, then place it into another "file folder".



You're going to make a copy of whatever you've got in your Jones file, then put it into the file named Name2.

Do you still have a Jones file... or a file with your last name on it, and using your nickname as a lockword? If not, take a few minutes to recreate one. You'll need it in this exercise.

Next, type: `FCOPY` Return

IMPORTANT:

FCOPY is another like **EDITOR**: an entire "command family", with a number of subcommands, and even its own prompt. As **MPE**'s prompt is a ":", and **EDITOR**'s is a "/", **FCOPY**'s is a ">".

```
>:fcopy
HP32212A.### FILE COPIER (C) HEWLETT-PACKARD CO. 1983
>
```

To start the copying process, you'll use two of **FCOPY**'s subcommands, **FROM**, and **TO**, separated by a semicolon. Just type:

`FROM=JONES;TO=NAME2` Return

then supply the lockword: `ARKIE` Return
 ↑ (your nickname)

IMPORTANT:

Remember, you won't be able to see the lockword on your screen. It'll be invisible, for security reasons.

Then the computer MAY warn you that it's going to be a tight fit, putting Jones into Name2. Reassure it that you aren't worried. Answer yes, with:

Y **Return**

```
>from=jones;to=name2
LOCKWORD: JONES.PUB.ACCOUNT?

EOF FOUND IN FROMFILE AFTER RECORD 5

6 RECORDS PROCESSED *** 0 ERRORS
>
```

IMPORTANT:

Your interchange with your computer may not look exactly like that. You may get warnings, and its reports to you about what's happening may vary. You can cope by telling it, as often as necessary, to carry on.

Finally, EXIT from FCOPY by typing: **E** **Return**

The colon prompt should return to the screen.

Checking The Copy

Reenter EDITOR, call for the TEXT of Name2 file, and ask for a listing, by typing: `EDITOR Return`
 then: `T NAME2 Return`
 and: `L ALL Return`

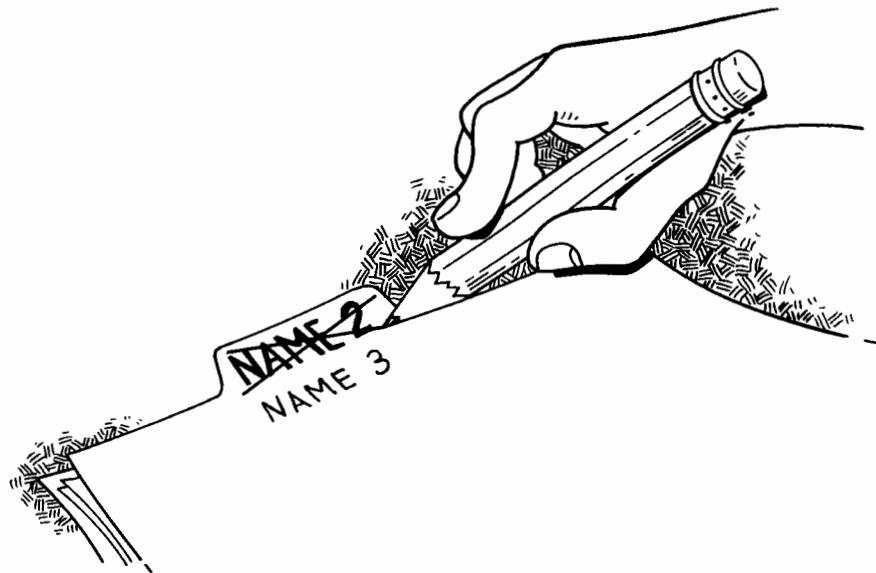
Sure enough, it should show up on your screen. The file that was empty is now full! Some interesting facts are:

- the information in the Name2 file is just a copy; the original information is still in Jones file;
- copying didn't transfer the lockword; the copy is now open to prying eyes; and
- you can also add a file onto the end of an already-existing file.

Leave EDITOR again with: `E Return`

Renaming A File

The name you've given this file may no longer describe it. It's really part full, so why don't you think of a new name that fits it better. For convenience, call it Name3.



The proper command is RENAME, and the way to use it is to type the present name, followed by a comma, and then the new name.

IMPORTANT:

You can only rename a file if YOU were the one who created it.

You can also add a new lockword, during the renaming process. It's not necessary, but if this is going to be private information, you may want to safeguard it.

For convenience, use your nickname again. This book used Arkie to represent your nickname.

Type: `RENAME NAME2, NAME3 / ARKIE Return`

IMPORTANT:

If you DON'T want to put a lockword in place, just leave out the "/ARKIE".

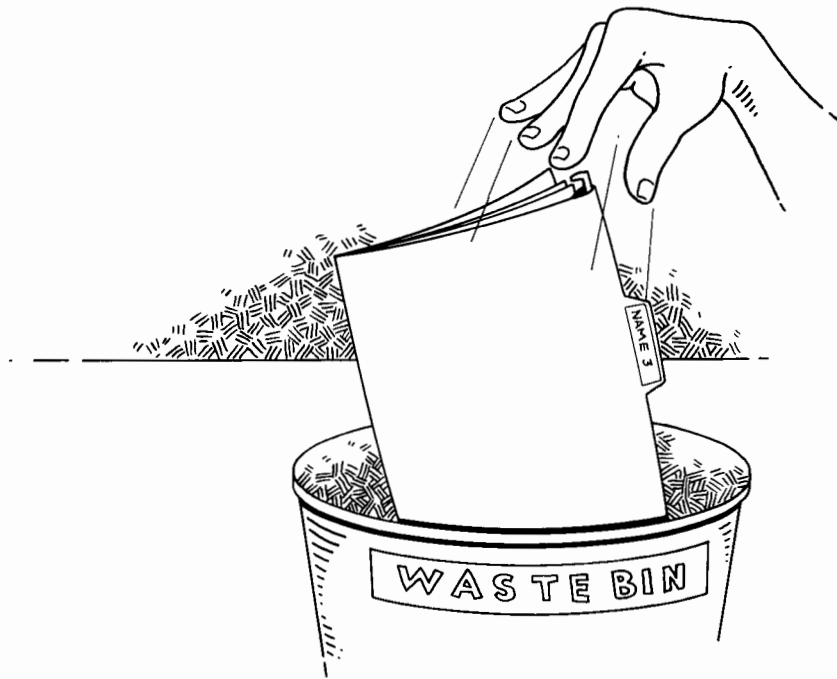
Then, to check what's happened, type: `LISTF Return`

There again, down in alphabetical sequence, you'll see that Name2's gone, and there's a new file, Name3.

If you want to assure yourself that everything went as it should have, take time to go into EDITOR, call for TEXT Name3, type the lockword and LIST ALL. Do you remember how? Go ahead... do it!

Removing A File

There may come a time when you want to get rid of a file. It may be out of date, no longer useful, and taking up needed disc storage space.



The proper command is PURGE. The way to use it is to just type PURGE, followed by the name of the file you want to remove.

If you've finished with file Name3, go ahead, practice on it. Be sure you have no further use for the file, because using the PURGE command will completely eliminate it.

Type: **PURGE NAME3** Return

When asked for the lockword, type: **ARKIE** Return

The return of the colon prompt is the signal that the file has been purged. Check it in the usual way.

Type: **LISTF** Return

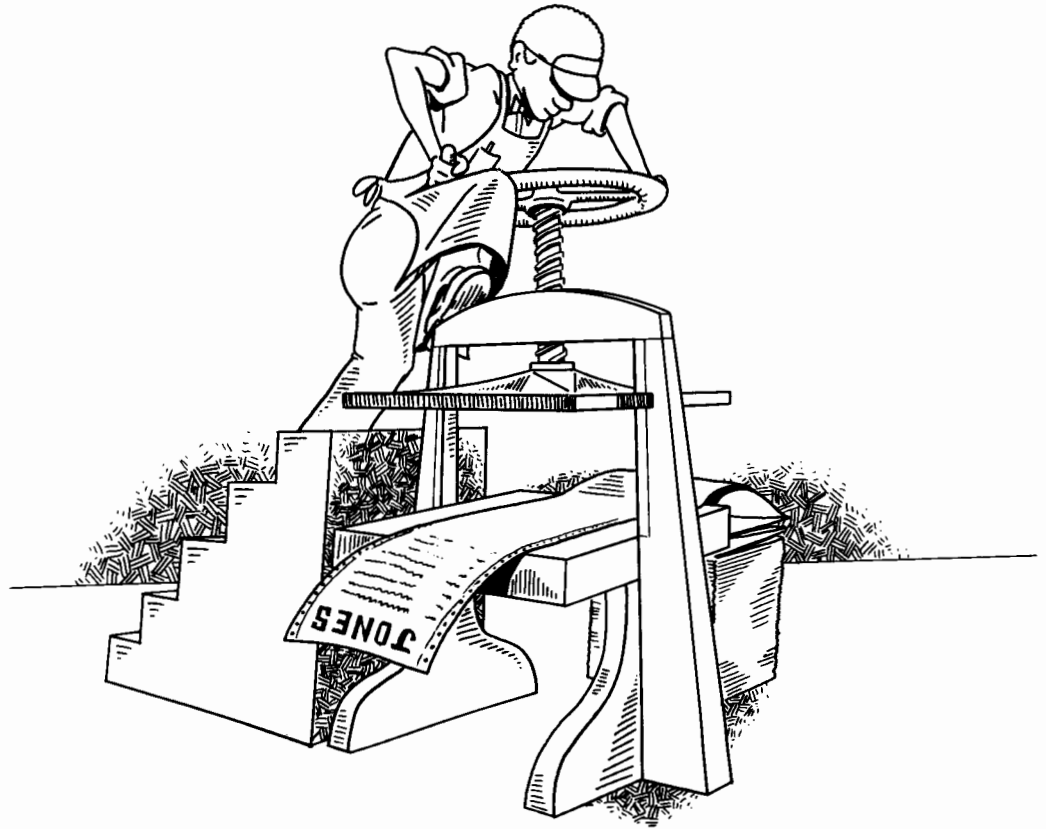
IMPORTANT:

It's clear that with PURGE you can remove any file in your ACCOUNT that appears when you type LISTF, and whose lockword you know. Think, before you use it; it can destroy other people's work.

You can only use the PURGE command on one file at a time. Other commands let you do two or more things with a single use of the command. This is not true of PURGE.

Printing a File

You've now typed some material, corrected it, and stored it away in what amounts to an electronic filing cabinet. The final step is to put it on paper, so it can be used by other people, or copied and distributed.



Computer people call this “printing a file”, and they speak of the end product as a “printout”, or “hard copy”.

The actual printing, on your computer system, is probably done by a thing called a "line printer". It's almost certainly not a part of your terminal, though it may be nearby. You can often identify a line printer by the paper coming out. Look around. Can you see one? If not, you may have to ask your Operator where it's located.

The Printing Request

This has a few more steps than typing a simple command. It starts with FILE, a new command. FILE is used when you already have a file, but want to define it more fully, or do something with it. In this case you're going to send it out to a DEV(ice) that you specify as an LP (Line Printer). Then, you go into EDITOR, call up your file, and then ask that it be LISTed "offline".

You should still have a Jones file, so try experimenting with that. If you've purged it, use your Pat file, or anything else you have available.

You should have a colon prompt on your screen.

Type: `FILE OUT; DEV=LP` Return

next: `EDITOR *OUT` Return

then: `T JONES` Return

and: `ARKIE` Return (or your nickname lockword)

and, finally: `L ALL, OFFLINE` Return

This should appear on your screen.

```
:file out; dev=lp
:editor *out
HP32201A.#.## EDIT/3000 (day, date & time)
(C) HEWLETT-PACKARD CO. 1984
/t jones
LOCKWORD: JONES.PUB.ACCOUNT?

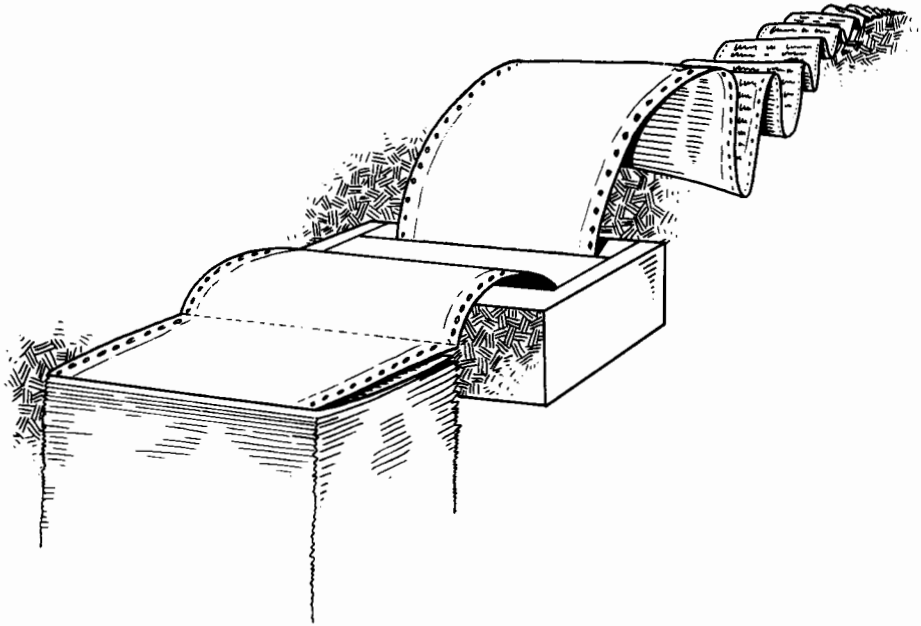
/l all, offline
*** OFF LINE LISTING BEGUN. ***
/_
```

IMPORTANT:

If, instead, there was an error message, check to be sure you typed all the letters, spaces, asterisk, comma and equals sign accurately. The error message will also try to help you.

The Printing Process

If all went well you got the message: ***** OFF LINE LISTING BEGUN. *****, and you got the picture that the printer has started your job.



Not necessarily true.

Your job doesn't go directly to the printer, but rather to the computer. The computer arranges all of the printer's incoming jobs into a queue, by priority.

If it had no other jobs, the printer could even have done yours already. If it had lots of other jobs ahead of yours, you may have to wait awhile.

If you'd like to check on it, leave EDITOR, and use the status command SHOWOUT SP. (Computer people may recognize SP as a "SPooling" parameter.) It lets you ask where your job stands in the computer's spooling sequence... or waiting line.

Type: **E** **Return**

and: **S****H****O****W****O****U****T** **S****P** **Return**

```
/e
END OF SUBSYSTEM
:showout sp

DEV/CL  DFID  JOBNUM  FNAME  STATE  FRM  SPACE  RANK  PRI  #C
#        ##### #S###  OUT    ACTIVE      ##      #   #
(and perhaps more lines of this table)

OUTFENCE = # (a number from 1 to 14)
:—
```

If your screen shows the table above, it means your job is still waiting to be done. Furthermore, it shows how many jobs are ahead of yours in the queue, what kind of priority each has, and something about each of them. If the table is long, with lots of high priority tasks, you may have a bit of a wait.

The printer IS fast, however.

If, on the other hand, the screen shows:

```
/e  
END OF SUBSYSTEM  
:showout sp  
NO SUCH FILE(S)  
OUTFENCE = # (a number from 1 to 14)  
:—
```

it means that there are none of your files waiting in the queue. That means your job is done, and waiting for you to pick up.

Go get it.

If your Operator takes care of the printer, your job may be waiting for you already, separated and folded. If not, you'll have to poke around in the paper that has come out of the printer, find your job, and tear it out of the accordian-folds, or whatever your printer uses.

Be careful with other people's work.

IMPORTANT:

If you can't find your job, check whether there's a second printer attached to your computer... or a third.

Help With MPE Commands

In EDITOR, there was a command called XPLAIN. It offered some help in understanding the other EDITOR commands. It also gave you clues into some of the broader uses of those commands. There's also one for MPE.

MPE, M(ulti)-P(rogramming) E(xecutive), is your computer's operating system. It's a group of programs and "software packages" that lets your computer do what it does. Its helpful command is HELP.



In its simplest form, HELP is typed, followed by the MPE command you're curious about.

Take the HELP command, itself. You can find out about it by typing:

```
HELP HELP Return
```

```
:help help
:HELP
```

Accesses the Help subsystem. *(what HELP does)*

SYNTAX *(how to use HELP)*

```
:HELP [HELP          ] (what you typed, above)
      [tablecontents  ] (8 definitions; see **)
      [command[,keyword] ] (command + keyword; see *)
      [ALL            ] (many pages, gen. info.)
      [EXIT           ] (to leave HELP)
```

KEYWORDS: PARMS, OPERATION, EXAMPLE *(see *, above)*
:_ *(prompt and cursor)*

- * Example: HELP BUILD, OPERATION results in a description of how the BUILD command operates.
- ** Example: HELP SESSIONS gives information on running sessions. Other subjects are: OPERATOR, PROGRAMS, SPOOLER, MANAGE, JOBS, FILES and UTILITY.

As a sample of how this all works, see how much there is to learn just about the HELP command. Use the ALL subcommand, by typing:

```
HELP HELP ALL Return
```

Controlling Scroll

Probably got away from you, didn't it?

If you typed the right things correctly, you should have seen a few pages of text and columns go scrolling upward on your screen. And, in this particular mode on your terminal, you can probably only scroll backward for about a page. Therefore, you lost the first page or two.

IMPORTANT:

It's all right to have this happen. It CAN be controlled.

The way to turn the scrolling off and on is by using your CTRL key:

- CTRL + S halts the scrolling
- CTRL + Q starts it up again

Getting a scrolling action started usually means typing at least a command and another word, and then pressing Return. After Return, hold CTRL down, then alternate between the letters S and Q.

Let's try it.

Again, type: `HELPHELPALL` Return

Once the scrolling starts, hold down: `CTRL`
then press: `S`

Everything should come to a halt. If you check your screen carefully, you'll see that the cursor marks the actual stopping spot. It could even have halted in the middle of a word or a sentence.

While still holding down: `CTRL`
press: `Q`

The action picks up again, and the presentation continues until the file ends, or you press S again.

IMPORTANT:

This works with ALL long files and work projects, not just those in the HELP subsystem.

Experiment with this. Learn to stop the scrolling action, and then restart it right where you want to. This skill will be very helpful in your work.

Continuing HELP

Now that you can control the scroll, go back and try reading the contents of the file that shows up when you type HELP ALL. There's some good, useful information.



It's a little "computerish", because it was originally meant for a different group of users, but you'll get the idea.

Notice the large number of commands that are mentioned. You'll be using at least some of them, depending on the kind of work you'll do.

Experiment. Try HELP on a command you've already used.

Type: `HELP SHOWME` Return

It should give you this very straightforward definition: "Reports job/session status", plus SYNTAX and KEYWORDS. Simple, right?

Experiment some more. Use HELP to check other commands you've already met: HELLO, BYE, SHOWTIME, REDO, EDITOR (although HELP knows nothing about EDITOR's commands, remember), LISTF, BUILD, SHOWOUT, RENAME, PURGE, FILE and FCOPY.

Notice that, according to the HELP information, some of the commands are more complex than you thought.

Experiment even further. Use HELP to look into some of the other commands from the HELP ALL lists that you think you might use, eventually.

IMPORTANT:

It isn't necessary for you to understand the HELP information. You should be getting enough guidance from this book to serve most of your needs.

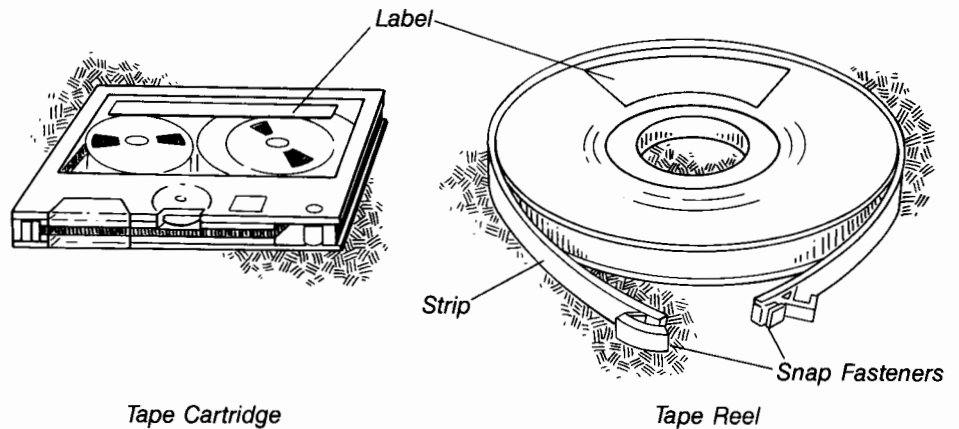


Using Tape Storage

There are times when you want to move a file from the computer's main, or disc, memory. Examples: to take word processing to a printer, monthly totals to the accountant, or files to another computer, or just to store big files out of the way more efficiently ("off-line").

This can sometimes be done over the telephone lines, by modems (a contraction of MODulator, DEModulator). Mostly, though, it's done by storing them on tape.

Cartridges and reels are the two most likely methods of tape storage in your computer system. Ask which your Operator prefers to use. Also ask if there's a special one kept aside for this step. Mention that you're ready to do it.



You might try taking the reel or the cartridge out of its protective container, just to see how it works.

- The reel may have a protective collar wrapped around it, with a snap fastener that isn't too hard to figure out. See if you can get it open. Examine the tape. Snap it closed again. Note the name on the label, if any.
- The cartridge may be in a rectangular housing of plastic. The housing's front edge pulls away from the box, forming a snap lid. Pull it open. Examine the cartridge. Close it.

Don't try to insert a tape cartridge or mount a reel on the computer by yourself. The Operator's the one who should do that for you.

You'll probably have to give the Operator the proper tape. Most of them don't keep libraries of work storage tapes. Keeping these is usually done by the accounts, the groups, and even the individual users, like you.

The STORE Procedure

When you give the tape, ask what it's "name" is, so you'll know how to refer to it later. In this discussion let's call it tape "ABC", for simplicity's sake.

To store your Jones file on the reel or cartridge tape named ABC, that you just gave to the Operator, AND to leave a record copy behind in the computer, you'll use the FILE and STORE commands in a two-step process similar to the sequence in "Printing A File".

Type: `FILE ABC;DEV=TAPE` Return

and: `STORE JONES;*ABC;SHOW` Return

IMPORTANT:

If you DON'T want to leave a copy of the file behind in the computer, for whatever reasons, insert: `:PURGE` in between SHOW and Return.

It should look like:

```
:file abc;dev=tape
:store ones;*abc;show
STORE/RESTORE, VERSION 2 (C) 1981 HEWLETT-PACKARD CO.
(the day, date & time)
```

Be ready to type in the lockword when it's asked for, and the responses to any questions.

You may get messages like these:

```

DEVICE UNAVAILABLE (FSERR 55)
FAILED TO OPEN FILE ABC (S/R 52)
UNABLE TO OPEN TAPE FILE (S/R 6093)
STORE ABORTED BECAUSE OF ERROR

```

Most of them are obvious, and are usually problems at the computer's controls. A suitable tape drive may not be available right away. The wrong tape was put in. In most cases, the Operator can either explain, or fix things.

If the process was succesful, you'll get the cartridge or reel back with your Jones file stored in it, and see something like this on the screen:

```

FILENAME.GROUP .ACCOUNT LDN ADDRESS REEL SECTORS CODE
JONES .GROUP .ACCOUNT 1%00257636 1 3
FILES STORED: 1

```

The RESTORE Procedure

When you want to get back into a file that you'd stored on tape, you just reverse the STORE procedure.

Start by giving the Operator your tape named ABC, and mention that you'll be using it soon. Then, a two-step process, like printing a file.

Type: **FILE ABC;DEV=TAPE** Return

and: **RESTORE *ABC;JONES;KEEP;SHOW** Return

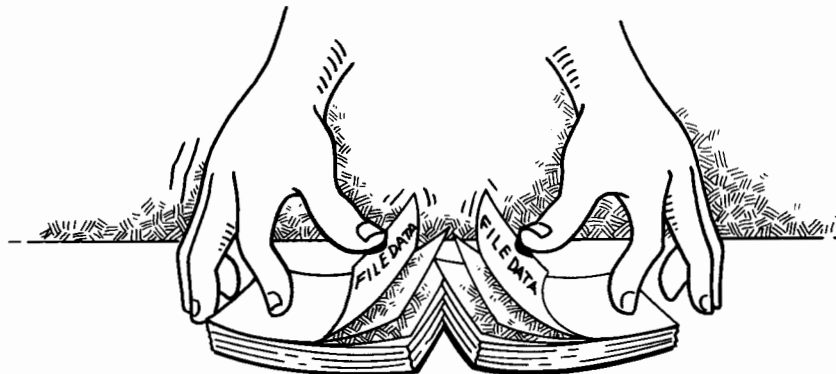
There may be a pause while the Operator mounts the tape. After the computer has scanned the tape, it will report to you on its success or failure in finding the file named Jones.

The SHOW request, at the end, asks that it list for you all the files it has restored... which means Jones.

Shuffling File Data

There's a very useful thing you can do with the information in your files.

You can shuffle it around to fit the often changing needs of the moment. Or, you can produce sets that highlight the most important facts. Or, you can assemble your facts in a random manner, and organize them later, when you have more time or a better idea what your plans or needs are.



To try this out, you'll need a couple of empty files for later use, and a make-believe batch of random information to work with.

Start by creating a couple of fresh, empty files. Again, pick names that aren't yet in use in your LISTF list. In this book they'll be called Listdate and Listalfa.

Type:

```
BUILD LISTDATE;REC=-8D,1,F,ASCII Return
```

and:

```
BUILD LISTALFA;REC=-8D,1,F,ASCII Return
```

IMPORTANT:

If you have any trouble in building the files, check to be sure you included the semicolon, equals sign, and 3 commas, but only one space.

Type: `L I S T F Return`

Be absolutely sure you can find them in the list.

Random Data File

Next, build your "random" data file. You'll use EDITOR to do this, so call it up, then ask for the ADD function.

Type: `E D I T O R Return`

then: `A Return`

As soon as the first line number appears, start the file

with: `J . S W I F T 1 6 6 7 W R T R Return`
 and: `F . G O Y A 1 7 4 6 P N T R Return`
 and: `J . A D A M S 1 7 3 5 P R E S Return`
 and: `S . L E W I S 1 8 8 5 W R T R Return`
 and: `G . R U T H 1 8 9 5 H T T R Return`
 and: `L . B E R I A 1 8 9 9 M K V D Return`
 and: `T . G R A Y 1 7 1 6 P O E T Return`
 and: `F . L O R C A 1 8 9 8 P O E T Return`
 and: `W . C L A R K 1 7 7 0 E X P L Return`
 and: `A . H I T L E R 1 8 8 9 P L T N Return`
 and: `C . E N - L A I 1 8 9 8 P L T N Return`
 and: `E . Z O L A 1 8 4 0 W R T R Return`
 and: `N . T E S L A 1 8 5 6 E N G R Return`
 and: `G . C A R V E R 1 8 6 4 B O T N Return`

```

and: J . D O N N E 1 5 7 2 C L G Y Return
and: A . F O K K E R 1 8 9 0 E N G R Return
and: E . K E A N 1 7 8 9 A C T R Return
and: J . B A C H 1 6 8 5 C O M P Return
and: C . N E R O 3 7 E M P R Return
and: G . S A N D 1 8 0 4 W R T R Return
and: J . B O O T H 1 8 3 8 C R T C Return
and: S . P E P Y S 1 6 3 3 W R T R Return
and: C . M O N E T 1 8 8 8 P N T R Return
and: M . C U R I E 1 8 6 7 C H E M Return
and: J . C R U Z 1 6 5 1 W R T R Return

```

Select still another file name for this list. Because the list will be used as raw data, this book will use the file name `Listraw` for it, but it would be better if you used your own equivalent name.

Because of its purpose, it's important to store it away WITHOUT line numbers.

So, finish with: `// Return`

Keep it with: `k LISTRAW, UNNUMBERED Return`

Check it by typing: `L ALL Return`

Finally, leave EDITOR with: `E Return`

```

/k listraw,unnumbered
/l all
  1      J. Swift  1667  wrtr
  2      F. Goya  1746  pntr
  3      J. Adams 1735  pres
  4      S. Lewis 1885  wrtr
  5      G. Ruth  1895  httr

```

IMPORTANT:

The columns **MUST** be vertically aligned: each initial, each name's first letter, etc. Take time to edit suitably.

Notice that the line numbers reappeared when you listed them. This is **ONLY** for convenience in editing. They are not stored with the file.

Sorting By Name

The goal, here, is to shuffle the twenty-five names so they're arranged alphabetically, by last name. To do this, you have to specify the last names, what you want done with them, and where to put the results.

The main tool is SORT-MERGE/3000... a command "family" like EDITOR and FCOPY. Like FCOPY, it has a prompt that looks like this: >. It's activated by the RUN command.

Specifying that you want to sort by last names is done by describing the names' position. In this case, they start at the 4th position; and the longest name is 6 positions long. See the POSITIONS figure below.

POSITIONS

```

          1          2
1234567890123456789012345

```

```

J. Swift 1667 wrtr
  ↑ The last names start at the 4th position,
  ... and are a maximum of 6 characters long.

```

Start by typing: **R U N S O R T . P U B . S Y S** Return

Wait for the SORT program to identify itself and provide its > (chevron) prompt.

Answer with: **I N P U T L I S T R A W** Return
 and: **O U T P U T L I S T A L F A** Return
 then: **K E Y 4 , 6** Return
 finishing with: **E N D** Return

```
:run sort.pub.sys

HP32214C.##.## SORT/3000 (day, date & time)
(C) HEWLETT-PACKARD (C. 1983

>input listraw      (file with the source names)
>output listalfa   (file that gets the sorted names)
>key 4, 6          (4th position; 6 characters long)
>end               (leave SORT)

                        STATISTICS

                    (followed by 9 lines of uninteresting information)
```

To see if it worked, re-enter EDITOR, ask for TEXT Listalfa, and have it LIST ALL.

Type: **E D I T O R** Return
 then: **T L I S T A L F A** Return

It will take a moment, then advise you that the file is unnumbered. After it provides another slash prompt, you can finish with: **L A L L** Return

The screen listing should show:

```

:editor
(I.D. & date)
/t listalfa
FILE UNNUMBERED
/1 all
  1      J. Adams  1735  pres
  2      J. Bach   1685  comp
  3      L. Beria  1899  mkvd
  4      J. Booth  1838  crtc
  5      G. Carver 1864  botn

```

IMPORTANT:

If it didn't work, review all the steps. Check for typing errors. Try it again. If trouble persists, see if you can locate a SORT-MERGE/3000 reference manual. Check the procedure in Section I. Try to figure out what happened.

Notice that line numbers are still provided for editing convenience, but they've been resequenced. J. Adams is no longer line 3, as in the Listraw file, but is now appropriately numbered line 1.

Leave EDITOR by typing: **E** **Return**

Sorting By Date

The next project is to shuffle them by birth dates. Start by making note of the dates' location. Do you remember how to use the POSITIONS figure?

POSITIONS

```

          1          2
1234567890123456789012345

```

```

J. Swift  1667  wrtr

```

```

↑

```

*The dates start at the 11th position,
... and are up to 4 characters long.*

Again, you'll take your source material from Listraw file, and put the sorted list into Listdate file.

Type: `R U N S O R T . P U B . S Y S` Return

After SORT identifies itself, and provides a > (chevron) prompt, type

`I N P U T L I S T R A W` Return

again: `O U T P U T L I S T D A T E` Return

then: `K E Y 1 1 , 4` Return

and leave with: `E N D` Return

After the processing and the STATISTICS, the program ends. Then you re-enter EDITOR, call up the Listdate files, and ask it to list all. Type:

`E D I T O R` Return

then: `T L I S T D A T E` Return

and: `L A L L` Return

On your screen, all those steps should look like this (a composite for convenience sake). The only variations might come from the addition of lockwords, or such.

```
:run sort.pub.sys
(I.D. & date)
>input listraw      (file with the source names)
>output listdate    (file that gets the sorted names)
>key 11, 4          (11th position; 4 characters long)
>end                (leave SORT)
```

STATISTICS

(9 lines of uninteresting information)

```
END OF PROGRAM
:editor
(I.D. date & time)
/t listdate
FILE UNNUMBERED
/l all
  1      C. Nero      37  empr
  2      J. Donne    1572 poet
  3      S. Pepys    1633 wrtr
  4      J. Cruz     1651 wrtr
  5      J. Swift    1667 wrtr
```

There it is! Everything shuffled sequentially, according to birth date... from Nero to Beria. And, it was placed properly into the file that was prepared for it.

That's a real demonstration of the computer's power.

Now, a few cleanup steps, so that the next person doesn't have your debris to worry about. If you're through with them, purge the three files you made during this part: Listraw, Listdate and Listalfa.

Type: `PURGE LISTDATE` Return

then: `PURGE LISTALFA` Return

then: `PURGE LISTRAW` Return

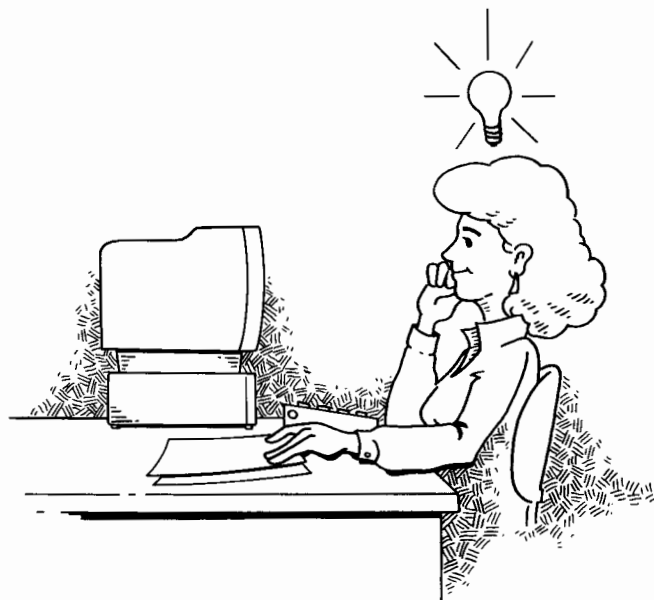
and, to check: `LISTF` Return

Examine the LISTF file list. Are there any of your files still showing except for Jones? This is a good time to purge them, as well. For instance, is Name1 still there?

Use the procedure above, and check when you're through to make sure they're gone.

Sorting Applications

You can already begin to see many of the things you could do with this ability. If you're imaginative, there are hundreds of uses in a normal business.



Using almost any file item as a Key:

- names
- dates
- titles
- telephone numbers
- zip codes
- inventory numbers
- payroll numbers
- part/model numbers
- locations

You can do such things as:

- sort a file's lines alphabetically: from A to Z
- sort a file's lines alphabetically: from Z to A
- sort a file's lines numerically, up: from 0 to 9+
- sort a file's lines numerically, down: from 9+ to 0
- merge together lines from two files.

So, using almost any item in one of your work files as a Key, you can shuffle the contents in almost any manner you might want to.

Alternatively, you can lay out your file contents so they're compatible, and then interleave them in any one of several ways.

If this looks useful, you can find out more about it in the Hewlett-Packard reference manual SORT-MERGE/3000, which carries HP part number: 32214-90001.

Looking Back

You've created files using a couple of different methods, named and renamed them, copied them in several ways, done purges, merges, shuffles, checks, stores and restores, and had one printed. If you've followed these instructions step by step, you've at least been introduced to most of the important concepts about files, and the things that can be done with them.

To check some of this new knowledge...

1. How do you ask for all available information on the command FILE?
2. Can you mix together the contents from two different files? (Yes/No)
3. How do you check on whether or not a file has been properly stored away for you?
4. Which subsystem is used to copy a file?
5. Which two commands are used to give a name to a file that is being created?

ANSWERS

1. `HELP FILE ALL Return`
2. Yes
3. Type: `LISTF Return`
4. FCOPY
5. MPE's BUILD command and EDITOR's KEEP command

USER.ACCOUNT.GROUP	5-1
Everybody Has Files	5-3
Checking the Neighbors	5-5
The SHOWJOB Listing	5-6
Sessions And Jobs	5-8
The Operator And Console	5-10
Security for Your Work	5-12
Passwords When Logging On	5-13
Lockwords On Files	5-15
Remembering Lockwords	5-16
The Pub(lic) Groups	5-18
Creating A PUB File	5-20
File Addresses	5-22
Default File Addresses	5-23
Getting Files From A Distance	5-23
Using Your PUB File	5-25
UDC's — User-Defined Commands	5-27
The Available UDC's	5-28
User-Level UDC's	5-29
Creating A UDC File	5-30
Creating User UDC's	5-31
Testing Your UDC's	5-33
Nesting UDC's	5-35
Account-Level UDC's	5-38
System-Level UDC's	5-39
Looking Back	5-40

Your Account

User.Account, Group

That's how you logged on, remember? "Hello User.Account,Group".

It identified you to the computer as a user named "USER", who was signing on as a member of a group named "GROUP", which belongs to one of the computer's accounts named "ACCOUNT"... as the diagram shows.

Each of the computer's accounts is assigned to a separate major activity. Its "ACCOUNT" account, for instance, is reserved for teaching new users the kind of things you've been learning. It's a dummy account... except, of course, that it actually works.

Unlike most other accounts, it just has one group (except for a PUB group, which we'll learn about later), and it only recognizes one user: "USER", the identity you've been using.

You'll eventually have your own personal identity for logging on. It might, for example, look like:

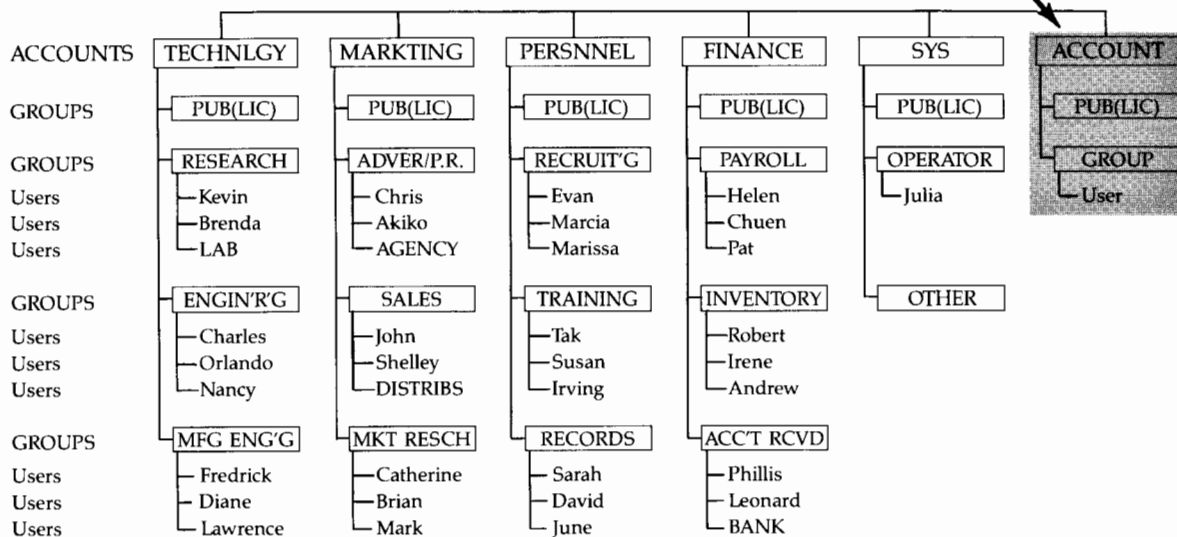
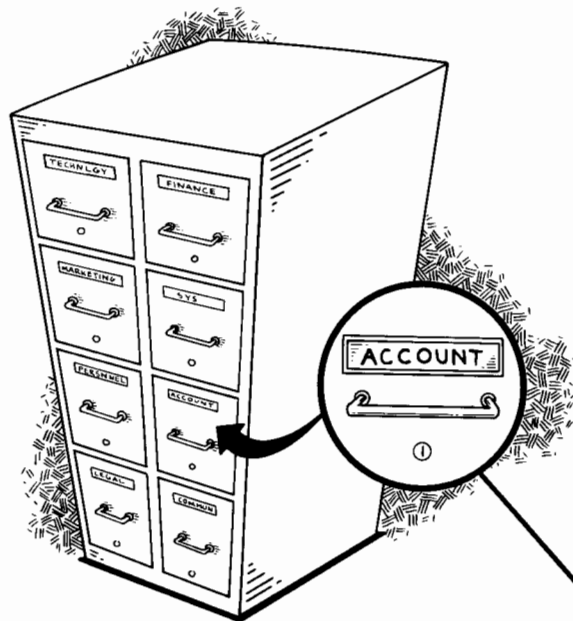
```
Pat.Legal.Contract  
(↑user ↑account ↑group)
```

if your name is Pat, and you're going to be working in your organization's Legal Department, in their Contracts Section. Or, under still other circumstances, it could look like this:

```
Pat.Finance.Payroll  
(↑user ↑account ↑group)
```

if your name is Pat, and you wind up in your Finance Department's Payroll Section. (Notice Pat.Finance,Payroll, by the way, in the chart on the next page.)

5-2 Your Account



This shows the relationships between accounts, groups and users. Take a few minutes to understand the differences here, where they're obvious. Also, notice that it's a lot like an organization chart.

It shows only four organization-type accounts (TECHNLGY, MARKTING, PERSNNEL & FINANCE), plus SYS, and your ACCOUNT account. In real life, a computer could have hundreds of accounts! Or, as few as two or three, when the machine is dedicated to just one or two people, or major projects.

Here are some things that the chart makes clear.

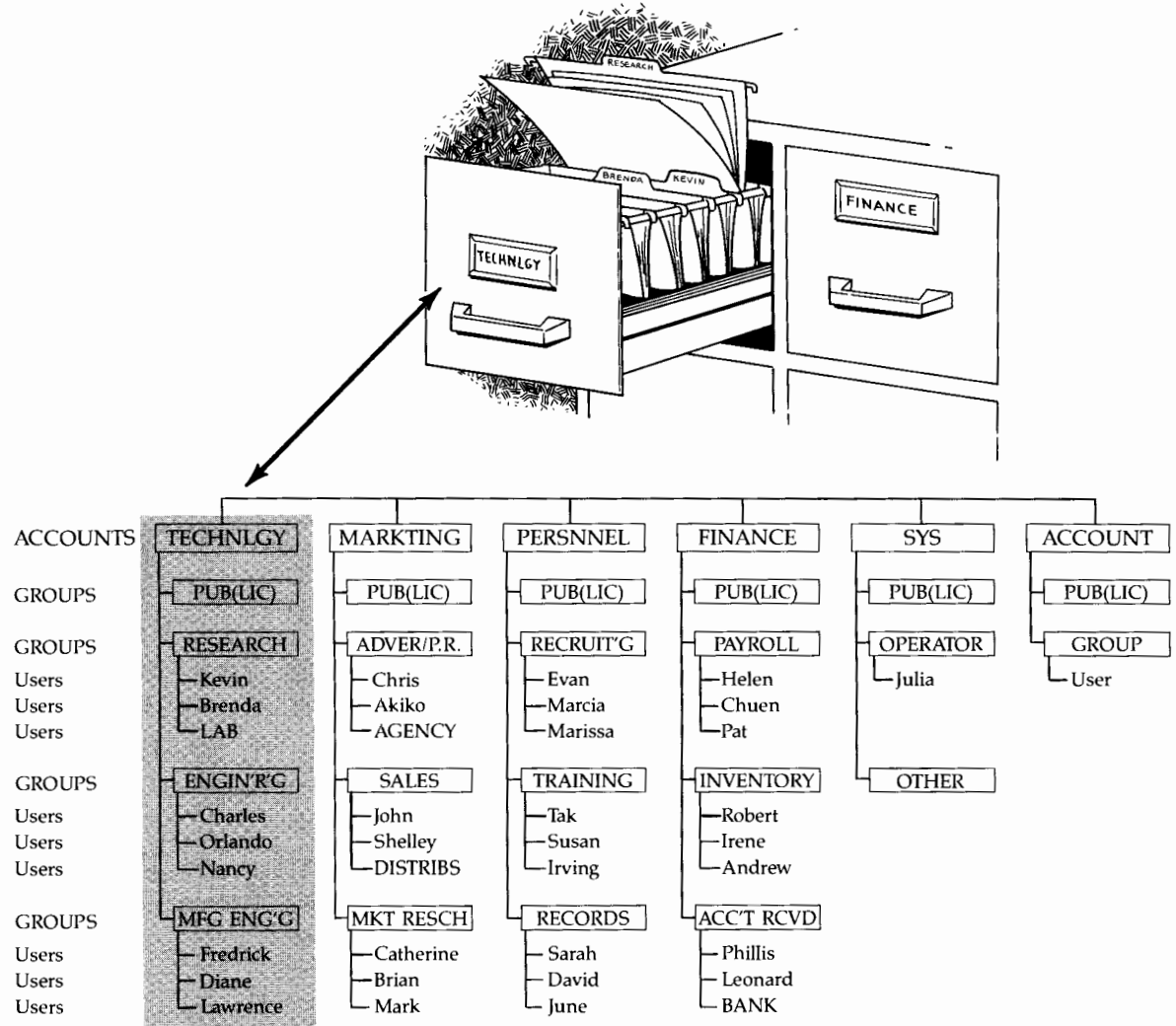
- Your ACCOUNT structure and files are just as real as any of the others, even though they're perhaps smaller, and isolated.
- The isolation means you can make mistakes without hurting the computer, destroying valuable files or bothering "real" users, so feel free to experiment and take chances.
- When the time comes, you can transfer from ACCOUNT to, for instance, FINANCE with confidence.
- Note: some of the "users" are entire organizations!
- Every Account has a PUB Group. This is a place for keeping PUB(lic) files, programs, and other things that're useful for everybody in that account.

Everybody Has Files

The computer's file capacity is VERY large. It has room for almost anything you might need to store, in either one of two ways.

- On line storage... meaning, things that are tucked away on the computer's disc, that are available instantly, just by asking for them. The amount of this storage space is somewhat limited.
- Off line storage... meaning, things that are put away, usually on tape, as with the STORE command. Availability is almost always slow, but the amount of potential storage is very, very large.

Even in on line storage, everybody's files are actually kept separately, as suggested here:

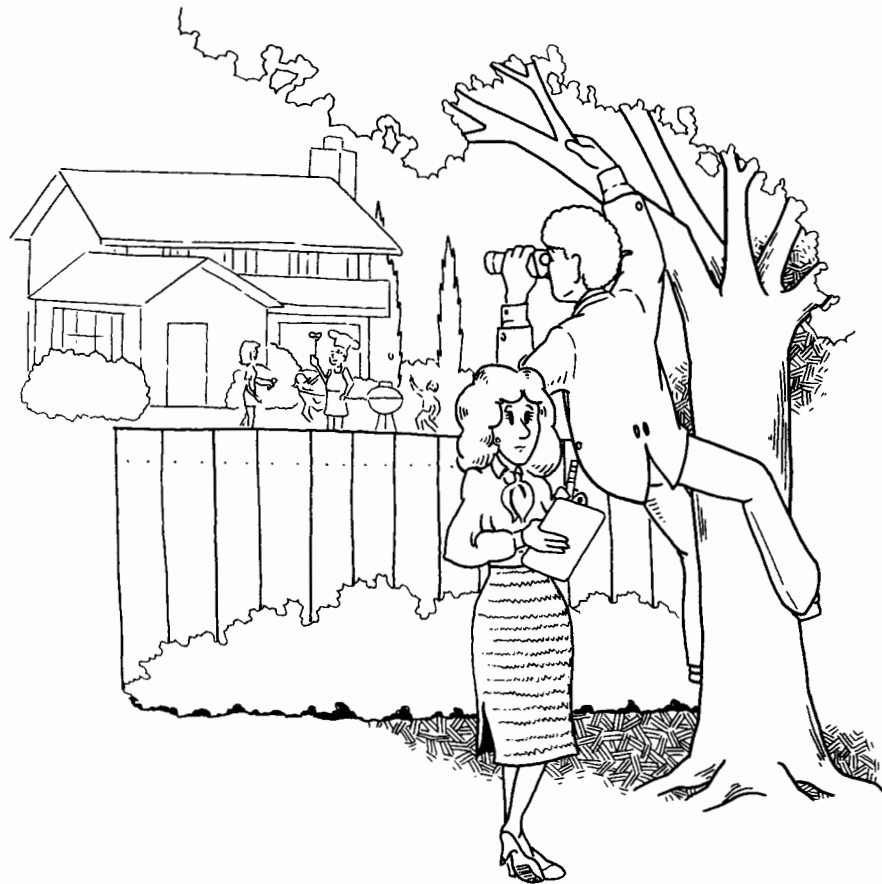


The picture shows the TECHNOLOGY account drawer pulled open, to display three main group's separators: Research, Engineering, and Manufacturing Engineering. The Research Group's subsections for users can be seen: Kevin, Brenda and LAB. Each of the subsections is obviously filled with work files. And each of these can be similar to the files you're already familiar with.

There can be as many "file drawers" as are needed to make sure that everything gets filed in the proper place, and to be sure that no person (or organization) using the computer is allowed access to files or information that isn't theirs.

Checking the Neighbors

In the chart, you can see all the accounts that are showing. Of course, this is an imagined organization, not the real one where you work. Your organization account structure could be just like the chart, or totally different.



5-6 Your Account

No matter HOW different, all systems will have two things in common: they will all have an ACCOUNT account, for learning, as you've been doing, and a SYS(tem) account... which is the System Operator's account, and which will be discussed at greater length later in this chapter.

Your computer may have as few as two or three terminals, or as many as hundreds. Some may be close enough to see, and others could even be in different buildings.

Your organization may also have more than one computer.

Not all of your computer's terminals will be in use at a given time. People doing word processing or financial data entry, for instance, may keep them going all day long, but other people may just use them for short projects... on, and then off again.

The SHOWJOB Listing

To see a report on what's happening at the moment, you use the command called SHOWJOB.

Give it a try. Type: `SHOWJOB Return`

You should get a table, scrolling upward on your screen, that looks roughly like the example below. Few of the details on your screen will match those below, point by point, but the effect ought to be about like this:

```
:showjob
```

JOBNUM	STATE	IPRI	JIN	JLIST	INTRODUCED	JOB NAME
#S1	EXEC		20	20	WED 4:42P	OPERATOR.SYS
#J24	EXEC		10S	PP	WED 5:49P	MM,MAIL.HPOFFICE
#S139	EXEC	QUIET	27	27	THU 8:31A	EVAN.PERSNNEL
#S136	EXEC		31	31	THU 7:46A	DAN,NANCY.TECHNLGY
#S144	EXEC		133	133	THU 10:06A	JOHN.MARKTING
#S151	EXEC		41	41	THU 1:13P	PAT,USER.ACCOUNT
#J25	EXEC		10S	EP0C	THU 6:38A	ED.MAILROOM
#S138	EXEC	QUIET	23	23	THU 8:24A	PAT.FINANCE
#S152	EXEC		39	39	THU 2:17P	AKIKO,USER.ACCOUNT

(plus, perhaps, many more lines)

```
46 JOBS (DISPLAYED):
  0 INTRO
  0 WAIT; INCL 0 DEFERRED
  46 EXEC; INCL 41 SESSIONS
  0 SUSP
JOBFENCE= 6; JLIMIT= 8; SLIMIT= 50
```

IMPORTANT:

If you're on a small system, there may only be a few lines.

IMPORTANT:

If there are LOTS of users on your computer system, you may have to do it again to return to the first lines. Retype SHOWJOB, and then use CTRL plus S and CTRL plus Q to control the scroll.

The table's headings are largely self-evident:

- JOBNUM J(job) and S(ession) numbers, each sequential
- STATE INTRO(duced), EXEC(uting), SUSP(ended) or WAIT(ing)
for an appropriate device

5-8 Your Account

- IPRI QUIET can't receive messages; others can
- JIN how the computer identifies job sources
- JLIST the job's working devices: PP = printer; 41 = your terminal; 20 = the operator's console; 133 = from another computer
- INTRODUCED is the log on day and time, if the computer's clock was set properly
- JOB NAME part of log on identification, for I.D. notice: it doesn't show group's names

Each line's a separate task that the computer's doing right now.

- When someone logs on, they appear on the list.
- The moment they log off, they disappear from it.

Sessions And Jobs

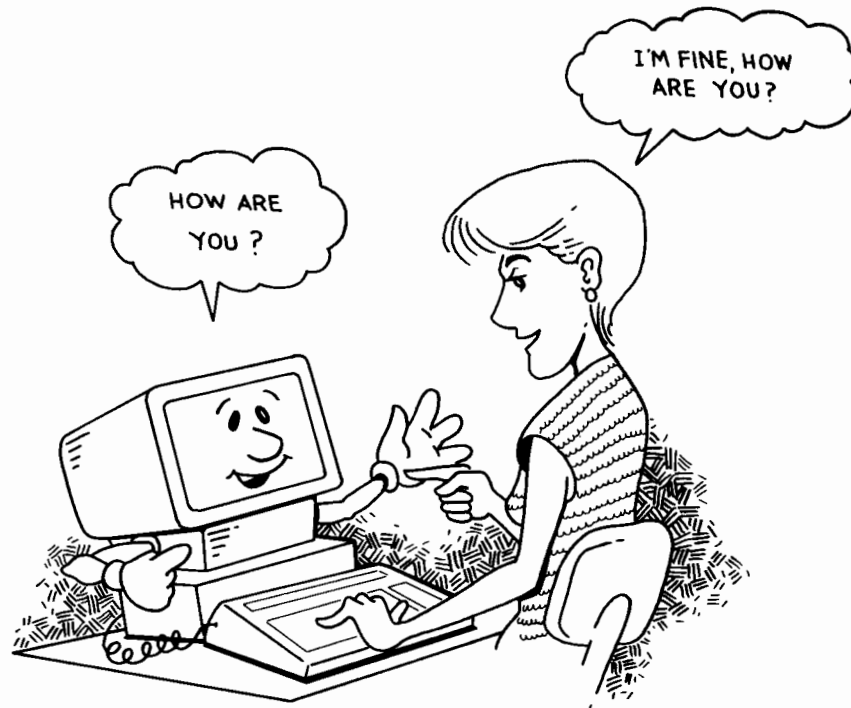
You've found YOUR line, haven't you? It's the one that looks something like this:

```
#S151 EXEC 41 41 THU 1:13P PAT,USER.ACCOUNT
↑↑↑ ↑↑ ↑↑ ↑↑↑↑↑
```

except, your line on the screen probably has different information where the ↑↑ symbols are.

Also, notice your name, Pat, or whatever you used. This is how the computer keeps track of you separately from the user named Akiko, and from anyone else who may be using the USER.ACCOUNT at this time. This can be useful for assessing computer time charges to the organizations using its services, or for other tracking purposes.

SESSIONS have numbers starting with #S. They could be as low as #S1, or as high as #S9999. The numbers start over again at #S1 whenever the computer is shut down, and then restarted. Session number #S1 is automatically assigned to the Operator (OPERATOR.SYS).

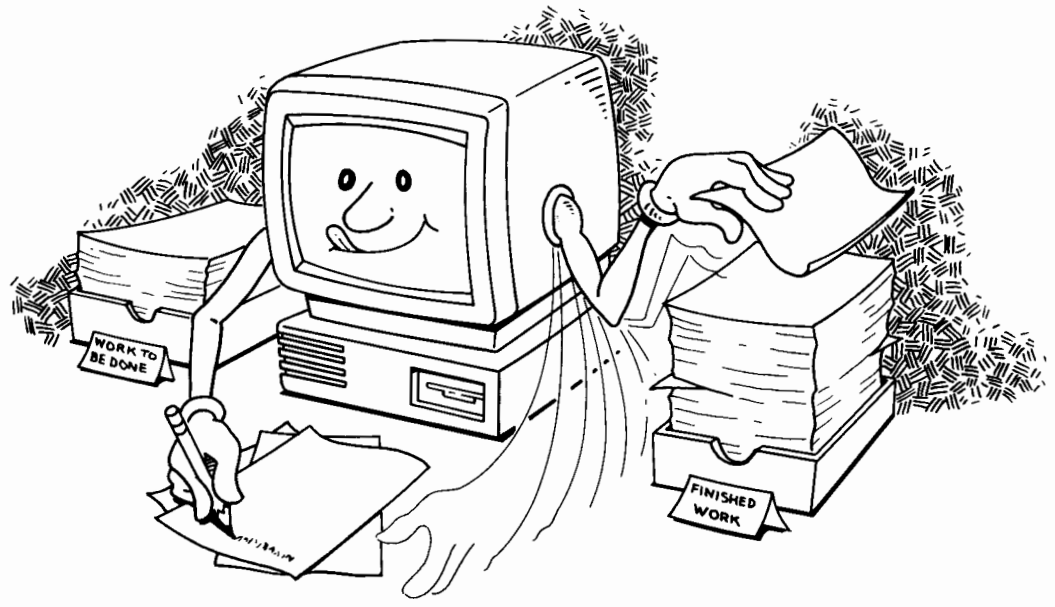


Sessions

If you remember earlier descriptions, a session is when you and the computer have sort of a "conversation" as a way of getting things done. As an example, you're having a session with the computer right now.

JOBS have numbers starting with #J. They, also, could be as low as #J1, or as high as #J9999, but the job numbers tend to be smaller, because there are usually fewer jobs.

A job is when the computer is working at ITS own speed, rather than YOURS. It's sometimes referred to as "batch processing", because the computer handles things in big end-to-end batches, working very efficiently, and just as fast as it can.



Jobs

The second line of the sample SHOWJOB listing is fairly typical of jobs:

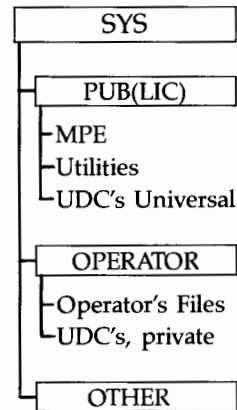
JOBNUM	STATE	IPRI	JIN	JLIST	INTRODUCED	JOB NAME
#J24	EXEC		10S	PP	WED 5:49P	MM-MAIL.HPOFFICE

where "10S" (the job source) is a system device number; "PP" is its definition, the line printer; "MM" identifies the person doing the job (like the "Pat," identity you added during your first log on); and "MAIL.HPOFFICE" describes the work being done.

The Operator And Console

One of your most important neighbors is the Operator... one of the people who use the SYS(tem) account. This is probably the best source of information you're likely to have... on the subjects of the computer, its MPE operating system, and your terminal.

The SYS(tem) account is where the MPE operating system is kept... much of it in the PUB(lic) group, as a matter of fact.



Also in the SYS(tem) account's PUB(lic) group, the Operator keeps a bunch of programs called "utilities". They are programs that help the computer run smoothly, let it rescue files when it goes down, and restore things later.

The last important things you'll find in SYS.PUB are some "publically available" UDC's. UDC's are U(ser) D(efined) C(ommands). These are usually shortcuts for the sometimes long computer procedures that you find yourself stuck doing, over and over again... like getting into EDITOR to check a text file. They can even be abbreviations of long commands that you use often... like using SJ instead of the full command SHOWJOB.

UDC's are usually added on an individual basis by each Operator, to fit local needs. There's no way to tell ahead of time which ones will be on your system. You'll find more information about them further on in this section.

As the diagram suggests, the Operator often also keeps some non-PUB(lic) files and UDC's.

Referring back to the sample SHOWJOB listing on page 5-7, Session #S1 is almost automatically assigned to the Operator as part of the computer system startup process. The line #S1... OPERATOR.SYS will remain, and at the top of the SHOWJOB list, until the Operator logs off. Then, after logging back on, the Operator would be assigned the next available session number.

The “console” is a name given to the Operator’s terminal. There is only one console per system, and it’s always device #20, unless the Operator commands that it be another, for some reason of temporary convenience.

Security For Your Work

Most people (and organizations) have a whole list of things they prefer to keep private: payroll figures, some other personnel data, future plans, technical/scientific developments, and even personal correspondence.

Sometimes it’s important to keep things away from the outside world, sometimes from other sections of your own organization, and sometimes even from those who work beside you.



The computer, thanks to MPE, allows that kind of security to every single one of its users.

No one can get into any file that they don't have a right to... unless the file is left available, or they're given permission, in some fashion.

- Every account is kept separate from every other, as though they were in different file drawers.
- The computer keeps every group's files apart from every other, even within each account.
- Every user's files can be kept private with regard to every other user's in a given group.
- Each individual file can have its privacy assured with a lockword.

Passwords When Logging On

Logging on, itself, is an important security measure, and the very first one that you face.

If you can't log on, you have no access to the computer, or the files it contains. You can sit at your terminal and do all those things that the terminal, alone, permits you to do, but you can't get into the files.



Even logging on as USER.ACCOUNT, GROUP doesn't let you transfer over to another account, or see files from another account unless you have the responsible person's permission.

When you do get your own personal identity for logging on, you can ask the Operator, or your account manager, for your own personal password. Then, when you begin logging on in this new identity, the computer will ask for proof of it, and you won't be allowed on until you provide it.

See if you can get a log on identity from the Operator. Assuming that your new identity is Pat.Finance, Payroll, and assuming that you've chosen the password "Arkie" again, type this:

```
HELLO PAT.FINANCE,PAYROLL Return
```

When it asks for your password, type: `ARKIE` Return

IMPORTANT:

Remember that, like lockwords, this will be "invisible" as you type it, so nearby people can't read it.

IMPORTANT:

If you make a mistake typing your password, don't worry; the computer will ask you again. You'll have three tries before it gives up on you. After that, you'll still be able to go back and start again.

If you were to make a mistake the first time, and then get it right the second time, your screen should look like this:

```
:hello pat.finance, payroll
ENTER USER PASSWORD:

ENTER USER PASSWORD:

HP3000 / MPE V (the program I.D., etc.)
```

This personal password is very helpful. At very minimum:

- it prevents strangers from logging on in your name;
- it discourages others in your own account from traversing over to check out your work without your knowledge or permission; and
- it even keeps away others in your very own group.

Lockwords On Files

The easiest ways to put lockwords on your files are:

- by adding a slash (/), plus the lockword, after its file name... as you use EDITOR's KEEP command... the technique you learned in Word Processing;
- by adding slash, plus lockword, as a part of naming or renaming your file, as in Using Files; and
- by adding slash, plus lockword, as a part of making a file copy while using the FCOPY subsystem.

You can also change a lockword using the RENAME command. For example, if you were back in your USER.ACCOUNT, GROUP identity, and wanted to change the lockword on your Jones file from ARKIE to TAK, this is the formula.

Type: `RENAME JONES, JONES/TAK` Return

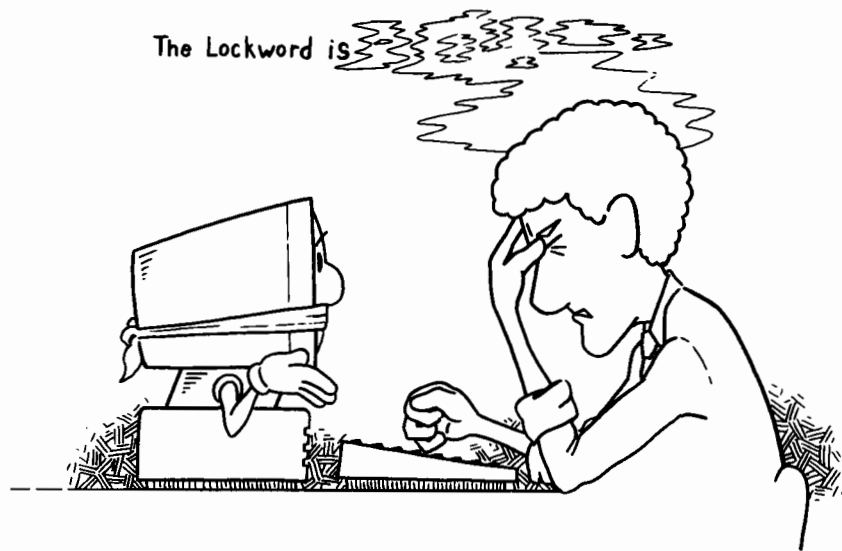
When it asks for the lockword, type: `ARKIE` Return

```
:rename jones,jones/tak
LOCKWORD: JONES.PUB.ACCOUNT?
:
```

If you'd like to check that, go into EDITOR, ask for TEXT JONES and, when it asks for a lockword, type TAK. Then, LIST ALL.

Remembering Lockwords

It's easy to forget lockwords. A day or two after you assign them they'll start to fade, and after a month or two they'll be almost gone... unless you use them constantly, or have a really remarkable memory.



Part of the reason is that you never see them in type on your screen. They're never visually "reinforced".

Then, how can you remember lockwords? Here are some hints that can help.

- DON'T choose obvious words like your nickname, your initials, your favorite musician, or other things your fellow workers might easily guess.
- DON'T pick a different one for every file. The more there are, the harder they'll be to remember.
- DO pick just a few good ones. You might use a single one for all of the files in one given category.

- DON'T tape your list of lockwords on your terminal, where everybody can see them.
- DO consider using, as lockwords, words that are now visible near your terminal, although not obvious (in book titles, labels, decorations, etc.), and that'll stay available for reference for awhile.
- DO consider using just one single lockword on all of your files, but changing it every month or so.

It's clearly easier not to use passwords and lockwords at all. In that best of all worlds, you'd just log on the way you've been doing it, and do your file handling in the simplest, most straightforward way. You're never delayed by those extra steps. You're never frustrated by a forgotten key. Right?

Sure. But, seriously, don't do it! Not unless you've got the whole computer to yourself.

- Take the extra trouble.
- DO use lockwords.
- Change them as necessary.
- Learn to remember them.
- Develop a little paranoia.

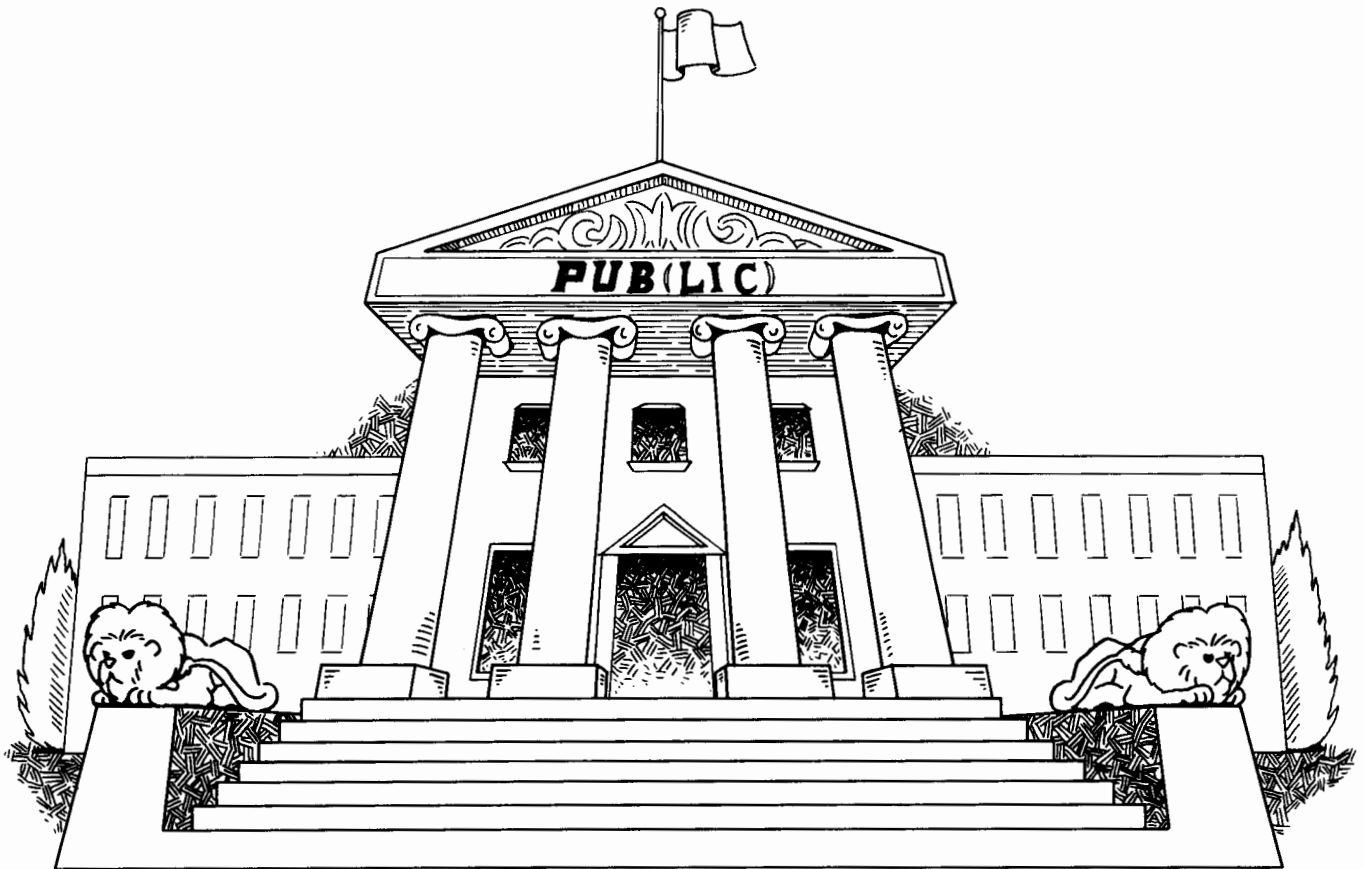
These things will pay. They will not only safeguard your own files, but improve the inaccessibility of the work and files of your group, and even your account. And you should expect the others to do the same for you.

One last way to retrieve a lockword is: ask your account manager. The manager of each account has the authority and knowledge to go into the computer's files, using the LISTUSER command or the LISTDIR5 utility, and dig that information out of their own account.

The PUB(lic) Groups

There's a PUB group in every single account. You'll remember that fact from the chart there at the start of this section. Even your ACCOUNT account was shown as having its own, and you've already learned that's where the Operator keeps the MPE and utility software.

It's called PUB(lic) because, like a public building, it can be used by anybody in the account. Thus, it stores files and programs that may be needed by multiple users, from more than one group.



It's also the default group. This means that anyone who logs on, specifying user and account names but no group name, will be assigned into the PUB group "by default".

That's an easy thing to check.

You're logged on as USER.ACCOUNT, GROUP right? If not, log off, then back on again, properly.

Then type: `S H O W M E Return`

You'll see:

```
:showme
USER: #S##,USER.ACCOUNT,GROUP (NOT IN BREAK)
(etc.)          ↑
```

which includes the familiar identity. Next, ask for the list of files in your group.

Type: `L I S T F Return`

When the list appears, make a little scratchpad note of the names of the first four to six names on the list, as a way of checking.

Next, log off, and then re-log on as just USER.ACCOUNT.

Type: `B Y E Return`

Ask for a fresh prompt: `Return`

and type: `H E L L O U S E R . A C C O U N T Return`

After the computer goes through its identifying process, type:

SHOW ME Return

```
:showme
USER: #S##,USER.ACCOUNT,PUB  (NOT IN BREAK)
(etc.)                        ↑
```

See that? You've defaulted into your account's PUB group.

To see this further, type: **L I S T F** Return

The list of file names that appears should be different than the list from GROUP group.

Also, check the changes with: [SHOW JOB](#) [Return](#)

Creating A PUB File

As an exercise, use EDITOR to create a small file here in the PUB group.

Make up a name and a lockword. Don't use the ones shown here. Then, incorporate them as text in the new file. Substitute them for the Coffee and Cup combination used below.

Type: EDITOR Return

next, start the new file with: `A Return`

```
type: A FILE NAMED C O F F E E Return
      ↑ use your choice
```

and: WITH THE LOCKWORD CUP Return
 ↑ here, too

ending it with: `// Return`

Keep it by typing: `K C O F F E E / C U P` Return

```
then: L A L L Return
```

Check that it worked. Type: `T C O F F E E Return`
 ↑ *here, too*

In response, type your lockword: `C U P Return`
 ↑ *here, too*

then: `L A L L Return`

At the end, exit EDITOR with: `E Return`

Your screen should now show something like this:

```
/a
    1      A file named Coffee
    2      with the lockword Cup
    3      //
...
/k coffee/cup
/t coffee
LOCKWORD: COFFEE.PUB.ACCOUNT?

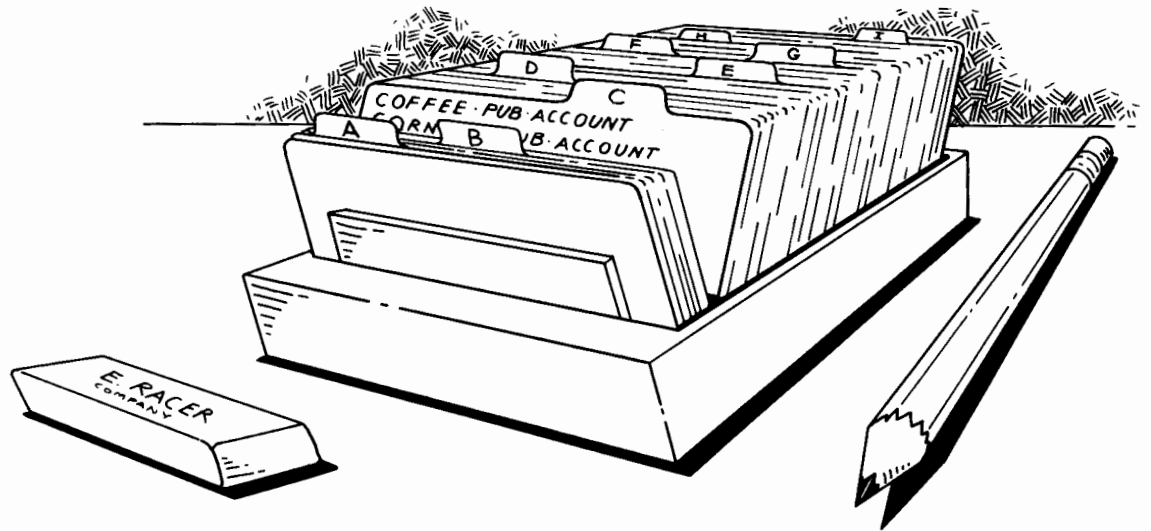
/l all
    1      A file named Coffee
    2      with the lockword Cup
/e
```

Finally, if you want to assure yourself that the file really got formally stored, check it with LISTF command.

File Addresses

While you're here in the PUB group, is it possible to ask for a file that you left behind you in the GROUP group?

Notice, above, the file name:
COFFEE.PUB.ACCOUNT



That's very important! It's the computer's "address" for your file. Notice its structure: the file's name, a period, the group's name, a period, and the account's name. The computer needs those items, in that order, to find a file... in particular if the file (or program) is in a different group, or another account.

This is worth memorizing, because much of the work that people do on computers involves working with files from other users, other groups, and other accounts.

Default File Addresses

You have already used default file addresses. You've used them from the very beginning.

Back in the Word Processing Section, you used the JOIN command to join file Pat... which you were working on... to file Jones... which was stored away. You typed: JOIN JONES, and the computer answered:

LOCKWORD: JONES.GROUP.ACCOUNT?



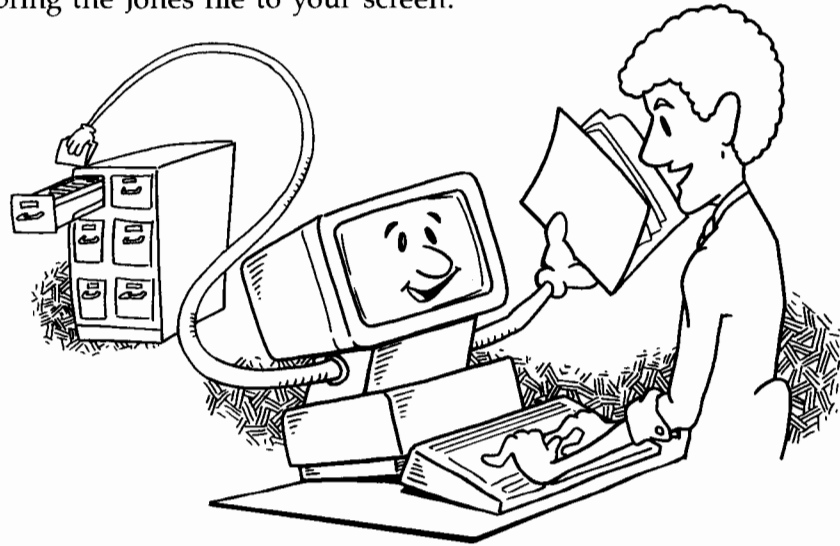
The computer assumed that, since you were in GROUP group and ACCOUNT account at the time, any file you asked for would have to be there, too.

Every time you asked for a file, like Name1, or Listraw, you should have added the identifiers .GROUP.ACCOUNT to it. Since you didn't, the computer supplied them.

You can continue to leave them off, as long as you're in the proper place for them to default correctly.

Getting Files From A Distance

You still have at least one file over in the GROUP group: the one referred to as the Jones file. You're going to go into EDITOR here, in the PUB group, and bring the Jones file to your screen.



This technique works for files in other groups, and other accounts. It also works on programs, like the utilities, and things with lockwords on them.

Type: `EDITOR` Return

and: `T JONES.GROUP.ACCOUNT` Return

When it asks for the lockword, type: `ARKIE` Return

and when the slash returns, type: `L ALL` Return

The sequence should look like this:

```
/t jones.group.account
LOCKWORD: JONES.GROUP.ACCOUNT?

/l all
  l      (your text)
```

IMPORTANT:

If it didn't happen that way, read the computer's messages to you, and follow them as well as you can; or repeat carefully, looking for missed steps.

If all went well, the file came to your screen from its storage place in the other group. You can bring any of them over to work on, as long as you know the address and lockword.

Using Your PUB File

Now, try doing the same thing again, but from back in the GROUP group. END your stay in EDITOR, log off, log back on again into the GROUP group, get back into EDITOR, and ask for TEXT file Coffee, or whatever you named it.

Type: **E** **Return**

then: **BYE** **Return**

Press: **Return** for a fresh colon prompt,

then: **HELLO USER.ACCOUNT, GROUP** **Return**

then, after the identification: **EDITOR** **Return**

then: **TCOFFEE.PUB.ACCOUNT** **Return**

After the lockword request, type: **CUP** **Return**

and: **LALL** **Return**

If this appears on the screen:

```
+--F-I-L-E---I-N-F-O-R-M-A-T-I-O-N---D-I-S-P-L-A-Y+
!  ERROR NUMBER: 52      RESIDUE: 0                      !
!  BLOCK NUMBER: 0      NUMREC: 0                        !
+-----+
*23*FAILURE TO OPEN TEXT FILE (52)
NONEXISTENT PERMANENT FILE (FSERR 52)
```

you probably used an incorrect address or an incomplete one. Check it carefully, and try it again.

If, on the other hand, everything fell into place, it should look like this:

```

/e

END OF SUBSYSTEM
:bye

CPU=##.CONNECT=## (day/date/time)

hello user.account.group
HP3000 + log on identification
:editor
(editor I.D.)
/t coffee.pub.account
LOCKWORD.COFFEE.PUB.ACCOUNT?

/l all
    1      A file named Coffee
    2      with the lockword Cup

```

This is the way to leave commonly used working files in a central location... the PUB group... where they can be conveniently available to everybody on the computer, and to work with them without leaving your group.

You're also getting a clearer idea of how easy it is to get into other people's files. With a bit of skill, just about anyone can rumage around, pull up a list of your file names, and ask to see them. So:

- use passwords and lockwords, even if you're alone in an account; and
- don't tell your lockwords to ANYONE! If you're letting someone into a file, YOU type the lockwords for them.

UDC'S — User-Defined Commands

Commands are your tools for controlling the computer. Each one does a different task, from ABORT to XPLAIN.

There are hundreds of them.

sm=SHOWME

But there are always just a few more things you'd like to be able to do, if you were using the best of all possible computer systems. Maybe, condense an often-used process down to a single step, or shorten a common command to a couple of letters... like "sm" for SHOWME.

A UDC, then, is a command that YOU design for your own convenience.

There are three kinds... or levels... of UDC's.

- User UDC's are put into the catalog by individuals, like you, to reduce the number of steps in their own, personal, daily jobs.
- Account UDC's are commands that are more generally useful throughout the account, and are installed by the account's Manager.
- System UDC's are felt to be the most widely useful, so are available to everyone who logs on. They're put in place by the system's Manager or Operator.

You'll probably use a lot of system UDC's during your work, but you may take a certain pleasure in tailoring your job with user UDC's.

logon=LOGGING ON

The Available UDC's

A computer that was just installed will have no UDC's in it, probably. They are, after all, USER-defined, and you haven't defined them yet.

sc=SHOWCATALOG

But, if your computer's been around for awhile, it should begin to have at least some system UDC's in its catalog, and maybe some others.

To find out what's there, use the command SHOWCATALOG. If the catalog listing is a long one, you may want to be ready to control scrolling with the CTRL plus S and CTRL plus Q technique.

Type: **S H O W C A T A L O G** Return

What you see on your screen may be only generally like the example. Yours may have only system UDC's, and the headings may be different, but if there are any UDC's at all, it'll look like this:

```
:showcatalog
SHOWCATALOG

USERUDC.PUB.ACCOUNT
  LOGON          USER
  INFO           USER
ACCTUDC.PUB.ACCOUNT
  OPENDS         ACCOUNT
  HOME           ACCOUNT
SYSUDC.UTIL.SYS
  NEWS           SYSTEM
  LDEV           SYSTEM
  LISTDIRS       SYSTEM
  SC             SYSTEM
```

Those are the UDC's that are available to you in whichever account you're in.

IMPORTANT:

As you go from group to group, and from account to account, it's worth using such status commands as SHOWCATALOG to see what kind of things change for you.

Try using them, particularly if some are user UDC's. Try typing their names. Watch to see what happens. If nothing happens, try typing the command RUN, followed by the UDC name. If something happens, try to follow what it means. And experiment.

news = ANNOUNCEMENTS

If nothing happens, don't be disappointed in the process quite yet.

User-Level UDC's

User-level UDC's are for YOUR OWN personal work convenience. They are for simplifying the unique processes that you go through as you work at your terminal. When properly coded, your personal UDC's can't be used by anyone else, without your permission and help.

Almost anything that you do over and over again can be shrunk to a single, simple command.

- A long command, like SHOWCATALOG, can be shortened to a couple of letters, like: SC.
- A several-step process, like logging onto another computer, can be compacted into: LOGON.
- An intricate set of tasks, like printing out files, can be condensed to, for instance: PRNT.

Creating A UDC File

It's expected that you won't be able to create UDC's as a USER user from the ACCOUNT account. You'll need your own real, log on identity for this.

Ask your Operator for one.

sm = SHOWME

Create a UDC file with EDITOR. Finish it with *****. Keep it, using a file name made up of your initials plus UDC, and use your initials again, as a lockword. Leave EDITOR, and use LISTF to make sure the file was opened.

Log on, and type: **E D I T O R** Return

Show that you want to add, with: **A** Return

Type the UDC: **S M** Return

and then the command: **S H O W M E** Return

End the UDC with: *** * * * *** Return

and then finish the entry with: **/ /** Return

Keep it by typing: **K** **A J E U D C / A J E** Return
 ↑ ↑ ↑ ↑ ↑ ↑

use your own initials

Leave EDITOR with: **E** Return

Check whether the file appeared, with: **L I S T F** Return

IMPORTANT:

The SM, for SHOWME, is your UDC file's first entry, but it hasn't yet been made into a functional UDC.

So far, it should look roughly like this, except with your own initials:

```
:editor
(I.D., date & time)
/a
    1      sm
    2      showme
    3      *****
    4      //
...
/k ajeudc/aje
/e

END OF SUBSYSTEM

:listf

FILENAME

AJEUDC      (and others, arranged alphabetically)
```

Creating User UDC's

In the next part you'll use the command SETCATALOG twice, to trigger your UDC file, and to reset it, activating its UDC's. You'll also add a second UDC (SC = SHOWCATALOG) in the same way as is shown above.

SC=SHOWCATALOG

Release the user UDC's with: **SETCATALOG** Return

then: **EDITOR** Return

Call for your file by typing: **T AJEUDC** Return

and, when it asks for a lockword: **AJE** Return

Tell it you want to add, with: **A** Return

After the 4, type the second UDC: **SC** Return
 and its command: **SHOWCATALOG** Return
 Finish the UDC with: ********* Return
 and end the addition with: **//** Return

Keep it with: **K** **AJEUDC** Return

Type the lockword: **AJE** Return

Purge old and keep new with: **YES** Return

Leave EDITOR by typing: **E** Return

Set it with: **SETCATALOG** **AJEUDC** Return

```
:setcatalog
:editor
(I.D., date & time)
/t ajeudc
LOCKWORD: AJEUDC.PUB.FINANCE?

/a
    4      sc
    5      showcatalog
    6      *****
    7      //

...

/k ajeudc
LOCKWORD: AJEUDC.PUB.FINANCE?

AJEUDC ALREADY EXISTS - RESPOND YES TO PURGE OLD AND KEEP NEW
PURGE OLD?yes
/e

END OF SUBSYSTEM
:setcatalog ajeudc
LOCKWORD:
```

Adding new UDC's to your list is done by repeating the elements in this last section, covering every step between the two uses of SETCATALOG.

Every time you add a UDC, the program will present the next open, available text line to you. For instance: UDC SC, above, has just used lines 4, 5 & 6. So, when you add the next one, the EDITOR program will start you at text line 7.

A UDC can use more than the three lines shown in the samples. In a complex UDC, each step or stage will take a separate line, and there can be dozens of steps.

IMPORTANT:

Modifying UDC's is tricky. Try to avoid that step until you're ready to read Chapter III of the MPE V COMMANDS manual (32033-90006). It deals with all aspects of UDC's in greater detail.

Testing Your UDC's

Testing this particular pair of UDC's offers the extra advantage of "two birds with one stone".

sm/sc

First, type: **SM** Return

You should see the familiar paragraph identifying you as you are currently logged on. If the Operator has given you your new I.D., this should be apparent.

Next, type: **SC** Return

You'll see the familiar UDC list. Near the top, in the USER section, will be your new ones, probably next to each other, and in the order that you created them. They could even be all alone in their own new USER section, if the Operator gave you the right kind of I.D. Later, you may also see the UDC's SM and SC, in the SYSTEM section, if your Operator felt they were useful enough.

IMPORTANT:

If these things DON'T happen, retrace your steps carefully through the whole procedure, starting with creating the UDC file.

Nesting UDC's Putting two or more simple UDC's into a larger one is called "nesting".

Nesting is a simple way to make a complex UDC.



Imagine causing the three commands LISTF, SHOWJOB and SHOWTIME to happen just by typing the nesting command arbitrarily named DOIT.

To do a sample one of these, try to do all of these steps during a single session.

Start by typing: `SETCATALOG Return`

then: `EDITOR Return`

Ask for your UDC file with: `T AJEUDC Return`

Supply your lockword: `AJE Return`

and start to add by typing: `A Return`

Type the nesting command: `DOIT Return`

plus the first desired UDC: `AA Return`

and the second desired UDC: `AB Return`

the third desired UDC: `AC Return`

and finish the nest with: `***** Return`

Repeat the first desired UDC: `AA Return`

plus its definition: `LISITF Return`

and its ending: `***** Return`

Repeat the second desired UDC: `AB Return`

plus its definition: `SHOWJOB Return`

and its ending: `***** Return`

Repeat the third desired UDC: `AC Return`

plus its definition: `SHOWTIME Return`

and its ending: `***** Return`

End the add with: `// Return`

Save it by typing: `K AJEUDC Return`

Supply your lockword: `AJE Return`

Purge old and keep new with: `YES Return`

Leave EDITOR by typing: `E Return`

Set it with: `SETCATALOG AJEUDC Return`

And give your lockword again: `AJE Return`

That should do it. Check it over to make sure nothing was missed. If all went well, it should look like this on your screen:

```
:setcatalog
:editor
(I.D., date & time)
/t ajeudc
LOCKWORD: AJEUDC.PUB.FINANCE?

/a
    7      doit
    8      aa
    9      ab
   10      ac
   11      *****
   12      aa
   13      listf
   14      *****
   15      ab
   16      showjob
   17      *****
   18      ac
   19      showtime
   20      *****
   21      //
...
/k ajeudc
LOCKWORD: AJEUDC.PUB.FINANCE?

AJEUDC ALREADY EXISTS - RESPOND YES TO PURGE OLD AND KEEP NEW
PURGE OLD?yes
/e

END OF SUBSYSTEM
:setcatalog ajeudc
LOCKWORD:
```

So, then... the only thing left to do is to try it. Do it! Type: **DOIT** Return

You should see a LISTF, a SHOWJOB and a SHOWTIME happen, smoothly, one after the other. This will happen every time you type DOIT from now on, until you make a change to the UDC.

IMPORTANT:

Nested UDC's only work forward (or downward). The nesting command MUST be further back (or higher up) than any of the UDC's that you want to use in it. UDC's put in before (above) the nesting command, can't be "seen" by it.

Account-Level UDC's

These commands are usually meant to simplify some of the complex processes that certain people in your account routinely have to plod through. These could include:

- logging across to another account for commonly used files;
- going quickly from one system UDC to another; and
- ordering your system's printer to print a file.

They often start out as creations of one impatient user, who sets out to streamline a process. They are then noticed as potentially useful to many other people in the account. Anyone can suggest that a UDC be included in your account's catalog.

The account manager is the person who usually approves them and installs them.

System-Level UDC's

System-level UDC's are more likely to be general kinds of commands, like utilities or program subsystems. They have to be available to everyone who is using the computer... across every account.

They can be used for such helpful things as:

- to have single-step logging onto support services;
- to run frequently-needed utility sets; and
- to make simple connections to other computers in your "local area network".

They can be created by anyone. You, for instance.

They're usually installed by the Operator, the system's Supervisor, or its Manager (this not the same as an account manager).

Looking Back

You should be a lot more comfortable with your account, by now... and with the other accounts on your computer: how they're structured, and how to use them.

These questions will help you review this new knowledge.

1. Which group will you always find in an account, no matter what else it may have?
2. If you were working in the Finance account, and wanted to use a file from the Persnnel account's Training group (a file named Memos) to work with, how would you address your request?
3. Which two accounts will always be found in the HP 3000 MPE operating system?
4. Which command is used to create UDC's?
5. Using the KEEP command, what sequence would you type to add the lockword "Market" to a file named "Sales"?

ANSWERS

1. A PUB group.
2. `MEMOS.RECORDS.PERSNNEL` Return
3. The ACCOUNT account, and the Operator's SYS account.
4. SETCATALOG.
5. `K SALES/MARKET` Return

6

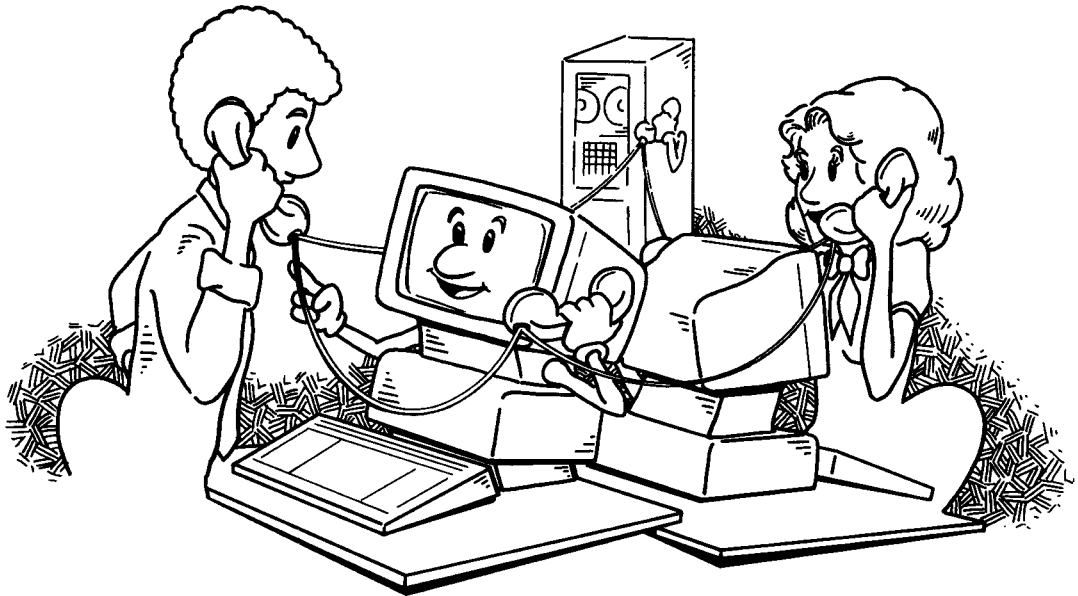
Messages

Your Communication Network.....	6-1
Messages to the Operator.....	6-2
A Message to Another User.....	6-4
Refusing Messages.....	6-6
Status Messages.....	6-8
The SHOW- Status Commands.....	6-9
Non-SHOW- Status Commands.....	6-10
"ERROR" Messages.....	6-12
"FAILURE" Messages.....	6-14
Looking Back.....	6-17

Messages

Your Communication Network

The computer and its terminals are a network for handling messages. It deals with many different kinds. It stores them, triggers and distributes them. In the process, it does what it can to provide useful information to both the users and the Operator.



Some of the kinds of information it handles include:

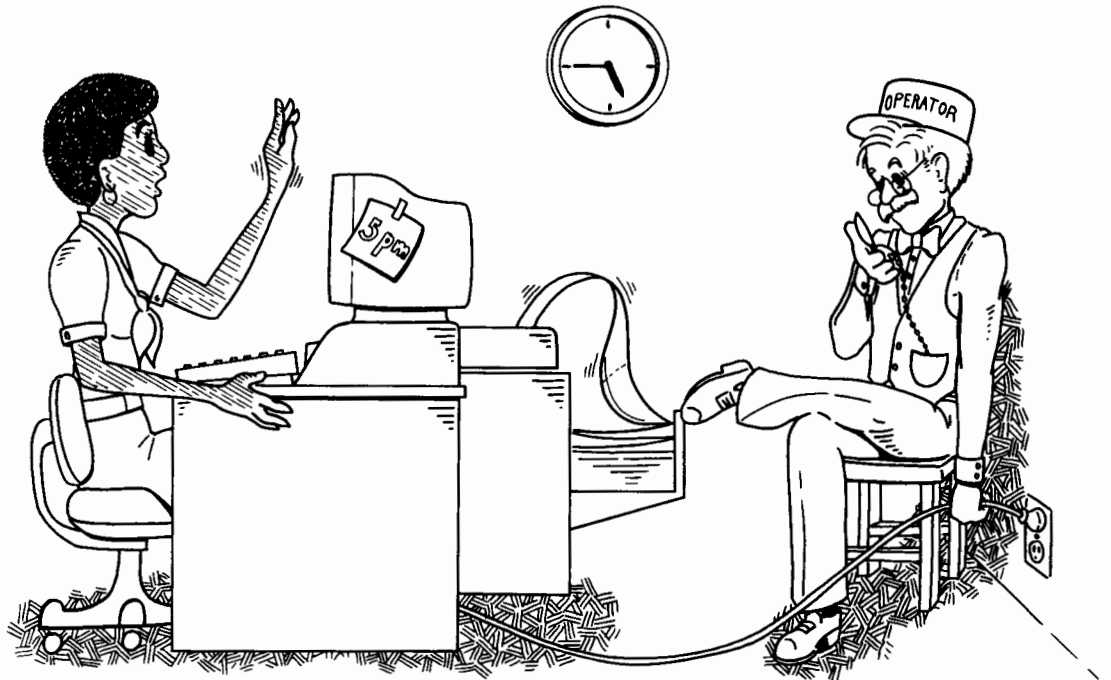
- your messages to the console... or the Operator,
- your messages to the other users,
- status messages to you from the computer,
- "error" messages to you from the computer, and
- non-error messages from the computer.

The goal is to keep everybody informed, and keep the jobs and sessions moving smoothly.

Messages to the Operator

It's quarter to five. You have one more file to print before you leave. It'll take about twenty minutes. Easy, right?

Then you remember: when you logged on this morning, there was a note from the Operator warning you that the system would shut down at five for a "dump" and modifications.



That puts you in a bind. You either have to quit before you're ready to, or try to squeeze the job in at the last minute, and hope you make it.

No problem! There's a command called TELLOP that'll let you send a message to the Operator... or at least to the console... warning that you're going to need a little extra time before the system gets shut down.

Try an experimental TELLOP message to see how it works.

Type: **TE L L O P T H I S I S A T R I A L** Return

By now you should know where your computer's console is. If not, this is a good time to find out. Ask someone. Then go, and see your announcement on its screen.

IMPORTANT:

Do not touch the console, or its keyboard unless you're invited to by the Operator. Be particularly careful if the console is an HP 150 TOUCHSCREEN.

It IS there, isn't it? It should look sort of like this:

```
16:47/#S131/19/FROM/PAT.FINANCE/This is a trial
  ↑ (time)   ↑ (session)   ↑ (i.d.)   ↑ (message)
```

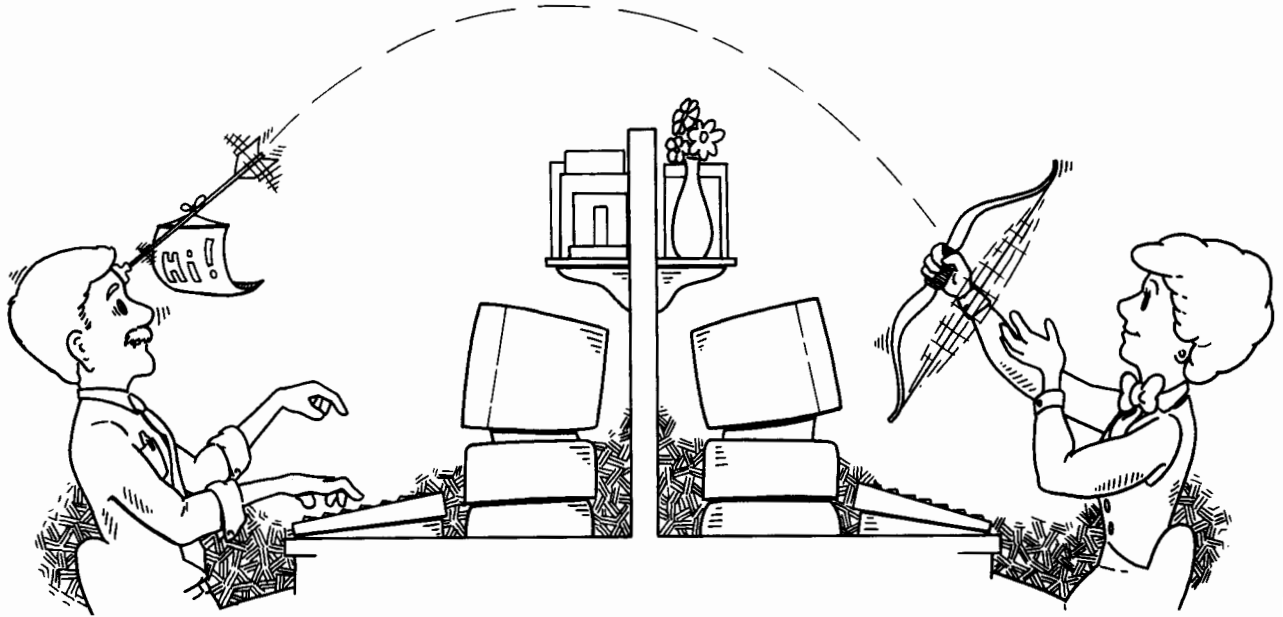
The screen will show the actual time that it got your message, not the 4:47 P.M. shown. The session number will be whatever's accurate. The next number (19 in the example) is one of those computer things... a P.I.N. (Process Identification No.). The I.D. will be your new one.

So, you can send the Operator a message at any time. You could say: "TELLOP I'm printing another file. Give me twenty minutes before you pull the plug." Or: "TELLOP Go ahead. I'll do my printing tomorrow." The Operator **MUST** return to the console before starting the dump, so your request will likely be seen.

A Message to Another User

You've seen the list of other users that appears when you type SHOWJOB. Remember, those are all the people who're using the system at the moment you ask for the information.

By using the TELL command, you can send an individual message to any one of those people, except the ones with the word "QUIET" showing in the third column. (That'll be covered a bit later.)



Try one. Pick somebody from the SHOWJOB list that you've gotten to know, or ask another nearby user if they mind having their work interrupted while you experiment. With that permission, check the SHOWJOB list for their session number, and use that as the "address" for your message. Type it carefully, like this:

```
TELL S24 A NOTE TO YOU. Return
```

Go over to the other user's terminal to see what happened when you did that. It should look like this:

```
FROM/S138 PAT.FINANCE/ A note to you.  
  ↑ (your session/i.d.)    ↑ (your message)
```

You can also command the other user's terminal to make a small noise to attract their attention, if you think they may not notice your message when it arrives. To do this, insert a CTRL G between the message and the Return.

Try it.

Type: `TELL S24 WAKE UP, CTRL G Return`

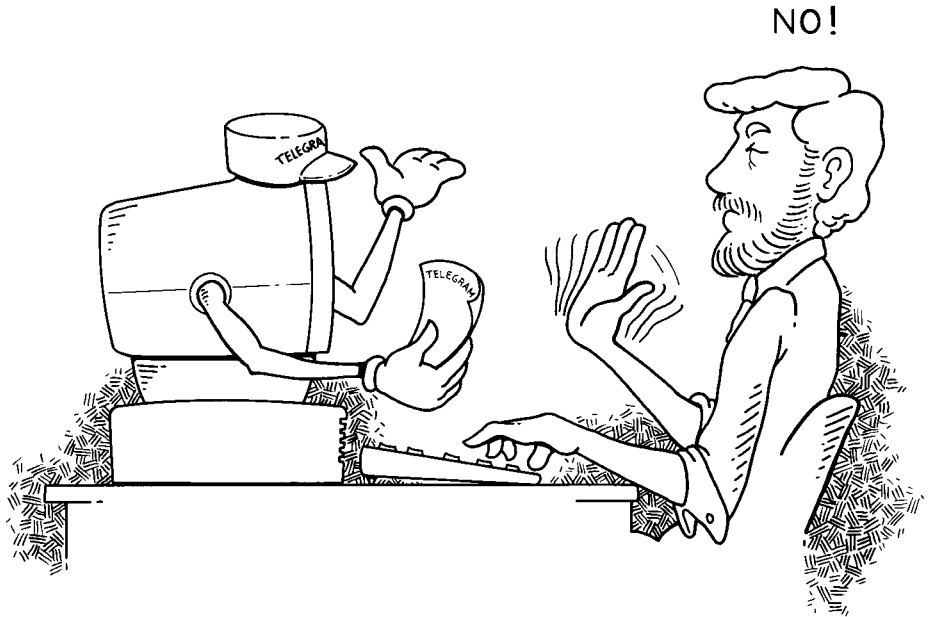
You'll hear a small noise on your own terminal as you press the CTRL G combination.

Ask the other user if they heard the message arrive. Ask if they'll do the same for you, so you can hear it on your own terminal.

As a matter of interest, the Operator also uses this TELL command to send you... and others... individual messages.

Refusing Messages

There will be times when a message would distract you badly, or would make a mess of some project on your screen.



On those occasions, you can use the command SETMSG to turn your receiving ability off and on.

This also results in adding a "QUIET" to the information about your session in the SHOWJOB listing. It warns people that, at least temporarily, you can't be contacted in this way, so they needn't bother trying. Not even the Operator can reach you through a QUIET setting.

To do it, type: **S E T M S G O F F** Return

Check it with: [SHOW JOB](#) [Return](#)

Is there a QUIET on the line with your session number and identity? Good. It'll stay there until you cancel... with SETMSG ON again... or until you log off and, then, back on again.

To test how it works, see if you can borrow another user's terminal for another experiment.

If you can, type: `TELL S138 HELLO, THERE, Return`

It should bounce back, looking like this:

```
:tell sl38 Hello, there,  
SL38 PAT.FINANCE NOT ACCEPTING MESSAGES.  
:
```

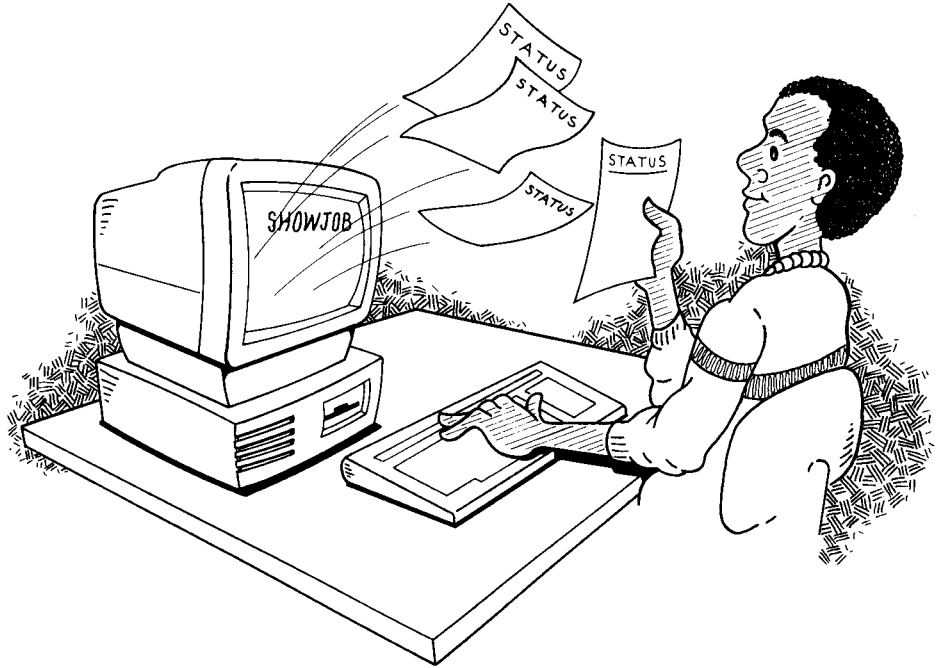
To return YOUR terminal to normal, remember to either log off, or toggle it with: `SETMSG ON` Return

Then, make sure of it with: `S H O W J O B` `Return`

Status Messages

The computer gives you a lot of status information in at least a couple of ways. Some it furnishes routinely, in answer to conditions that change. Other kinds are only furnished if, and when, you ask for them; the "SHOW" commands are a case in point. You've seen both kinds in earlier paragraphs.

Example: using SHOWJOB gives you status information in specific batches, in response to a specific request.



S138 PAT.FINANCE NOT ACCEPTING MESSAGES is an example of the other kind: part of an ongoing stream of information that the computer supplies to you... letting you know the status of your session, your project, your account, your co-workers, and of the operating system and the computer, itself. You've seen dozens of examples, already.

The “SHOW-” Status Commands

You remember these. They’re SHOW- commands that result in a useful report to you by the computer. They can be as short as a line, or as long as pages. They can change minute by minute, or month by month.

- SHOWTIME gives the day, date and time.
- SHOWME provides your own personal status.
- SHOWJOB tabulates current jobs & sessions.
- SHOWCATALOG gives current user, account & system UDC’s.
- SHOWOUT reports the status of output device files.



Next come some more SHOW- commands that are new to you. These, also, produce computer reports that may be helpful in a somewhat more technical fashion. You may not, yet, have enough computer knowledge for these commands to be immediately useful, but you almost certainly will have an idea of what they are.

- SHOWALLOW lists commands permitted globally and personally.
- SHOWCOM gives the status of a communication device.
- SHOWDEV provides the status of input/output devices.
- SHOWIN shows the status of input device files.

See what you can learn by typing the commands themselves, one at a time, and watching what happens after each. Try starting with:

```
S H O W A L L O W Return
```

If any of them look interesting, you might try learning a bit more about them with the HELP command; for instance,

typing: `HELP SHOWALLLOW` Return

Non-SHOW-Status Commands

There are still other commands for getting status information upon request. These are NOT members of the SHOW- family. This trio is typical.

- LISTF tabulates your current files.
- REPORT shows how much file space, CPU & connect time your account's groups have used.

LISTF is the now-familiar way to find out what your most recent list of file names looks like. It's used in MPE, after the colon prompt (:), and in other places.

REPORT lets you know how much storage space and computer time have been used, cumulatively, by your group and your account. If your organization has some sort of chargeback system for computer time, this helps keep track of it.

The command only gives you as much information as you have authority for. If you're an account manager, you'll get summaries for every group in the account. If you're a user, you'll get a status report just for your group, and account totals.

For an account manager, the report might start like this:

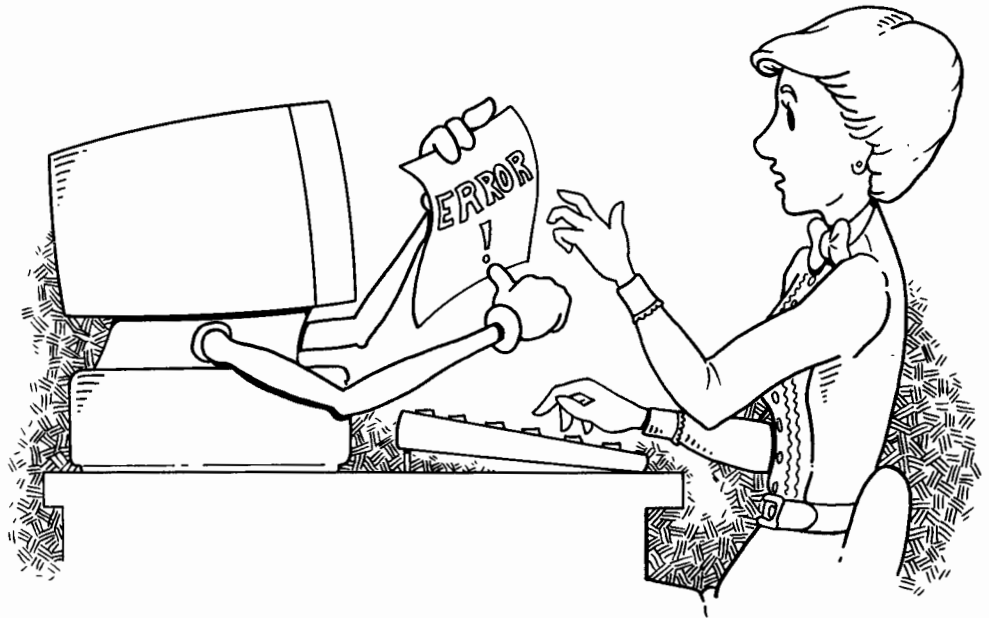
```
:report
ACCOUNT      FILESPACE-SECTORS  CPU-SECONDS  CONNECT-MINUTES
  /GROUP      COUNT    LIMIT    COUNT    LIMIT    COUNT    LIMIT
FINANCE       3958      **    2778      **    78501      **
  /PAYROLL    1810      **     770      **    21159      **
  /ACCTRCVD    1151      **    1185      **    47618      **
  /INVENTORY   997       **     823      **     9724      **
```

Notice that the numbers at the top are totals, and those in the column below are the contributing values. Organizations that charge their departments for the computer time they use find this accounting tool helpful.

The numbers in this report are updated every time you log off, and then back on again. If you don't log off for a week, they'll stay the same. If you log off and on again every ten minutes, that's how often they'll change. Try it, and see!

“Error” Messages

The computer produces some actual error messages. They usually mean failures of some major kind, and they appear on the screen of the console, where the Operator can see and cope with them.



You won't often see them.

Then, there's a bunch of things that LOOK like error messages, but really aren't. In fact, they look enough like them to intimidate users who see them. And you probably WILL see these.

There are hundreds of them, and they'll look like these typical samples:

```
:editoe
^
UNKNOWN COMMAND NAME. (CIERR 975)
.....
:reply
^
REPLY HAS AT LEAST TWO PARAMETERS. (CIERR 3048)
.....
:run listdir5
^
PROGRAM FILE LISTDIR5.PUB.FINANCE NOT FOUND. (CIERR 622)
.....
:helkp
EXPECTED HELLO, :JOB, :DATA, OR (CMD) AS LOGON. (CIERR 1402)
```

First: they AREN'T errors! They simply appear on screen whenever the computer doesn't know what to do next, but can't manage to think of the right question to ask you.

Second: they're all the information you're going to get, until you react by doing something. So, experiment!

IMPORTANT:

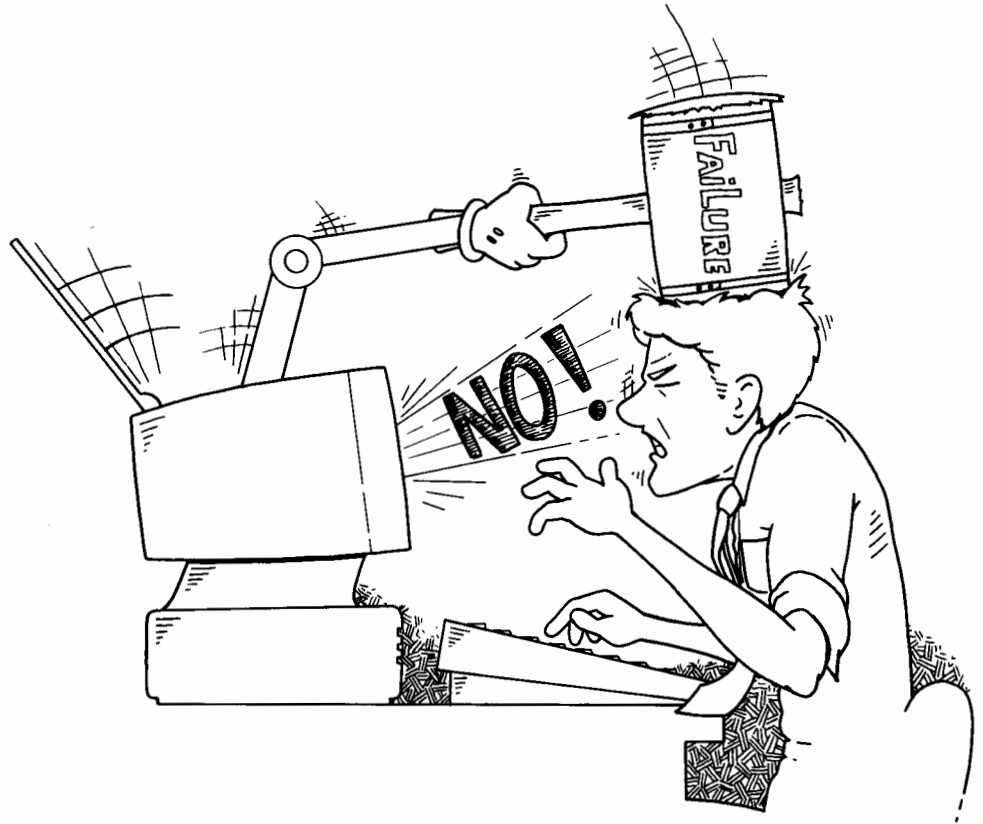
They seem to be referring you to a place where you'll find more information (see entry 975 in the C.I.ERROR TABLE in some reference book somewhere).

This is misleading. There is no C.I.ERROR TABLE with more details on what you have to do next. You'll just have to figure it out for yourself, or ask questions.

“Failure” Messages

Then there are messages that make it seem as though you’ve done something really terrible... or even illegal!

When things like this appear on your screen, they can give you a real impression of having either damaged something, or caused alarm bells to ring down at the police station.



```
:run vdx
ILLEGAL CAPABILITY (LOAD ERR 39)
UNABLE TO LOAD PROGRAM TO BE RUN (CIERR 625)
```

```
.....
/t partful
```

```
-F-I-L-E--I-N-F-O-R-M-A-T-I-O-N--D-I-S-P-L-A-Y
  ERROR NUMBER: 52      RESIDUE: 0
  BLOCK NUMBER: 0          NUMERIC: 0
```

```
*23*FAILURE TO OPEN TEXT FILE (52)
NONEXISTENT PERMANENT FILE (FSERR 52)
```

```
/
```

```
.....
LOCKWORD: PARTFULL.GROUP.ACCOUNT?
```

```
-F-I-L-E--I-N-F-O-R-M-A-T-I-O-N--D-I-S-P-L-A-Y
  ERROR NUMBER: 92      RESIDUE: 0
  BLOCK NUMBER: 0          NUMERIC: 0
```

```
*23*FAILURE TO OPEN TEXT FILE (92)
LOCKWORD VIOLATION (FSERR 92)
```

```
/
```

IMPORTANT:

These warnings often make it sound as though you're just a step or two away from absolute disaster. The truth is, mostly, that you're NOT. So, don't let them worry you. Just stop and think.

These messages usually mean nothing more serious than: you've asked the computer to do something it's not able, or not allowed, to do. That means it's probably going to just sit there, very unhelpfully, until you figure out a more accurate way of asking for what you want. Some of the possibilities are:

- spelling more accurately;
- being more specific in the command or file name;
- using better wording in your action request;
- checking whether there are any file contents;
- checking on the correct lockword; and
- trying a different sequence.

The most important thing to remember, of course, is that you probably haven't done anything "terminal", and you can almost certainly get back on track with a bit of experimenting.

Remember to use the HELP command, and any other computer-originated messages that might get you out of binds.

Looking Back

Now, on the subject of the computer as a message center, you might be interested in a last quiz.

1. Does an "ERROR" message mean you've made a mistake?
2. What's the meaning of the word "QUIET" that can appear beside your session number in a SHOWJOB listing?
3. What would you type to make it appear?
4. Do status commands all start with "SHOW-''?
5. How does the Operator send a message to you, alone?

ANSWERS

1. No. It's just reporting that a condition exists.
2. A QUIET session can't receive messages that are sent by the TELL command.
3. Type: `SETMSG OFF` Return
4. No. Remember: LISTF and REPORT?
5. With the TELL command... not the TELLOP command.

Index

- A**
- ACCOUNT account, 5-1
 - Account manager, 5-17
 - Accounts, what they are, 5-1
 - ADD, EDITOR command, 2-3
 - Adding files to files, 2-23
 - Additional EDITOR commands, 2-37
 - Addresses,
 - file, 5-22
 - default, 5-23
 - Application for sorting, 4-35
 - Areas of the keyboard, 3-2
 - Arrow key, hollow, 3-16
 - Arrow repeater keys, 3-12
-
- B**
- Backspace corrections, 1-7
 - Block mode, testing, 3-8
 - Building files, 2-5, 4-4
 - BUILD, user command, 4-4, 4-27
 - BYE, user command, 1-11
-
- C**
- Capital letters key, 3-26
 - Caps key, 3-26
 - Cartridge tapes, 4-24
 - Changing a file's name, 4-10
 - Character manipulation, 3-17
 - Chevron prompts, 4-7, 4-30
 - Clear display key, 3-17, 3-20
 - Clear line key, 3-17, 3-19
 - Clearing the area, 2-18
 - Closing ADD (//) EDITOR command, 2-4
 - Colon prompts, 1-6
 - Column setup, 4-30
 - Combining files, 2-19, 2-23

- Command
 - abbreviations, EDITOR, 2-35
 - dictionaries, 2-12, 4-19, 4-22
 - 'families', EDITOR, 2-37
 - 'families', FCOPY, 4-8
 - sets, UDC's 5-27
 - sets, status, 6-8, 6-9, 6-10
 - sets: EDITOR, 2-37
- Commands,
 - a discussion, 1-21
 - defining them, 2-12, 4-19, 4-22
 - EDITOR, //, 2-4
 - EDITOR, ADD, 2-3
 - EDITOR, COPY, 2-30, 2-32
 - EDITOR, END, 2-4
 - EDITOR, GATHER, 2-26, 2-28
 - EDITOR, JOIN, 2-19, 2-23
 - EDITOR, KEEP, 2-5
 - EDITOR, LIST, 2-9
 - EDITOR, TEXT, 2-9
 - EDITOR, XPLAIN, 2-12
 - EDITOR, definitions, 2-12
 - FCOPY, FROM, 4-9
 - FCOPY, TO, 4-9
 - MPE, LISTUSER, 5-17
 - SORT, END, 4-31, 4-33
 - SORT, INPUT, 4-31, 4-33
 - SORT, KEY, 4-31, 4-33
 - SORT, OUTPUT, 4-31, 4-33
 - UDC's, 5-27
 - user, BUILD, 4-4, 4-27
 - user, BYE, 1-11
 - user, definitions, 4-19, 4-22
 - user, EDITOR, 2-2
 - user, FCOPY, 4-8
 - user, FILE, 4-15, 4-25
 - user, HELLO, 1-7, 1-10
 - user, HELP, 4-19, 4-22

- user, LISTE, 2-7, 6-10
- user, PURGE, 4-12, 4-35
- user, REDO, 1-15
- user, RENAME, 4-11, 5-15
- user, REPORT, 6-10
- user, RESTORE, 4-26
- user, RUN, 4-30, 4-33
- user, SETCATALOG, 5-31, 5-36
- user, SETMSG, 6-6
- user, SHOWALLOW, 6-9
- user, SHOWCATALOG, 5-28, 5-32, 6-9
- user, SHOWCOM, 6-9
- user, SHOWDEV, 6-9
- user, SHOWIN, 6-9
- user, SHOWJOB, 5-6, 6-9
- user, SHOWME, 1-20, 6-9
- user, SHOWOUT, 6-9
- user, SHOWOUT SP, 4-17
- user, SHOWTIME, 1-14, 6-9
- user, STORE, 4-25
- user, TELL, 6-4
- user, TELLOP, 6-2
- using one letter, 2-35

Computer

- control key, CTRL, 3-29
- files, 5-2
- files, a description, 4-1
- filing cabinet, 4-1, 5-2
- system diagram, 1-3
- terminals, 1-3

- Config keys window, 3-11
- Console, 5-11
- Contents of file, checking, 4-6
- Control key, CTRL, 3-29
- COPY, EDITOR command, 2-30, 2-32
- Copying
 - files, FCOPY, 4-7
 - text lines, 2-30, 2-32
- Corrections,
 - using 'Backspace', 1-7
 - using REDO, 1-15
- Creating
 - a PUB file, 5-20
 - files, 2-5, 4-4, 4-27
- CTRL key, 3-29
- Cursor,
 - description, 1-5
 - use in correcting, 1-17

D

- Date sorts, 4-33
- Default file addresses, 5-23
- Defaults, what are they, 2-36
- Define fields window, 3-11
- Defining EDITOR's commands, 2-12
- Defining user commands, 4-19, 4-22
- Del/Esc 'Escape' key, 3-30
- Delete char(acter) key, 3-17
- Delete line key, 3-17, 3-20
- Deleting
 - letters, 1-15, 2-15
 - lines, 2-25, 3-17, 3-20
 - many lines, 2-25, 3-20
 - your work area, 2-20
- Device control window, 3-11
- Dictionary
 - of EDITOR commands, 2-12
 - of user commands, 4-19, 4-22

-
- E** Editing keys, 3-17
EDITOR commands,
 //, 2-4
 ADD, 2-3
 COPY, 2-30, 2-32
 definitions, 2-12
 END, 2-4
 GATHER, 2-26, 2-28
 JOIN, 2-19, 2-23
 KEEP, 2-5
 LIST, 2-9
 TEXT, 2-9
 XPLAIN, 2-12
EDITOR
 subsystem, 2-2
 one-letter commands, 2-35
 the user command, 2-2
END,
 the EDITOR command, 2-4
 the SORT command, 4-31, 4-33
Enhance video window, 3-11
Enter/Print key, 3-32
Erasing files, 4-12, 4-35
Error messages, 6-12
Escape key, 3-30
Explaining EDITOR commands, 2-12
Explaining user commands, 4-19, 4-22

- F**
 - f4 key 'Remote Mode', 1-5
 - Failure messages, 6-14
 - FCOPY commands,
 - FROM, 4-9
 - TO, 4-9
 - FCOPY subsystem, 4-8
 - FCOPY, user command, 4-8
 - File
 - ABC, 4-25
 - AJEUDC, 5-30, 5-32
 - Coffee, 5-20, 5-25
 - Jones, 2-20, 4-8, 4-15, 4-25, 5-15, 5-23
 - Listalfa, 4-27, 4-31, 4-35
 - Listdate, 4-27, 4-33, 4-35
 - Listraw, 4-29, 4-31, 4-33, 4-35
 - locks, 2-20, 5-15
 - Name1, 4-3, 4-4, 4-6
 - Name2, 4-5, 4-8, 4-10
 - Name3, 4-11
 - names, 4-3
 - Pat, 2-5, 2-9, 2-18, 3-12
 - File contents,
 - checking, 4-6
 - shuffling, 4-27
 - FILE, user command, 4-15, 4-25
 - Files,
 - added to files, 2-19, 2-23
 - copying them, FCOPY, 4-8
 - creating them, 2-5, 4-4, 4-27
 - displaying them, 2-9
 - everybody's, 5-2
 - from other addresses, 5-23
 - getting them back, 2-9
 - information about them, 4-1
 - lists of, 2-7
 - printing them, 4-15

- PUB(lic), creating one, 5-20
- PUB(lic), using them, 5-25
- removing them, 4-12, 4-35
- renaming, 4-10
- restoring from tape, 4-26
- security, 2-20, 5-15
- shuffling data, 4-27
- storing onto tape, 4-25
- the computer's, 5-2
- their addresses, 5-22
- unnumbered, 4-29
- Filing your work away, 2-5
- Fresh colon prompts, 1-6, 1-7
- FROM, FCOPY command, 4-9
- Function control keys, 3-10
- Function keys, 3-6

G

- GATHER, EDITOR command, 2-26, 2-28
- Graphics use of number keys, 3-24
- GROUP group, 5-1
- Group, PUB(lic), 5-18
- Groups, what they are, 5-1, 5-2

H

- Hard reset, 3-4
- HELLO, user command, 1-7
- HELP, user command, 4-19, 4-22
- Helping you with commands, 4-19
- Hollow arrow key, 3-16
- HP 150 Touchscreen ON switch, 1-1
- HP 2392A terminal ON switch, 1-1

I Identifying self (logging on), 1-9
Invisible typing, 3-30
Incremental text line numbers, 2-32
INPUT, SORT command, 4-31, 4-33
Insert
 char(acter) key, 3-18
 letters, 1-16, 2-15
 line key, 3-19
 lines, 2-21

J Jobs, 5-10
JOIN, EDITOR command, 2-19, 2-23
Jumping the work window, 3-13
Jumping to file ends, 3-16

K KEEP, EDITOR command, 2-5
Keeping your work, 2-5
Key f4 'Remote Mode', 1-5
Key for capital letters, 3-26
Key for computer control, CTRL, 3-29
Key for entering data, 3-32
Key for 'Escaping', 3-30
KEY, SORT command, 4-31, 4-33
Keyboard overview, 3-2
Keyboard sectors list, 3-2
Keyboard sectors location, 3-2
Keys for
 display control, 3-12
 editing, 3-17
 function controls, 3-10
 functions, 3-6
 message windows, 3-6
 number entry, 3-22
 shifting, 3-32
 terminal control, 3-3

-
- L**
- LDEV, shown, 1-21
 - Letter
 - deletion, 1-15, 2-15
 - insertion 1-16, 2-16
 - replacement, 1-17, 2-17
 - Line
 - deletion, 2-25, 3-17, 3-20
 - insertion, 2-21, 3-19
 - printers, 1-3, 4-15
 - manipulation, 3-19
 - LIST, EDITOR commands, 2-9
 - LISTF, user command, 2-7, 6-10
 - Listing offline (printing), 4-15
 - Listing your files, 2-7
 - LISTUSER, MPE command, 5-17
 - Local mode/Remote mode, 3-7
 - Locking your files, 2-20, 5-15
 - Lockwords,
 - an introduction, 2-20, 5-15
 - if forgotten, 5-16
 - picking good ones, 5-16
 - remembering them, 5-16
 - Logging off, 1-11
 - Logging on, 1-9
-

- M**
- Making file copies, FCOPY, 4-8
 - Manipulating characters, 3-17
 - Manipulating lines, 3-19
 - Margins/tabs/col window, 3-11
 - Menu key, 3-10
 - Message window
 - 'Remote Mode', 1-5
 - switching, 3-7
 - 'Terminal', 1-5
 - toggle keys, 3-7

Messages
 about 'errors', 6-12
 about 'failure', 6-14
 about help, 1-8
 being refused, 6-6
 network, 6-1
 showing status, 6-8
 to Operator, 6-2
 to other users, 6-4
Misspellings, 1-6, 1-16, 2-17
Modems, 4-24
Modes window, 3-11
More EDITOR commands, 2-37
Moving the work window, 3-13
Moving to end of file, 3-16
Moving to start of file, 3-16
MPE user command definitions, 4-19, 4-22
MPE command, LISTUSER, 5-17

N Name sorts, 4-30
Name1 file, 4-3, 4-4, 4-6
Name2 file, 4-5, 4-8, 4-10
Name3 file, 4-11
Naming files, 4-3
Neighbors in the files, 5-5
Nesting UDC's, 5-35
Next/Prev keys, 3-13
Number keys, 3-22
Number keys, graphics use, 3-24

O OFF/ON switch, 1-1
Offline storage, 4-25
Offline listing, 4-15
One-letter commands, 2-35
ON/OFF switch, 1-1
Operator messages, 6-2
Operator, 5-10, 5-18
Other EDITOR commands, 2-37
OUTPUT, SORT command, 4-31, 4-33

P PAM (Personal Applic. Mgr.), 1-4
Passwords, 5-13
Personal Applica. Mgr. (PAM), 1-4
Picking a file name, 4-3
Prev/Next keys, 3-13
Print/Enter key, 3-32
Printers, 1-3, 4-15
Printing a file, 4-15
Printouts, yours, 4-18
Privacy for files, 2-20, 5-15
Prompts,
 chevron, 4-7, 4-30
 colon, 1-6
 slash, 2-3
PUB(lic) files,
 creating one, 5-20
 using them, 5-25
PUB(lic) groups, 5-18
PURGE, user command, 4-12, 4-35
Purging a file, 4-12, 4-35

R REDO subcommands, 1-15, 2-15
REDO, user command, 1-15
Reels, tape, 4-24
Remote mode/Local mode, 3-7
RENAME, user command, 4-11, 5-15
Renaming files, 4-11
Renumbering text lines, 2-26
Reorganizing file contents, 4-27
Replacing letters, 1-17, 2-17
REPORT, user command, 6-10
Resequencing lines in EDITOR, 2-27
Reset, hard, 3-4
Reset, soft, 3-3
Reset/Break key, 3-3
Restart scrolling, 4-21
RESTORE, user command, 4-26
Restoring files, 4-26
Return, first use, 1-6
Return, for fresh colon, 1-7
RUN, user command, 4-30, 4-33

S Saving your work, 2-5
Scrolling
 the work window, 3-13
 and controlling it, 4-21
Security, 2-20, 5-12, 5-15
Service keys window, 3-11
Session, status report, 1-20, 6-9
Sessions, 1-11, 5-8
SETCATALOG, user command, 5-31, 5-36
SETMSG, user command, 6-6
Set time window, 3-11
Setting up
 Margins/tabs, 3-11
 Screen appearance, 3-11
 The time, 3-11

- Shift keys, 3-32
- SHOWALLOW, user command, 6-9
- SHOWCATALOG, user command, 5-28, 5-32, 6-9
- SHOWCOM, user command, 6-9
- SHOWDEV, user command, 6-9
- SHOWIN, user command, 6-9
- SHOWJOB, user command, 5-6, 6-9
- SHOWME, user command, 1-20, 6-9
- SHOWOUT, user command, 6-9
- SHOWOUT SP, user command, 4-17
- SHOWTIME, user command, 1-14, 6-9
- Shuffling
 - by date, 4-33
 - by name, 4-30
 - file contents, 4-28
- Single-letter EDITOR commands, 2-35
- Slash prompt, 2-3
- Soft reset, 3-3
- SORT commands,
 - END, 4-31, 4-33
 - INPUT, 4-31, 4-33
 - KEY, 4-31, 4-33
 - OUTPUT, 4-31, 4-33
- SORT subsystem, 4-30
- SORT.PUB.SYS program, 4-30, 4-33
- Sorting,
 - by date, 4-33
 - by name, 4-30
 - how to use it, 4-35
- Start scrolling up again, 4-21
- Start up, 1-1
- Status commands,
 - SHOWJOB, 5-6, 6-9
 - SHOWME, 1-20, 6-9
 - SHOWOUT SP, 4-17
 - SHOWTIME, 1-14, 6-9
- Status messages, 6-8
- Stop key, 3-3
- Stop scrolling, 4-21
- STORE, user command, 4-25
- Storing files, 4-25

Subcommands, REDO, 1-15
 Subsystems,
 EDITOR, 2-2
 FCOPY, 4-8
 SORT, 4-30
 SYS(TEM) account, 5-10

T Tab setup, 3-11
 Tape,
 cartridges, 4-24
 reels, 4-24
 storage of files, 4-25
 TELL, user command, 6-4
 TELLOP, user command, 6-2
 Terminal
 ON/OFF switch, 1-1
 control keys, 3-3
 Terminals, computer, 1-3
 Tests on the subjects of:
 accounts, 5-40
 messages, 6-17
 the first session, 1-23
 the keyboard, 3-33
 using files, 4-37
 word processing, 2-38
 Text, adding to it, 2-10
 TEXT, EDITOR commands, 2-9
 Text lines,
 by increments, 2-32
 copying, 2-30, 2-32
 deletion, 2-25
 relocating, 2-27, 2-28
 renumbering, 2-26
 Time savers in EDITOR, 2-34
 Time, by command, 1-14, 6-9
 TO, FCOPY command, 4-9
 Touchscreen instructions, 1-4
 Turning terminal on, 1-1

-
- U**
- UDC's (User-Defined Commands)
 - account-level, 5-38
 - creating a UDC file, 5-30
 - creating them, 5-31
 - definition, 5-27
 - discussions, 5-11, 5-27
 - nesting them, 5-35
 - system-level, 5-39
 - testing them, 5-33, 5-38
 - those available, 5-28
 - user-level, 5-29
 - Unnumbered files, 4-29
 - USER.ACCOUNT, GROUP log on, 1-9
 - USER.ACCOUNT, GROUP, 1-20, 5-1
 - User commands,
 - BUILD, 4-4, 4-27
 - BYE, 1-11
 - definitions, 4-19, 4-22
 - EDITOR, 2-2
 - FCOPY, 4-8
 - FILE, 4-15, 4-25
 - HELLO, 1-7, 1-10
 - HELP, 4-19, 4-22
 - LISTF, 2-7, 6-10
 - PURGE, 4-12, 4-35
 - REDO, 1-15
 - RENAME, 4-11, 5-15
 - REPORT, 6-10
 - RESTORE, 4-26
 - RUN, 4-30, 4-33
 - SETCATALOG, 5-31, 5-36



SETMSG, 6-6
SHOWALLOW, 6-9
SHOWCATALOG, 5-28, 5-32, 6-9
SHOWCOM, 6-9
SHOWDEV, 6-9
SHOWIN, 6-9
SHOWJOB, 5-6, 6-9
SHOWME, 1-20, 6-9
SHOWOUT, 6-9
SHOWOUT SP, 4-17
SHOWTIME, 1-14, 6-9
STORE, 4-25
TELL, 6-4
TELOP, 6-2
User list, 5-6
USER user, 5-1
User-Defined Commands (UDC's), 5-11, 5-27
User/System key, 3-10
Users, other, 5-5
Users, what they are, 5-1

W Word processing description, 2-1
Work window, moving it, 3-13, 3-14, 3-16
Writing text, 2-3

X XPLAIN, EDITOR command, 2-12

Y Your files,
 adding to them, 2-10
 are they there? 2-7
Your printouts, 4-16

Appendix A

Creating The USER.ACCOUNT,GROUP Log On

USER.ACCOUNT,GROUP is a “skeleton key” identity that a person can use for learning about the system before they have a personal log on identity of their own. Creating it requires the following three-part process.

IMPORTANT:

You must have System Manager (SM) capability to accomplish this.

Part 1—Creating Account And Account Manager

If you're not already using the computer, log on and wait for the colon prompt (:) to appear.

Type: `NEWACCT ACCOUNT,MGR Return`

This creates the ACCOUNT account, and one user who is the account manager. You must specify an account manager in order to create the USER and the GROUP within the account. Beyond that, you probably won't find much use for MGR.ACCOUNT.

Part 2—Creating The Group

Log on by typing: `HELLO MGR.ACCOUNT Return`

Wait for the colon prompt (:) to appear.

Type: `NEWGROUP GROUP Return`

Verify that the ACCOUNT account now has the proper two groups: PUB, which is automatically created in any new account, and GROUP, which you just added.

Type: `LISTGROUP @ Return`

If only the PUB group is listed, repeat Part 2. If both groups are listed, go on to Part 3.

Part 3—Creating The User

Type: `NEWUSER USER;CAP=IA,BA,ND,SF Return`

This gives anyone who logs on as USER.ACCOUNT these four basic capabilities:

- to conduct computer sessions (IA capability);
- to give jobs to the computer (BA capability);
- to use computer devices like terminals, disc drives, tape drives, and printers (ND capability); and
- to save files (SF capability).

Verify that the ACCOUNT account now has two users, MGR and USER, by typing: `LISTUSER @ Return`

If both MGR and USER are listed, you've been successful in creating the USER.ACCOUNT,GROUP skeleton key.

Test it by typing: `HELLO USER.ACCOUNT,GROUP Return`



Printed in U.S.A. 1/86
32033-90009