MEMOTECH FLOPPY DISC SYSTEM

NewWord™

NewWord™

NewWord

on the

MEMOTECH FDX

Congratulations on your new **Memotech FDX** Disc Drive System.

An integral part of any serious Disc System must be a Word Processor Package. We had the oppurtunity at **MEMOTECH** to look at a wide variety of Word Processor Packages. We spent a great deal of time learning to use each one, trying to find out what each offered, and what advantages one package might have over another.

Eventually after a great deal of thought and "hands on" experience, we decided to go with **NewStar's NewWord.** We found it to be quite sophisticated and yet so simple to use.

NewWord is on one of the discs that came with your **Memotech FDX** System. To get the most out of **NewWord,** we suggest that you go through the **READ-ME** file on the disc. In fact the following pages contain a print out of the contents of READ-ME, MEMOTECH.SUP and NEWWORD.SUP files. So it will most probably be easier for you to read through the print outs first. After you have read over READ-ME, then we suggest that you go on and use **NewWord,** while at the same time reading the **"Do It Yourself"** manual.

Good Luck!

NWINSTAL.COM	Installation program. It will prompt you through installing Newword for your terminal and printer. You also personalize **NewWord** with this program.
NWU.COM	**NewWord:** Pre-installed for a **Memotech FDX** with a monochrome screen and draft printer.
NW.OVR	**NewWord** overlays.
NWPRINT.OVR	**NewWord** printer drivers.
NWMSGS.OVR	Messages and menus.
PRACTICE.DOC SAMPLE1.DOC SAMPLE2.DOC	Practice files for the Tutorial: Do It Yourself
NWKEY.COM	Use this program to select **NewWord** functions for the function keys F1 through F8 at the righthand side of the keyboard. The file **MEMOTECH.SUP** explains how to use the program.
NWCOLOR.COM	Use this program to change the color for text and background, if you don't like the colors we have chosen. The file **MEMOTECH.SUP** on this disk explains how to use the program.
NEWWORD.SUP	Information about conditional merge printing, features added to **NewWord** since the manual was printed. These are advanced features; if you are just learning to use **NewWord**, you may skip reading this file for now.
MEMOTECH.SUP	Specific information about the **Memotech** version of **NewWord** that is either different from the information in the manuals--or not covered in the manuals. Read this file before starting with the Tutorial: Do It Yourself.
READ-ME	This document.

Tips on printer installation:

NewWord cannot be installed to use your printer unless the operating system can print on it correctly. Auto line feed must be disabled on your printer. To test whether your system can print properly, put a disk containing this file in drive C (varies with the system), ready your printer and enter:

```
PIP LST:=C:READ-ME
```

If this document is printed single-spaced with no characters lost, **NewWord** can be installed to use your printer. If it isn't printed correctly, you must customize **NewWord** to run your printer (see appendix C of the **NewWord** Encyclopedia).

There are several printer drivers in NWPRINT.OVR:

NON-SPECIFIC PRINTER DRIVERS

SIMPLE This driver does not support print enhancements (bold, doublestrike, underline, strike-out). It should only be used if the other drivers won't run your printer, or to make a quick review copy. Printers which advance the paper to print (Printronix, Versatec, Gould, etc) must use this driver.

DRAFT Most dot matrix printers. This driver implements the print enhancements by carriage return and overprinting.

TYPEWR Typewriters and many backspacing printers. Print enhancements are done by backspacing and overprinting. If your printer can backspace and isn't directly supported by one of the other drivers, you should test both this driver and the DRAFT driver to see which prints faster (or better).

PRVIEW This driver outputs to a disk file named PREVIEW.NW on the logged drive. It may be used to preview what your printed output will look like. This is useful for merge-printed documents and headers and footers. If you need to save this output, you must copy it to another file, or rename PREVIEW.NW prior to using the PRVIEW driver again. PREVIEW.NW is not intended to be printed.

WHEEL PRINTERS

D1610 Most Diablo (and Diablo emulating) daisy wheel printers. The Diablo 620(some), 630(some), 1610, 1620, 1640 and 1650 can be run successfully by this driver.

D630 Actually the 1610 printer driver. It will drive most Diablo 620 and 630 daisy wheel printers.

QSPR11 Actually the 1610 printer driver. QUME Sprint 5,7,9 & 11 daisy wheel printers.

SR550 Silver Reed daisy wheel printers.

N2000 NEC 2000 series thimble wheel printers.

N3500 NEC 3500 series thimble wheel printers.

N5500 NEC 5500 series thimble wheel printers.

SPECIFIC DOT MATRIX PRINTERS

MX80
MX100 Epson MX and FX series printers. This driver supports microspace justification for flush right margins, different character widths (.cw n), different line heights (.lh n), sub and superscripts. Microspace justification is done by adding space between words in 1/60 inch increments.

MT160L Mannesmann Tally MT160L dot matrix printer. This driver supports different character widths (.cw n), different line heights (.lh n), sub and superscripts, Text is printed in correspondence mode for 10 and 12 pitch; draft mode for other character widths. Text is NOT microspace justified. Some versions of the MT160L don't eject to top of form properly without using a form-feed character, so test yours, and install Newword the way it works best.

OKI84
OK192
OK193 Okidata microline model 84 (step 2), 92 and 93 dot matrix printers. This driver uses the correspondence mode to produce very high quality output. Print speed is reduced from draft mode. This driver supports microspace justification for flush right margins, different character widths (.cw n), different line heights (.lh n), sub and superscripts. There is a limit of 10 inches of print space for the model 84 printer when using 17 pitch (.cw of 9 or less). Use the draft driver and custom print controls if you wish to print at 17 pitch with more than 170 characters. For the model 92 and 93 printers, cw of 9 or less switches to draft mode and .cw of 15 or more doesn't work accurately. Older models of the 84 and other Oki may not be run successfully by this driver.

C1550 C.Itoh Prowriter models 8510 and 1550. This driver supports microspace justification for flush right margins, different character widths (.cw n), different line heights (.lh n) and sub and superscripts.

You can install which printer driver **NewWord** will use as its default with NWINSTAL. You may always select a different printer when you are ready to print. You might use this to advantage, for example, by installing the DRAFT or SIMPLE printer as the default, then using the enhanced driver for final copies.

NewWord on the Memotech FDX

This file contains specific information about **NewWord** for the **Memotech FDX.** It is intended to be a supplement to the **NewWord "Do It Yourself."** Section 1 discusses the video display; section 2 discusses the keyboard and should be read in conjunction with Chapter 1 of the **"Do It Yourself";** section 3 discusses loading **NewWord.** Read this before reading Chapter 2 of the **"Do It Yourself."** Finally, section 4 discusses the use of the programs NWCOLOR.COM, which is used to change the colors displayed on the color monitor; and NWKEY.COM, which is used to program the function keys on the lefthand side of the keyboard.

To obtain a printout of this file, first make sure your printer connected and the disk containing this file is in Drive C:. At the Operating System prompt, enter:

PIP LST:=C:MEMOTECH.SUP

(Note: this is the same procedure used to print out the READ-ME file.)

1. The Video Display

On a monochrome monitor, **NewWord** displays bold and doublestrike highlighted onscreen. Subscript, superscript, and strikeout text is displayed against a half intensity background. Underlined text is displayed with an underline. Marked blocks are displayed in reverse video.

On a color monitor, normal text is displayed white on blue. Bold and doublestrike are yellow. Subscripts are magenta. Superscripts are red. Strikeout is green. And underline is black.

NWINSTAL can be used to select either monochrome or color for the **Memotech FDX** from the Terminals Menu.

Color **NewWord** will work on a monochrome screen, and vice versa. However, the screen will usually look very strange and you may have to reset the computer to restore the screen to normal.

The color of the background and text can be changed with the program NWCOLOR.COM. See section 4.

2. The Keyboard

Chapter 1.6 of the **Do It Yourself** discusses some of the special keys on computer keyboards. This discussion is still pertinent. The **Memotech** keyboard has several function keys on the

numeric keypad that can be used instead of the normal **NewWord** commands. As you read through the **Do It Yourself**, especially Chapter 3, and learn the **NewWord** commands that move the cursor and scroll the text onscreen, you might refer to this section of the addendum to learn which function key can be used as an alternative. For example, the normal **NewWord** command to move the cursor one character to the right is ^D (hold the CTRL key and the "D" key down at the same time). On the **Memotech** keyboard, the "right arrow" (located on the "3" key in the numeric keypad) performs the same function.

The text of the **Do It Yourself** does not talk about the function keys specifically. It will be up to you to learn the function keys.

KEY	WHAT IT DOES
arrows	Move cursor up, down, left and right. (^E, ^X, ^S, ^D)
HOME	Moves the cursor to the top of the document. (^QR)
INS	Turns insert mode on or off. (^V)
CLS	Cancel or un-erase. (^V)
TAB	Moves to next tab stop. (^I)
DEL	Deletes the character at the cursor. (^G)
PAGE	Finds a specific page number. (^QP)
EOL	Erases from the cursor to the end of the line. (^QY)
BRK	Turns bold text on or off. (^PB)
F1	Save a file, then resume editing. (^KS)
F2	Save a file, return to CP/M. (^KX)
F3	Center text at cursor. (^OC)
F4	Margin release. (^OX)
F5	Save, return to Opening Menu. (^KD)
F6	Abandon editing, return to Opening Menu. (^KQ)
F7	Scroll screen up. (^R)
F8	Scroll screen down. (^C)
shift+F1	Change left margin. (^OL)
shift+F2	Erase word at cursor. (^T)
shift+F3	Erase line at cursor. (^Y)
shift+F4	Search for text. (^QF)
shift+F5	Change right margin. (^OR)
shift+F6	Align paragraph. (^B)
shift+F7	Set a tab stop. (^OI)
shift+F8	Find and replace text. (^QA)

You may reconfigure the keys labeled F1 through F8 to anything you find useful with the program NWKEY.COM. See section 4.

3. Loading NewWord

Chapter 2 of the **Do It Yourself** was written specifically for CP/M in general. However, on the **Memotech FDX**, drives A and B are one and the same. Drive C is the second drive. Therefore, every reference to drive B should be replaced with drive C.

It will probably not be necessary for you to use the **NewWord** installation program, NWINSTALL.COM, as outlined in Chapter 2, because **NewWord** has been shipped to run immediately on a **Memotech FDX** with monochrome screen. So, when you get to Chapter 2.12, skip that section, and enter these commands instead:

```
pip c:nw.com=nwu.com
```

If you do not want to program the function keys on the righthand side of the keyboard or change the color of the video display, go on to section 2.13. Otherwise, go on to section 4 of this addendum before going on to section 2.13 of the Do It Yourself.

4. Changing colors and programming the function keys

If the colors we have chosen for the background or for the text (on the color monitor, of course) do not please you, you may change them. On your distribution disk is the program NWCOLOR.COM. Put the distribution disk in Drive C:, and put your Newword working disk in Drive A:. (This is the state of affairs as described above at section 3e.) Enter the following:

```
c:<cr>
nwcolor a:nw.com<cr>
```

The text on the screen explains how to use the program. We will not repeat those instructions here.

As can see, there are seven "types" of text that **NewWord** displays on the monitor. Normal text is by far the most common. We recommend that you choose text and background colors that will be easy on the eyes.

To move the cursor, use the arrow keys on the numeric keypad.

Pay careful attention to the example display shown on screen, because changing the color in one place might have an effect on the way some other kind of text is displayed. (For example, if you change the background color to blue for normal text, notice that the background color changes for all other types of text. You may chose any combination of text and background color that you want for all seven types of text.

Type ^C (hold down the CTRL key and "C" at the same time).

Unless you now want to program the function keys, go on to section 2.13 of the Do It Yourself.

To program the function keys on the left side of the keyboard, enter the following:

nwkey a:nw.com<cr>

The text on the screen explains how to use the program. We will not repeat those instructions here.

There are eight function keys. Programming a function key means specifying which **NewWord** command (or series of commands) will be done when you press the function key. For example, function key F1 is pre-programmed for the **NewWord** command ^KS, which is used to save your file and then resume editing.

Holding down the shift key and pressing a function key is different from just pushing the function key alone. It is possible to program up to sixteen **NewWord** commands on the function keys.

The amount of memory available for storing the **NewWord** commands associated with each function key is limited. Watch the display at the bottom of the function key table on screen that tells how much space is remaining.

When you are finished programming the function keys (and the monitor colors), go on to section 2.13 of the **Do It Yourself.**

Advanced NewWord on the Memotech FDX

This file contains information about **NewWord** features that are new for the **Memotech FDX**, and are not explained in the **NewWord "Do It Yourself"** or the **"Newword Encyclopedia."** These are advanced features. If you are just learning how to use **NewWord**, you do not need to read this now. Just remember where the information is.

To obtain a printout of this file, you may use the "PIP" program as outlined in the read-me file on this disk. You may also print it out with **NewWord**.

1. "Wildcard" option for find and find/replace

When you get to Chapter 6 of the **Do It Yourself**, come back and read this section.

You may, in response to the Find what? prompt, enter question marks (?) in the word or phrase you want to look for. Then, if you select the question mark option at the next prompt, **NewWord** will treat the question marks in the original word or phrase as "wildcards" that match any character in the text. Look at the following examples:

????? matches the next five characters in the document, no matter what the characters are.

m?n matches any occurrence of "m" and "n" that have any character between them, such as "man" "men" or even "m3n."

&? matches any character preceeded by an ampersand. (See Chapter 11 of the **Do It Yourself.**) If you do not specify the question mark option, then **NewWord** will find only "?????", "m?n" and "&?."

2. Conditional merge-print dot commands

Use the following discussion in addition to the information in Chapter 11 of the Do It Yourself.

NewWord's conditional merge-print dot commands allow you to use the same merge-print document to produce letters that are different, depending on criteria you select. For instance, you can use one document and one data file to generate letters with language specific for the region of the country the letter is going to, or specifically for the client addressed.

We use conditionals every day: If the telephone rings, we answer

it; if the boss is away, we play. Notice that what we do in these examples is dependent on certain conditions, namely, the phone's ringing and the boss' whereabouts. We evaluate the conditions. If the conditions are met, we answer the phone or play. If the conditions are not met, we do something else. These kinds of statements are useful in helping us decide what to do in specific situations.

When you design a document for merge-printing, it would be helpful to have a means for having the printout appear a certain way depending on certain conditions. For example, suppose you want to include a certain paragraph in your letters if the addressee has a dog, but want a different paragraph included if the addressee has a cat. The merge-print dot commands will allow you to do this with one merge-print document and one data file.

You might state the task in this way:

> If the data item in the variable PET is the same as "dog," insert the file called "dog.doc." If the data item is the same as "cat," insert the file called "cat.doc."

Once you have gotten what you want to do figured out like this, the dot commands to perform the task are:

```
.IF &PET& = dog
.FI dog.doc
.EI
.IF &PET& = cat
.FI cat.doc
.EI
```

The .IF and .EI dot commands

NewWord determines if a condition is met usually by comparing one item with another. The items that **NewWord** compares may be data items taken from a data file, or items that never change (that is, constants). Typically, one item is a data item and the other is a constant, but any combination can be used.

NewWord is able to compare items in the following ways:

1. Are they the same or equal?

2. Does one come before or after the other in alphabetical or numerical order?

3. Are they not the same or equal?

You tell **NewWord** how it is supposed to compare items by using one of the following operators (this is computer jargon for the symbols that follow).

Symbol	Meaning
=	is the same as
>	comes after (alphabetically)
<	comes before (alphabetically)
<=	comes before (alphabetically) or is the same as
>=	comes after (alphabetically) or is the same as
<>	is not the same as
£=	is equal to (for numbers)
£>	is greater than (for numbers)
£<	is less than (for numbers)
£>=	is greater than or equals (for numbers)
£<=	is less than or equals (for numbers)
£<>	is not equal to (for numbers)

The most common way of specifying a condition to be evaluated in a .IF command is the following:

.IF item operator item

Recall the example of the cat or dog letter. The statement was: If the data item contained in the variable PET is the same as "dog," insert the file called dog.doc. Translating the first clause into a merge-print dot command is relatively easy:

.IF &PET& = dog

The second clause translates into a dot command that inserts a file:

.FI dog.doc

Just as every sentence must have a period to mark its end, every conditional merge-print "sentence" must have a .EI command to mark its end. So, our example looks like this altogether:

```
.IF &PET& = dog
.FI dog.doc
.EI
```

When merge-printing, **NewWord** places a data item from the data file into the variable PET (actually, the data item could just as easily come from a .AV or .SV dot command). **NewWord** then evaluates the condition. If the data item is "dog," **NewWord** moves to the .FI command; otherwise, **NewWord** skips down to the text or dot commands other than .FI between the .IF and .EI.

The other sentence in our example had to do with cats. The merge-printing "sentence" for this is exactly like the first, except for a different constant item, and a different file to insert.

```
.IF &PET& = cat
.FI cat.doc
.EI
```

Putting these two "sentences" together results in the following:

```
.IF &PET& = dog
.FI dog.doc
.EI
.IF &PET& = cat
.FI cat.doc
.EI
```

If the variable &PET& is neither "dog" nor "cat," NewWord will insert no file (in fact, **NewWord** will do nothing, since none of the conditions specified are met).

While comparing two items is the most common way of determining if a condition is met, there is one other form of the .IF command that can be useful:

```
.IF &PET&
```

Notice that there is no operator. The condition here is met if the data item in &PET& is not zero and not blank. If there is no data item, **NewWord** skips to the next .EI or .EL command.

The .EL (ELSE) Dot Command

In the previous example, we used two merge-print "sentences" to take advantage of two possible conditions, i.e. whether the variable PET is "dog" or "cat." It is possible to combine these two sentences into one. First, in English:

> If the data item in the variable PET is the same as "dog," insert the file dog.doc; or else, insert the file cat.doc.

Notice that there is a subtle shift in meaning, too. The file cat.doc will be inserted if the variable is anything other than "dog," for example, "fish" or if the data is missing.

The phrase or else should be translated into the merge-print dot command .EL. Thus, the merge-print translation of this sentence is:

```
.IF &PET& = dog
.FI dog.doc
.EL
.FI cat.doc
.EI
```

Comparing Two Data Items

In the examples we have shown you so far, the items to be compared by **NewWord** in evaluating if a condition is met have been a data item and a constant.

```
.IF &PET& = dog
```

You may also compare one data item with another, like this:

```
.IF &VAR1& < &VAR2&
```

Here, the condition is met if the data item in VAR1 comes before the data item in VAR2 according to alphabetical order.

More on Comparisons

Normally, **NewWord** ignores any spaces that come between the operator and the following item. So, the following are equivalent:

```
.IF &PET&=dog
.IF &PET& = dog
.IF   &PET&   =          dog
```

In order for two data items to be the same (=), they must have exactly the same characters. So, if you know that a data item from a data file has spaces in front of the letters, you will want to make sure that those leading spaces are included in the comparison. Therefore, you enclose the constant in quotes. Thus, the condition in the following will be met only if the data in &PET& is "dog" preceded by 8 spaces.

```
.IF &PET& = '        dog'
```

If you want to use the character ' in the constant along with leading spaces, you may enclose the entire item in double quotes ("), like this:

```
.IF &PET& = "        dog's"
```

The condition will be met if the data item is dog's preceeded by 8 spaces.

If the data item has a double quote, you may enclose the item with single quotes. However, yuu cannot have double quotes, single quotes and leading spaces in a constant all at the same time. If there are no leading spaces (and consequently no leading quote), you may use the quotation marks freely anywhere.

Alphabetical order is the governing criterion when comparing words. The word "cat" comes before the word "catalog" in

alphabetical order, so the condition is met in the following examples:

```
.IF cat < catalog
.IF catalog > cat
```

If your alphabetic comparisons involve data that have numbers in them, keep in mind that the numbers come before the letter "a." Also, case is ignored in comparisons, so that the following examples are the same as those above.

```
.IF CAT < catalog
.IF CATalog > cAT
```

Numeric Operators

The operators with the symbol £ before them are to be used with numbers, rather than words. It is possible for either item to be compared to have non-numeric characters in them, but before **NewWord** proceeds with the comparison, it removes all the non-numeric characters (except for "." and "-"). Thus, the following examples are equivalent:

```
.IF 12.45 £= 12.45
.IF $12.45 £= abc1d2.sdf45
.IF $12.45 £= 12.45
```

Notice that if you use the operator without the £ symbol in these examples, only the first would meet the condition, because numerals are considered by NewWord to be alphabetic characters that come before "a" in the alphabet. With the £ symbol, all three meet the condition since **NewWord** ignores all the non-numerals. Generally speaking, it is best to use the £-operators when comparing numbers.

What Comes Between .IF and .EI?

So far, the only thing to come after a .if command has been the come after a .if command. Instead of inserting a file from the disk, we could have typed in the actual text of the paragraph, like this:

```
.IF &PET& = dog
```

We know that your dog will love new BEEFO dog food, made from the highest quality beef and beef byproducts. As an inducement for you to come in and look over our new facility, we are offering a 25 pound bag of BEEFO for only 10 cents, this week only.

```
.EI
.IF &PET& = cat

   Your  cat will just love new FISHO cat food, made from  real
   fish  and  aromatic  herbs.  Come  in,  look  over  our  new
   facilities,  and take home a 50 pound bag of FISHO for  only
   25 cents, this week only.

.EI
```

One more example. Let's say you have two data files, one for dog owners and one for cat owners. You could design a merge-print document that allows you to use the proper data file depending on whether you choose to write to cat owners or dog owners.

```
.AV "Are you writing to cat owners or dog owners? ",PET
.IF &PET& = dog
.DF dogs.dta
.RV NAME,ADDRESS,CITY,STATE,ZIP,DOGNAME,AGE,SEX,SERIAL
.EI
.IF &PET& = cat
.DF cats.dta
.RV NAME,ADDRESS1,ADDRESS2,CATNAME,COLOR,BREED
.EI
```

Notice that both the data file and the data items to read from the files are different, depending on what &PET& is. Notice also that we have used the .av command to allow you to choose the data file at the time you merge-print the document.

Nested .IF Commands

It is possible to use the conditional merge-print commands to handle making choices among three or more alternatives by nesting the commands. Let's say you want to have a special dog paragraph in your letters to dog owners, a special cat paragraph to cat owners, and a general paragraph to all the other pet owners on your list.

First, state the task something like this:

> If the data item in the variable &PET& is the same as "dog," insert the file dog.doc; or else, if the data item is "cat," insert the file cat.doc; or else, insert the file general.doc.

At this point, we will show you a technique that will make working with nested .if commands a little easier. (Nested conditionals can become very complicated.) The technique involves writing the commands in outline form, with each .if command indented. An outline of the example would look like

this:

```
.IF &PET& = dog
.FI dog.doc
.EL
     .IF &PET& = cat
     .FI cat.doc
     .EL
     .FI general.doc
     .EI
.EI
```

WARNING: YOU CANNOT MERGE-PRINT A DOCUMENT WITH THE DOT COMMANDS INDENTED LIKE THIS! All dot commands must begin in column one, so if you make the outline on screen, you must make sure to eliminate the indentations before you try to merge-print the document.

The rule is that every .IF command must have a corresponding .EI command. The .IF command may or may not have a .EL command corresponding to it. Looking at the outline, you will notice that going down the column that each .IF command heads, there is a .EI command (there is also a .EL command). This technique will help you make sure that the conditional commands do the task you intend.

Let's say that you want to send a letter to everyone on your list who lives in St. Louis, Missouri, EXCEPT for the people who live around the University of Missouri. ZIP codes there range from 63101 to 63199, and the zip code for the University is 63121. The conditional merge-print commands would be:

```
.IF &ZIP& £<= 63199
.IF &ZIP& £>= 63101
.IF &ZIP& £<> 63121
.FI letter.doc
.EI
.EI
.EI
```

The first .IF command eliminates ZIP codes greater than 63199, and the second eliminates codes less than 63101. The third .IF eliminates the 63121 ZIP code. Only ZIP codes that meet all three conditions will generate letters. All others are ignored.

The next example is more complicated, so we will give it in outline form. REMEMBER: IF YOU TRY TO MERGE-PRINT A DOCUMENT THAT IS INDENTED LIKE THIS, IT WON'T WORK. DOT COMMANDS MUST BEGIN IN COLUMN 1.

This is what the dot commands do:

a. If the ZIP code is between 00000 and 07999, the file NEWLAND.TXT is inserted.

b. If the ZIP code is between 08000 and 09000, the file 8000.TXT is inserted.

c. If the ZIP is between 09001 and 09999, the file 9000.TXT is inserted.

d. If the ZIP is 10000 or greater, **NewWord** prints a message stating what the ZIP code is, and asks if a letter should be printed. Answering YES causes the file OUTSIDE.TXT to be inserted. Answering NO causes **NewWord** to display the message "Skipped," and no file is inserted.

e. If there is no ZIP code at all, **NewWord** displays the message "NO ZIP, NO LETTER," and goes on to the next letter.

```
.df ZIPS.DTA
.rv NAME, ADDRESS, CITY, STATE, ZIP
.if &ZIP&
   .if &ZIP& £< 10000
            .if &ZIP& £<= 09000
                     .if &ZIP& £>= 08000
                     .fi 8000.TXT
                     .el
                     .fi NEWLAND.TXT
                     .ei
            .el
            .fi 9000.TXT
            .ei
   .el
   .av "ZIP is &ZIP&.  Print? ,ANSWER
            .if &ANSWER& >= Y
            .fi OUTSIDE.TXT
            .el
            .dm SKIPPED
            .ei
   .ei
.el
.dm NO ZIP, NO LETTER
.ei
.pa
```

NEWWORD

DO IT YOURSELF

October, 1983

TABLE OF CONTENTS

1. INTRODUCTION TO NEWWORD

1.1 PREVIEW

In this chapter, we will:

DISCUSS THESE TOPICS and INTRODUCE THESE COMMANDS

HOW TO USE THIS MANUAL
JOB AIDS
DEFINITIONS OF TERMS
THE COMPUTER KEYBOARD

1.2 WORD PROCESSING

Newword is a **word processing** program. It adapts a computer to the special needs of people who type.

How does word processing differ from typing?

On the latest, self-correcting typewriters, fixing mistakes can be easy, but only if you catch them right away. Otherwise, fixing typing errors is generally a nuisance. In word processing, correcting errors is **always easy.**

Editing typewritten documents can be a real chore, involving both retyping and cutting-and-pasting. In word processing, editing is **a breeze.**

With typewriters, changing the look of the page--using different typefaces, altering line spacing and indentations, and so forth--means more retyping and cut-and-paste work. In word processing, it means **a few keystrokes.**

Form letters, envelopes and mailing labels, and

other repeated documents are **far more satisfactory** when done with a word processor.

And because Newword works on a computer, you can combine word processing with other programs--spreadsheets, data forms, gin rummy, statistical analysis--**whatever you like.** The computer can even check your spelling and syntax.

Can a typewriter match this? No!
Newword's most distinguishing feature is **what you see is what you get.** Although there are exceptions, if the text onscreen looks the way you want it to, it'll look just as good in print. And you'll always be able to check the screen to **see** how a change looks.

The variety and effectiveness of its print commands is another of Newword's distinguishing features. As you become accustomed to word processing, you'll come to appreciate, more and more, that Newword's print function **rewards creativity** in ways you never dreamed of.

There are many **automatic functions** in Newword, too. For the most part, they relieve you of many fundamental "housekeeping" chores you'd otherwise have to do. However, advanced word processing operators will appreciate that some of the automatic functions can be turned off.

Newword's many menus and help messages make it very **versatile.** At first, you'll probably use them all the time, since they explain how the program works. As you come to learn Newword, though, you'll gradually stop using them. You can do this easily, because there are several levels of **help.**

1.3 ROADMAP FOR DO-IT-YOURSELF'ERS

The **Do It Yourself** is a particular kind of training manual, called a **tutorial.** Its underlying principle is that we feel we have some responsibility for ensuring that you learn to use Newword as easily as possible.

The manual starts with several introductory sections; this is one of them. Next, you'll learn to make **working disks** and to **load Newword.** (These terms are just computer jargon--don't let them throw you.)

Then you'll plunge right into Newword, learning to type a document. After that, there are a number of sections

on correcting and changing documents. This is the real nitty-gritty of word processing.

Finally, you'll learn how to print what you've created.

Do It Yourself

To help you do this, we've laid out a very detailed set of instructions. All you have to do is follow them. Once in a while we'll explain **how** a command works, but, mostly, we'll just tell you **what to do.**

The **Do It Yourself** has 2 parts ...

The first part (chapters 1 through 8) covers the basics. But you can do most of the word processing anyone would want to with these basics.

The second part (chapters 9 through 14) covers everything else.

As you work your way through the first part, you may notice things on the screen that we don't talk about in the text. Don't be concerned. We'll get to it in part 2.

>>> Here in the left margin is a useful feature of the **Do It Yourself.** We call those 3 arrows the **input sign.**

As you work your way through the manual, you'll find that you have to read a sentence or 2, type something, read a paragraph, type something, read more, type more, and so on.

We'll display the input sign every time there's something to type or to pay special attention to.

Helpers

The **Do It Yourself** is just one of several "helpers" we've developed.

In Newword itself there are **menus** and **help messages.** You'll learn about them as you go along.

Three other helpers are **job aids.** They're in chapter 14. You may also have been provided special one-page copies that you can post on the wall or carry from work station to work station ...

Command Card

This lists all of Newword's commands in simple alphabetical order. When you're more familiar with Newword, you'll find this list to be very handy.

Flowchart

Some people find a flowchart useful. It shows the various paths you can take through Newword's menus. If you're visually oriented, you'll like this.

Function Card

All the commands are listed here, too, but they're grouped by function. For example, when you want to erase something, but can't remember the appropriate command, look for it here, under **ERASING.**

We won't mention the job aids in the **Do It Yourself** very often, but we recommend that you refer to them frequently. They're very convenient, of course, but they're also useful as reinforcements. The tutorial is so detailed that it's easy to lose sight of Newword's overall structure. The job aids will remind you of it and reinforce your learning.

The **Newword Encyclopedia** is another helper. It's a complete reference manual. After you've finished the **Do It Yourself,** when a problem comes up, or you can't remember something, or you don't know how to make Newword do something, look in the **Encyclopedia.**

And, if you're the sort of person who wants to know **why** a thing works, the **Encyclopedia** will make good reading.

1.4 WANT SOME ADVICE?

If you're familiar with hardware, software and the operating system, skip the rest of this chapter. Go on to **chapter 2.**

But if all this is new to you ...

Relax! Take your time. Work at your own best pace.

Be aware of when you're growing tired, and **stop** when you need to.

Don't be afraid of the computer. Granted, you may be confused by it, at first. But it's just a tool like any other, though a fairly complicated one. In fact, once you're an old hand, you'll realize that computers are actually pretty dumb.

Focus on learning the correct procedures, not on avoiding mistakes. You're bound to make errors; just don't be distracted by them.

Proofread. After you've typed something, read the screen carefully. Compare your entry with our directions. Be sure it's right before going on.

1.5 FIRST, GET TO KNOW YOUR HARDWARE

What's "hardware?" In computer jargon, **hardware** is machines and equipment, like your terminal and keyboard. **Software** is instructions encoded in programs like Newword. The instructions tell the hardware what to do. Manuals like the **Do It Yourself** and **Newword Encyclopedia** are called **documentation.**

You have a computer, of course. It has a **keyboard,** a video **terminal** that looks like a TV set, and probably one or more slots for disks. If you were to look in the slots you'd see the **disk drives,** which spin the disks so the computer can read and write on them.

If you don't have disk drive slots, don't worry. Newword works the same for you as for the rest of us.

Go over your hardware manuals carefully. Be especially sure to locate the **reset button,** and see if you have a **brightness knob** for the terminal screen.

If you use floppy disks, practice inserting and removing them. We can't emphasize too often that the disks are very sensitive to abuse, especially dust, grease and magnetism.

You probably have a printer, too, but you won't be using it for a while.

1.6 DO YOU KNOW?

Before we get into Newword itself, we want to go over some general terms and definitions, and familiarize you with the keyboard.

If you're at your keyboard now, we should warn you that the keys may not work in exactly the way we describe. They will when you're in Newword, though.

typing — We assume that you know how to type, but it isn't essential. In fact, some people have found that not knowing how to type is good, since they don't have to unlearn old typing habits.

characters — A character can be a letter, number, punctuation mark or symbol--anything whose image appears onscreen when you press its key. In a line of text, even a blank space is a character.

function keys — Most keys on the keyboard are character keys--press a key and it types a character. There are a few function keys, as well--the SHIFT key, for example, or the TAB key.

Your keyboard may have additional function keys, as well. Usually they're labeled **F1**, **F2**, **F3** and so forth. You may also have keys with arrows pointing up and down, and left and right.

These added keys supplement some of Newword's commands. They're explained in your hardware manuals. Throughout the **Do It Yourself**, though, we'll only show you Newword's commands.

cursor — The cursor is displayed onscreen. It points to where the next character will be entered. Depending on the terminal, it may be a brightly lighted rectangle about the size of a capital "O," or an underline. Throughout this tutorial it will be shown as **[]** (usually with a character inside).

RESET button — Somewhere on the computer or keyboard, there's a reset button. Pushing it causes most computers to come to a complete halt, allowing you to start over from

scratch. You'll have to load Newword again, though, because resetting empties the computer's memory.

Your reset button may be called a boot key, because resetting is often called "booting" or "rebooting." Or it may have another name. Refer to your hardware manuals to find it.

DEL — DEL's a function key; it means "delete," or erase. On some keyboards it's labeled "RUB OUT," on others it has a special symbol. You'll practice using it later on.

ESC — There's probably an ESCAPE key on your keyboard. It's used once in a while in Newword. We'll point it out as we go along.

BACK — You may have a BACKSPACE key that moves the cursor to the left. It may have a special symbol.

LINE FEED — The LINE FEED key is on most keyboards--it advances the paper roller on the printer, one line space.

In Newword, however, it has a special function: **help**! You'll learn how to use it soon.

REPEAT — In one way or another, nearly every keyboard has a **repeat** function that lets you type a character over and over again.

On some keyboards, all you have to do is hold down the character key. Holding down the D key, for example, will produce this on the screen: **ddddddddd**. This is called "auto repeat." On other keyboards, you have to type **d** once and then hold down a special key, labeled **REPEAT**, or something similar.

CTRL — Most computer keyboards have a CONTROL key labeled "CTRL." It works like a shift key, but instead of changing letters to upper case, it enables you to give commands to Newword.

^ — This little symbol is a **caret**. It stands for the CONTROL key. It will be shown with another character or 2, as in "^d"

or "^**pb.**" Whenever you see this combination, hold down the **CONTROL** key while you type the letter(s) shown after the caret.

Note that most keyboards can type a caret. Usually, one of the number keys has a "^," but, in the **Do It Yourself,** don't type a caret unless you're specifically told to.

CAPS LOCK **CAPS LOCK** is a function key not found on typewriters. It's similar to a shift lock, but it works mainly for letters, not for symbols or punctuation marks. For instance, even with **CAPS LOCK** on, to enter a dollar sign you have to hold down the shift key while pressing "4."

Sometimes the **CAPS LOCK** key is labeled "**ALL CAPS.**" Yours may have another name.

However, your computer may not have a **CAPS LOCK** key at all. If that's your situation, here's a Newword command that does the same thing ...

CONTROL
^

Hold down the **CONTROL** key and **type the caret.** (Yes, here's an instance when you should **type** the caret.)

On some keyboards, you have to hold down both the **CONTROL** key **and** the **SHIFT** key, before typing "^."

On a few keyboards, there is no "^" character at all; ask your local computer whiz or dealer about generating the character with your keyboard.

This procedure is clumsy, so don't use it unless you have to. It turns the **CAPS LOCK** function on **and** off, by the way. If the function's off, this procedure turns it on; if the function's on, the procedure turns it off.

<cr> Remember this sign; it means "carriage return." Whenever you see it, press the **RETURN** key. The key has several pur-

poses, which you'll learn as we go along.

enter This is a term often used instead of, "Press these keys." But it means a little bit more; it means, "Press these keys **and** press **RETURN**."

Sometimes Newword will ask you for information or instructions. After you type your response, you'll have to press **RETURN** to get Newword to act on it.

files Any information stored on the computer's disks is a file. Files can be just a word or 2, or thousands of words.

Most people have trouble, at first, using "file" this way, but you'll get the hang of it after a while.

In word processing, most files are documents. There are other kinds of files, too, called "**non**-documents."

Not all computers use the keys and terms we've just reviewed. Check your hardware manuals and keyboard against our list. Remember the differences as you go through the **Do It Yourself.**

1.7 SIMPLE?

You may not think so. But please remember--starting is really the hardest part. In a short while, you'll be doing fine.

So take a deep breath, relax, and go on.

1.8 REVIEW

Here's what we've covered so far. If any of it is particularly unfamiliar, go back and review it. But you don't need total recall. If you feel generally confident that you're doing okay, you **are** doing okay.

At each of these reviews, it's a good idea to take a break. Stretch, go for a short walk, have a snack--do whatever helps clear your mind, so you can go on to the next sections of the **Do It Yourself** feeling refreshed.

Introduction to Newword	* Typing and word processing * Newword's most important features
Roadmap	* The **Do It Yourself** * Job aids and **Encyclopedia**
Advice	* Relax! * Definitions * The computer keyboard

2. LOADING NEWWORD

2.1 PREVIEW

In this chapter, we will:

DISCUSS THESE CONCEPTS and INTRODUCE THESE COMMANDS

NEWWORD & CP/M
FOUR KINDS OF DISKS
  System Disks
  Distribution Disks
  Boot Disks
  Working Disks
INSTALLING NEWWORD
LOADING NEWWORD
BACKUP FILES

2.2 GETTING GOING

This chapter explains how to get going with Newword. Mostly, this means preparing some disks.

We'll give you detailed instructions that are generally applicable to desk-top microcomputers with 2 disk drives, for 5 1/4-inch **floppy disks.** We'll assume, too, that your computer uses the **CP/M** operating system.

However, before you can follow our instructions, you **must** read your computer manual **and** your CP/M manual. We won't show you how to operate your computer. Nor will we explain very much about CP/M. It's up to you to learn these yourself.

If you've already studied the manuals, and if the 6 topics listed below sound familiar (You don't have to be an expert.), keep going in this chapter ...

Choosing blank disks that are appropriate for your computer, and inserting disks in the disk drives.

Resetting the computer.

Typing at the keyboard, reading the screen and pressing the **BACKSPACE** and **RETURN** keys.

Formatting disks, with the **format** program.

Putting CP/M on disks, with the **sysgen** program.

Copying files from one disk to another, with the **pip** program.

If you haven't studied, or if the topics sound new and "foreign," stop right here. Close this manual and read your computer and CP/M manuals, instead. Remember to pay special attention to the 6 topics. Come back to the **Do It Yourself** when you're done.

The programs named in the above list are standard ones supplied with many computers that use CP/M. If you have programs with these same names, the instructions we give you will probably be all you'll need. Just follow our directions from one step to the next.

However, the names of your programs may be different from ours. If they are, you'll have to look through your programs to find ones that do the same things ours do. Find the programs that format the disk, put CP/M on it, and copy files from one disk to another. Then, as you work through this chapter, you'll have to compare our instructions with yours. When the instructions are different, follow yours, **not** ours.

2.3 NEWWORD, CP/M & YOUR COMPUTER

In the previous chapter, in section 1.4, we mentioned some of the basic parts of a computer--keyboard, terminal, disk drives. Inside the computer, there are other parts, especially the central processing unit, which does most of the computer's "thinking." The operating system keeps all these parts working together--it runs the whole computer. It also coordinates all the equip-

ment you hook up to the computer, like printers and modems.

There are several operating systems on the market. CP/M is one of the most popular. It's published by a company called Digital Research, which sells it to individual computer manufacturers, the same way other programs are sold.

What does the operating system have to do with Newword?

Well, your computer can't do anything at all, if CP/M's not in there, coordinating all the parts and equipment. So, in order to use Newword, you first have to put in CP/M.

"Loading" is the computer term for putting a program in a computer; actually, it goes into the computer's **memory.** Check your CP/M and computer manuals for instructions on loading CP/M. Usually, you do it in just 2 steps ...

Put a disk in drive A that has CP/M recorded on it.

Reset the computer, by pushing the **RESET** button.

The computer will read the CP/M program that's on the disk, and then write the program in its memory. This whole procedure is called "booting the system."

On some computers there's a 3rd step, pressing the **RETURN** key after the **RESET** button.

2.4 DISKS, DISKS, DISKS

In this chapter, you'll learn about these 4 kinds of disks ...

Boot disks	Disks that have the CP/M program recorded on them.
Distribution disks	The disks on which programs are supplied by the computer manufacturer or program publisher.
Working disks	The ones you use for day-to-day work.

Backup disks	Reserve copies of other disks, used only when the originals are damaged or erased.

First, some reminders about disks. **Be sure you handle floppy disks carefully.** Take them out of their loose sleeves when you put them in the computer, but **don't ever** remove the tightly glued-together wrapper that completely covers the disk itself. It's important, too, to protect disks from magnets, from dust, dirt and smoke, and from extreme heat and cold. And ...

DON'T TOUCH THE DISK'S PLASTIC SURFACE!

Disks are often called **diskettes.** They're called **floppies,** too, because they're so flexible, and the disk drives are sometimes called **floppy drives.**

In the following sections, you'll format a disk. Then you'll put CP/M on it. And you'll put the file, PIP.COM, on it, too.

Then you'll use the disk to make a backup copy of the Newword distribution disk.

Then, you'll make a **Newword** working disk. This will involve formatting another disk, copying files to it, and a procedure called "installing." Then you'll make a backup copy of the disk.

Finally, you'll load **Newword** in the computer.

We'll use standard, CP/M-based programs and procedures for these tasks, but, remember, if your computer came with other programs, you should use them, instead.

There are lots of small steps in the procedures we'll show you, so take your time and follow along carefully. Whenever it's your turn to do something specific, like type or proofread, we'll put 3 arrows (>>>) in the space to the left of the text. The arrows are the **input sign.** They're a signal that it's time for you to stop reading and do something.

Along with lots of steps, there are many new concepts and terms in this chapter, and throughout the manual. When you feel confused by all the new information, **stop.** Take a short break, to clear your mind. Read the confusing sections again. Look for some detail that you missed the first time--missing some small thing is often the cause for not understanding.

LOADING NEWWORD

2.5 FORMATTING A DISK

Formatting is the way the computer prepares a disk for use. Disks that are formatted for one kind of computer usually won't work on other kinds. It would be more convenient if they did, but they just don't. So you have to format your own disks yourself, with a program that's usually called **format.**

Here goes.

>>> Start by pressing the computer's RESET button.

When you do, a message like this will probably appear on the screen, but the message isn't necessary ...

Load a boot disk in drive A

>>> Next, in drive **A,** put the boot disk provided by your computer manufacturer. It may be called a "system disk," instead. It will have CP/M recorded on it.

It should have these 3 programs on it, too ...

FORMAT.COM
SYSGEN.COM
PIP.COM

(Remember, use your own programs, instead, if you have any.)

How do you know if CP/M's on the disk? How do you know if the 3 programs are on it?

Well, your computer and CP/M manuals should tell you how. But here's a way ...

CP/M on disk?

>>> In drive **A,** insert the boot, or system, disk.

If CP/M is on the disk, the computer will put the program in its memory. Then it will write these 2 characters below the other lines on the screen ...

A>

... and the cursor (■) will be right after the arrow, like this ...

A>■

This very short line is called the "operating system prompt." You don't have to type it--the computer does.

The prompt is a cue, a signal to you that CP/M, i.e., the operating system, is in the computer's memory. It's displayed whenever a program ends, to show that CP/M is ready for you to start the next one.

Also, if the prompt is "A>," it means you're currently working on drive A. If the prompt is "B>," you're working on drive B. And so on. Your CP/M manual explains this in detail.

If CP/M's **not** on the disk, the operating system prompt will **not** appear. Instead, the computer will display an error message, like this ...

Disk error: boot disk not in drive A.

The error message may have instructions about what to do next, too.

>>> If you get an error message, find a real boot disk, i.e., a disk that really does have CP/M on it, reset the computer (by pressing the **RESET** button), and try again.

Programs on disk?

To see what programs are on a disk, use a **directory** program. CP/M has one built-in, so, once you have a boot disk in drive A, all you have to do is type "dir" and then press **RETURN**. The screen will display a list of all the files on the disk.

Check the disk in drive A now, to be sure the 3 programs you need are on the disk. Just type ...

>>> **dir**

The letters will appear to the right of the operating system prompt, like this ...

LOADING NEWWORD

A>dir

... and the cursor (■) will be after "dir" ...

A>dir■

>>> Proofread what you typed. If it's correct, press **RETURN.** If it's not, use the **BACKSPACE** key to erase the wrong letters, then type the right ones and press **RETURN.**

All right. Those were the preliminaries. Let's format a disk now.

>>> Leave the disk in drive A alone, but put a new, blank disk in drive **B.** Make sure it's a disk that's appropriate for your computer.

Next, type ...

>>> format

On the screen, it will appear like this: **A>format■.** (Remember, "■" stands for the cursor.)

>>> Proofread what you typed. If it's okay, press **RETURN.** If it's not, use the **BACKSPACE** key to erase the wrong letters, then type the right ones and press **RETURN.**

The screen is displaying a new message now, something like this one ...

Disk drive? (A, B, ...) ■

The program is asking what drive the blank disk is on. You put the disk in drive B, so type ...

>>> B

(Some programs may let you type "b" instead.)

>>> Proofread what you typed. If it's okay, your screen will have another message on it, below the first. **If it's not okay,** reset the computer and type "format" again.

Depending on your computer, the next message may, or may not, be like this ...

Density? (S = single, D = double) ■

"Density" refers to how compactly data is recorded on the disks. Check your computer manual, then type the letter that's appropriate for your computer.

However, if the message doesn't appear, it's because you don't need it. So don't worry.

Next, the screen will display something like this ...

Insert disk to be formatted in drive B, then press RETURN. ■

This is a reminder, to be sure you actually have a disk in the drive.

>>> To be on the safe side, check to be sure there's a blank disk in drive B. When you're sure there is, press the **RETURN** key. Go ahead; press it.

Now the computer is formatting the blank disk. It will take a minute or 2, and the program may be displaying a message that says the formatting is underway. Check the screen, to see if it is.

When the formatting's done, a message like this will appear ...

Disk is formatted.

Press RETURN to end program, or type F to begin again. ■

>>> You're finished with the **format** program for now, so press **RETURN**, to end it.

2.6 PUTTING CP/M ON A DISK

The next job is putting CP/M on the blank disk, so type ...

>>> **sysgen**

On the screen, it will appear like this: A>sysgen■.

LOADING NEWWORD

>>> Proofread what you typed. If it's correct, press **RETURN.** **If it's not,** use the **BACKSPACE** key to erase your errors, type the correct letters, and press **RETURN.**

The screen will display a message like this ...

Source drive? (A, B, ...) ■

The computer's asking where it can find CP/M. It's on the disk in drive A, so drive A is the source. Therefore, type ...

>>> **A**

(Some programs may let you type a small "a.")

>>> Proofread what you typed. If it's correct, fine. **If it's not right,** reset the computer and type "sysgen" again.

The computer may, or may not, display a reminder message now, like this ...

Put source disk in drive A, press RETURN. ■

If there's a reminder message like this on the screen, press the **RETURN** key. But if there's no message, don't press it.

Next, a message like this will come onscreen ...

Destination drive? (A, B, ...) ■

Now the computer's asking where it should put CP/M. You want it on the disk in drive B, so drive B is the destination. Therefore, type ...

>>> **B**

(Some programs may let you type "b.")

>>> Proofread what you typed. If it's correct, fine. **If it's not,** reset and start again.

The computer may display another reminder message now, like this ...

Put destination disk in drive B, press RETURN. ■

... or it may display a simpler message, like this ...

Press RETURN to begin. ■

>>> In either case, go ahead--press **RETURN.**

The program is very quick, so it's probably finished already. The screen is displaying a message like this ...

Press RETURN to end the program, or type S to begin again. ■

>>> You're finished with this program, so press **RETURN.**

2.7 COPYING FILES WITH PIP

Now it's time to put PIP.COM on the blank disk. As your CP/M manual explained, PIP.COM enables you to copy files, or to send them from one disk to another, or from one piece of equipment to another.

Leave the 2 disks in their drives, and type this ...

>>> **pip b:=pip.com**

Onscreen, it will look like this: **A>pip b:=pip.com■.** It means you're going to "pip," or copy, the file PIP.COM to the disk in drive B.

>>> Proofread what you typed. If it's exactly like our example, press the **RETURN** key. **If it's not,** use the **BACKSPACE** key to erase your mistakes, type the line correctly, and press **RETURN.**

When the copying is done, the computer will let you know by displaying the operating system prompt below the other lines (A>■).

2.8 TIME OUT

How are you doing? You've accomplished quite a bit. The disk in drive B is no longer blank. It's been formatted, CP/M has been recorded on it, and there's one file on it, PIP.COM.

If you want to, you can see the contents of your disk, by typing "**dir b:**," and pressing **RETURN**. CP/M's built-in directory program will display the contents of the disk in drive B. Only PIP.COM will be listed; CP/M itself never shows up in the directory.

In a minute, you can take a break. But, first, we want to explain a couple of things.

First, about typing upper and lower case letters.

Upper case letters are capital letters ...

A B C D E

Lower case letters are small letters ...

a b c d e

Sometimes it doesn't matter to the computer whether you type upper or lower case letters. The computer will understand what you mean, no matter which you use. Sometimes, however, it does matter. It depends on the program.

Typing upper case letters involves an extra keystroke--holding down the **SHIFT** key. Therefore, here in the **Do It Yourself,** we'll have you type lower case letters whenever possible, simply because they're easier.

Second, about the way we give instructions. We're not going to remind you to proofread anymore. You'll have to remember to do it yourself. And we're going to start using a new symbol ...

<cr>

Pronounce it as "carriage return." If you're an experienced typist, you know where the term comes from. If you're not, don't worry about it. Just remember that it means, "Press the **RETURN** key."

We'll start using a new computer term in place of "type," too. We'll use "enter." It's more accurate.

Okay. That's all for now. Take a break. Get up and stretch. Don't think about computers for a while.

Come back when you feel refreshed.

2.9 DISTRIBUTION & BACKUP DISKS

Newword is supplied on a floppy disk called a "distribution disk." It's the original disk on which Newword is distributed to computer users like you. Most distribution disks are write-protected, meaning you can't write, i.e., record, anything on them. You can only read their files, and copy the files **from** the distribution disk **to** other disks.

Do not use the Newword distribution disk for routine work. What would happen if the disk were damaged? You'd lose Newword.

In fact, the first thing to do with the distribution disk is to make a copy of it. You'll make a backup copy, also called a "backup disk."

Backup disks are a safety measure. They're not used in day-to-day work. Instead, they're put away for safekeeping, preferably **not** in the same room as the computer. Backup disks are used only when the original disks are damaged or inadvertently erased.

To make the backup disk, follow these steps ...

>>> Take the disk out of drive **A**, and put it aside.

>>> Move the disk you just formatted from drive **B** to drive **A**.

>>> Put the Newword distribution disk in drive **B**.

>>> Reset the computer.

Enter ...

>>> **pip a:=b:*.***

(Remember to proofread and correct mistakes.)

LOADING NEWWORD

<cr>

(This means, "Press the RETURN key.")

Onscreen, the line will look like this: "A>pip a:=b:*.*■." It means every file on the disk in drive B, i.e., on the distribution disk, will be copied to the backup disk in drive A.

As each file is being copied, its name will be displayed onscreen. When the copying's over, the last line on the screen will be the operating system prompt ...

A>■

>>> Make a label for the backup disk. It should say something like, "Backup--Newword distribution disk." Disk manufacturers recommend that you write on the label **before** affixing it to the disk. Most of them also recommend that you **not** use a ball-point pen--the ink tends to bleed through the label and disk wrapper, onto the disk itself.

>>> Remove the disk from drive **A**, put the label on it, and put it away, preferably in another room. (Storing backup disks in another room is extra protection against loss in case of fire, theft, etc.)

2.10 BOOT DISKS

As we explained in sections 2.3 and 2.4, boot disks are simply ones on which the CP/M program has been recorded. Here's why they're important.

Every time you press the RESET button, CP/M is erased from the computer's memory. However, since CP/M has to be in the memory for the computer to work, you have to put it back in, right away. The way you do it is to put a boot disk in drive A. This is called "booting the system."

Once you've reset a computer, you have to boot the system before you can use the computer again. If there's a boot disk in drive A when you reset, the computer can boot right away, because it will be able to put CP/M in its memory immediately. If there's no disk and,

instead, you have to insert one, the computer will wait until you do.

Some computers also require you to press the **RETURN** key after pressing **RESET**. They won't look for CP/M on the disk until you do.

When you're more experienced, you may want to keep just one boot disk, to use only if you've had to reset the computer. However, while you're learning Newword, you'll probably be resetting often, so we'll make sure you always have a boot disk in drive A.

2.11 WORKING DISKS

A working disk is one that you use for day-to-day work. Period.

It's not a distribution disk, because files are put on it by you, not by the computer manufacturer or program publisher. It's the opposite of a backup disk, because it's used for routine work. And it can do double-duty as a boot disk, if you put CP/M on it.

Let's make a **Newword** working disk. To do that, you'll have to prepare another blank disk.

Here are the instructions for preparing the disk ...

>>> In drive **A**, put your boot, or system, disk. It's the one that has CP/M, FORMAT.COM, SYSGEN.COM and PIP.COM on it. You used it to format the first blank disk.

>>> In drive **B**, put a blank disk.

>>> Reset the computer. This will boot the operating system.

>>> Format the blank disk, by typing "format" and pressing **RETURN**. Follow the directions in the onscreen messages. Refer to section 2.4, if you want help.

>>> Put CP/M on the disk, by typing "sysgen" and pressing **RETURN**. Follow the directions in the onscreen messages, and refer to section 2.5 for help.

Copy PIP.COM to the disk in drive B, by entering ...

LOADING NEWWORD

>>> pip b:=pip.com

(Proofread and correct mistakes.)

<cr>

(<cr> means, "Press **RETURN**.")

Okay. The disk has been formatted, and CP/M and the file PIP.COM are on it.

Here's the next set of instructions ...

>>> Remove the boot disk from drive **A**, and put it away.

>>> Move the disk in drive **B** to drive **A**.

>>> Put the Newword distribution disk in drive **B**.

>>> Reset the computer by pressing **RESET**.

That part was easy.

You're ready now to copy files from the distribution disk to your new working disk. You'll use PIP.COM for this.

Here are the instructions ...

Enter this ...

>>> pip<cr>

On the screen, the last line will look like this now ...

*■

So enter ...

>>> a:=b:*.ovr<cr>

The line will look like this: *a:=b:*.ovr■; it will copy every file ending with .OVR from the

distribution disk in Drive B to the working disk in Drive A. There are 3 files: NWPRINT.OVR, NW.OVR and NWMSGS.OVR. They'll be listed onscreen as they're copied.

When the copying's finished, this will be the last line on the screen ...

```
*■
```

Now enter ...

>>>

```
a:=b:*.doc
<cr>
```

It will look like this: ***a:=b:*.doc■**, and it will copy all the files that end with .DOC. There are 3; they'll be listed as they're being copied.

>>> When the 3 .DOC files have been copied, you can get out of the **pip** program. To do this, simply press **RETURN.**

The computer will write the operating system prompt on the screen ...

```
A>■
```

Let's check the working disk with CP/M's **directory** program, so you can see all the files you copied to it.

Enter this ...

>>>

```
dir
<cr>
```

The computer will display the files that are on the disk, like this. On your screen, though, the files may be in a different order ...

```
NW       OVR : NWMSGS   OVR : NWPRINT  OVR : PIP      COM
PRACTICE DOC : SAMPLE1  DOC : SAMPLE2  DOC
```

If you're feeling a little overwhelmed, take a break. Come back when you're ready.

LOADING NEWWORD

2.12 INSTALLING NEWWORD ON YOUR WORKING DISK

The next job is called "installing." It means adapting a program to the individual characteristics of your hardware--terminal, computer, printer, etc. Once you've installed a program on a particular set of hardware, you don't have to do it again.

At this point, we're concerned only about your terminal. The rest of your hardware will be taken care of automatically.

The following is based on **Nuts & Bolts,** the user's guide that accompanies Newword's installation program. Refer to that manual for more help.

Okay?

First, enter this ...

>>> **b:instalnw**
<cr>

(Proofread, correct errors, then press RETURN.)

This means, "Start the install program, which is on the disk in drive B." That's the distribution disk.

On the screen, you'll first see a copyright notice and information about the version of install that you have. It's not important just now. After a few seconds, the following will appear ...

Name of file to install?

Enter ...

>>> **b:nwu** (Yes, it ends with a "u.")
<cr>

NWU is the uninstalled version of **NEWWORD.** It's on the distribution disk, too.

This message will appear now on the screen ...

File to hold installed Newword?

Enter ...

>>> a:nw (This time, no "u.")
<cr>

You just told install to put the installed version of **Newword** on your working disk, which is in drive A, and to name is **nw.**
The screen will now display this ...

MAIN MENU

Please select one of the following to install.

A Terminals	**E Document related items**
B Printers	**F File related items**
C Computer related items	**G Fresh user area**
D Newword related items	**H Special patches**
J Help with this menu	**X End of installation**

What is your choice?

(A menu is a list of options from which you can choose. Install has many menus, but you'll be using only 2, this time.)

Type ...

>>> a (Yes, there's no need to press **RETURN.**)

Now install will display this message ...

The terminal that Newword is now installed for is the TeleVideo 925/950.

This will be followed by ...

Do you want to change this? (Y/N)

If your terminal is not a TeleVideo 925 or 950, follow these steps. (If your terminal is a TeleVideo 925 or 950, read through the steps, but don't do anything. We'll let you know when you need to go to work again.) ...

Type ...

LOADING NEWWORD

>>> y (This means you want to make a change.)

Install will display a list of terminals, followed by ...

What is your choice?

>>> Look for your terminal on the list.

If your terminal is not on this list, you'll have to do some research before going any further ...

"Emulation" is a technical term for imitation. Most computer terminals, especially the newer, free-standing models, can imitate a few other terminals. They do this to make up for there being few industry-wide standards for terminals.

>>> Find the manual, or section of a manual, that explains your terminal's operations and capabilities. Look up the section on emulation; it may be listed in the table of contents or the index. If you can't find "emulation," try looking for "terminal parameters," "DIP switches," "other terminals," "installing," etc.

>>> Read the section on emulation. See if your terminal can be made to imitate one of those on the above list. If it can, follow the manual's directions for setting up the emulation. Then come back to the **Do It Yourself.** When it's time to install the terminal, install the one you're emulating.

(An alternative would be to ask your dealer, or someone else who might know, about emulation.)

>>> If your terminal, or the one you're emulating, is on the list, type its letter.

In response, install will display this ...

The terminal that Newword is now installed for is the [one selected].

Do you want to change this? (Y/N)

Check to be sure the terminal really is the one you chose. If it is **not,** type **y,** and choose again. If it is, type ...

>>> n

The Main Menu will return to the screen.

>>> You're ready to end the install program, so type x.

When install is ending, it displays ...

The changes you requested have been made.

Newword is now installed for the following:

Adds 25 [or whatever terminal you selected]
Draft Printer

Do you want to change this? (Y/N)

Check once again, just to be on the safe side. If the terminal's correct, type n. If it's not, type y and start again.

When the installing is done, the computer will write the operating system prompt, A>, at the bottom of the screen.

If your terminal is a TeleVideo 925 or 950, or if it emulates a TeleVideo 925 or 950, follow these steps ...

Type ...

>>> n (This means you don't want to change the current installation.)

The Main Menu will be displayed.

Type ...

>>> x

Install will display ...

LOADING NEWWORD

The changes you requested have been made.

Newword is now installed for the following:

TeleVideo 925/950
Draft Printer

Do you want to change this? (Y/N)

Check once again, just to be on the safe side. If the terminal's correct, type n. **If it's not,** type **y** and start again.

When the installing is done, the computer will write the operating system prompt, A>, at the bottom of the screen.

>>> Take the distribution disk out of drive **B.** Put it away. You won't have to use it for a while, as long as you don't change your terminal or other hardware.

>>> Leave the working disk in drive **A.**

2.13 LOADING NW, AT LAST

It's taken a long time to get here, to when you can finally load our word processing program in your computer and see what it's like. After all the work you've done to get to this point, loading **nw** may be an anticlimax, because it's so easy. But we hope not.

Here's what you have to do ...

Onscreen, the last line is the operating system prompt: A>■. So enter ...

>>> nw<cr>

(Yes, you have to press **RETURN,** again.)

Onscreen, it looks like this: A>nw■. It will load **nw** in the computer's memory.

What happened after you pressed the **RETURN** key? Well, if **nw** is properly installed, first, the computer read

the NW.COM file that's on the working disk. (The install program put it there.)Then it followed the file's instructions. For the most part, they had to do with putting parts of the **nw** program into the computer's memory.

While this was going on, Newword displayed the copyright notice and the install specifications ...

```
Newword 1.00N    Serial No. 000000
Copyright (c) 1983 Newstar Software Incorporated
All rights reserved

     [Terminal]
     [Printer]
```

The copyright and install message stayed on the screen for several seconds. Then, when all the necessary parts of the program were in the computer's memory, Newword displayed the Opening Menu ...

O P E N I N G M E N U

D get a document to change, or create a new document	**L change logged disk drive**
N create or change non-document	**E rename a document**
P print a document	**O copy a document**
M merge-print a document	**Y delete a document**
C protect a document	**F directory off**
X all done with Newword (exit)	**R run a program**
	J help

Once this is displayed, the loading is complete, and Newword is ready to go to work for you.

However, if something went wrong, the onscreen display could have been garbled in any number of ways. We can't tell you exactly what went wrong, but we can suggest a few things ...

If you're emulating another terminal, be sure you've adjusted your own terminal correctly. Review the emulation section of your terminal manual, carefully. Call your dealer if you have any problems.

Repeat the installation procedure. If you make mistakes while typing "b:ns" or "b:nwu" (with a "u"), the computer will let you know very soon. But it won't let you know if you make mistakes while typing "a:nw." So proofread that line extra carefully.

LOADING NEWWORD

When you select a terminal at the Main Menu, be sure it's the right one. If you're emulating another terminal, select the one you're emulating, **not** your own terminal.

After you make your selection, what does say about the current terminal? Make sure it names the right one. If you're emulating another terminal, it should name the one you're emulating--it should **not** name your own terminal.

If you're still having trouble, turn to Appendix B of the **Newword Encyclopedia** and go through the custom terminal installation procedure outlined there.

Keep installing and loading **nw** until you have it right.

Having learned how to start Newword, how do you stop it? Well, the easiest way, right now, is just to reset the computer.

>>> Go ahead; press RESET.

>>> It's wise to make a label for the disk right away. It should say, "**nw** working disk," or something to that effect. **Don't** use a ball-point pen.

2.14 MORE DISKS

In the next chapter, you'll begin using Newword. Before ending chapter 2, though, we want to show you a few more things.

First, now that you have a **nw** working disk, you should make a backup copy of it right away. Here's how ...

>>> Remove the working disk from the computer, and put the label you just made on it. Set the disk aside.

>>> Reset the computer.

>>> Get a blank disk, and format it, the way we showed you in section 2.4. You'll need to put the boot, or system, disk that came with your computer in drive **A**, and the blank disk in drive **B**.

>>> Put CP/M on the blank disk, the way you did in section 2.5.

>>> Remove the boot disk from drive **A**. Put the **Newword** working disk in its place.

>>> Reset the computer.

Enter this ...

>>>
```
pip b:=*.*
<cr>
```

Onscreen, it will look like this: A>pip b:=*.*■. It will copy all the files on the working disk to the backup disk in drive B.

>>> Make a label for the backup disk ("Backup-**nw** working disk"), remove the disk from the computer, put the label on it, and store it with the other backup disk you made.

There's another way to make a backup disk. But you need a special program, **backup**. Sometimes it's called **fastcopy** or **diskcopy**, instead. It copies, just like **pip**, but it **only** copies the whole disk, whereas **pip** can copy individual files, as well as the whole disk.

Here's a quick summary of how to use **backup** ...

First, put the **backup** program on your **nw** working disk. Use PIP.COM to copy the program to the working disk from the boot, or system, disk that came with your computer.

Next, use that same boot disk to format a blank disk.

Next, put the blank disk in drive **B** and the working disk in drive **A**.

Reset the computer.

Enter ...

backup

(Use your program's name, if it's different.)

<cr>

Additional instructions may be displayed onscreen after you press RETURN. If they are, follow them carefully.

LOADING NEWWORD

Keep in mind this further difference between the 2 programs. Even though **pip** can copy the whole disk, it cannot copy the CP/M program on a boot disk. Only **backup** can do that.

Also, if you haven't already done it, you should take time now to make a backup copy of the boot, or system, disk that came with your computer. Format a blank disk and use one of the programs we just showed you, i.e., either **pip** or **backup.**

And, remember to **update your backup disks** every time you change the original working disks.

Okay. That's the first thing--making more backup disks. Here's the next thing.

Later in the **Do It Yourself,** we'll tell you to put a new, formatted disk in drive B. You don't have to prepare the disk right this minute, but you can, if you want the practice. All you really need to do is format it, but you can make it a boot disk, as well, if you want to. Putting PIP.COM on it usually proves to be an advantage, too.

Finally, throughout this chapter, we've shown you how lines look on the screen. Well, we're not going to do that anymore, except when we feel it's really necessary. For example, we won't show you, A>, the operating system prompt, or the cursor, ■. At least, not very often.

2.15 .BAK FILES

A word to the wise. You'll be making your share of mistakes, while you're learning Newword. Don't let them bother you. Everyone goofs when they're getting used to a new program. In fact, even experienced word processing operators make mistakes. They're inevitable.

Unfortunately, your computer will make mistakes, too. When they happen, they'll probably erase the document you're typing. There's not much you can do about them. Just reset the computer, and start again.

But here's where backup disks come in handy. You'll have a backup copy of every working disk. When the computer goofs and swallows a document, you can get yesterday's version of the document from the backup disk. Use the **pip** program to copy it from the backup disk to the working disk.

What's more, if you're working with Newword when the computer goofs, here's something else you can do. (This explanation may be hard to understand, now, but remember that it's here when you need it.)

Newword automatically makes its own backup copies of all your documents, right on the working disks. You'll see the copies listed in the directory as you work through this manual. They're called ".BAK files," because their names always have the same ending: .BAK. For example, Newword's .BAK file of a document called "THIS.DOC" would be named "THIS.BAK."

.BAK files are usually more current than the copies on backup disks. So, when the computer swallows a document that you've made with Newword, remember that you can replace it with its .BAK file.

However, as a precaution, Newword won't let you use a .BAK file **until you change its name.** All you have to change is the ending. Do this with CP/M's built-in **rename** command. For example ...

> Suppose you're working on a document called "TODAY." It's .BAK file would be named "TODAY.-BAK." Suppose, too, that your computer swallows TODAY.
>
> You'd have TODAY.BAK to replace it, but before you could work on TODAY.BAK, you'd have to rename it.
>
> To do that, reset the computer. Then, at the operating system prompt, enter this ...
>
> ```
> ren today=today.bak
> <cr>
> ```

(Check your CP/M manual for directions about keeping track of what drive you're working on, and about naming drives when you use the built-in commands.)

You can rename a .BAK file to anything you like--just don't use the .BAK ending. And, note that, even if your document has its own ending, such as "TODAY.MON," Newword's .BAK file will still be called "TODAY.BAK."

2.16 THAT'S ALL

Take a break. You really deserve it!

2.17 REVIEW

* Studying your computer and CP/M manuals.
* Newword and CP/M.
* 4 kinds of disks.
* Formatting a disk, with **format.**
* Putting CP/M on a disk, with **sysgen.**
* Copying files, with **pip.**
* Making a backup copy of the Newword distribution disk.
* Boot disks.
* Working disks.
* Making a **Newword** working disk.
* Installing **Newword,** with **instalnw.**
* Loading **Newword.**
* .BAK files.

LOADING NEWWORD

3. OPENING

3.1 PREVIEW

In this chapter, we will:

DISCUSS THESE TOPICS and INTRODUCE THESE COMMANDS

TYPING A DOCUMENT
OPENING A DOCUMENT . . . . . D
CLOSING A DOCUMENT . . . . . ^KD
OPENING MENU HELP. . . . . J
EXITING NEWWORD. X

3.2 OPENING MENU

>>> In section 2.12, you learned how to load **Newword**, and how to leave it. Review that section, then put your Newword working disk in drive **A** and load the program.

Once **Newword** is loaded, on the screen you're confronted by the Opening Menu. It looks like this ...

OPENING MENU

D get a document to change, or create a new document	L change logged drive
	E rename a document
N create or change non-document	O copy a document
P print a document	Y delete a document
M merge print a document	F turn directory off
C protect a document	J help
X all done with Newword (exit)	R run a program

DIRECTORY Drive A
FIRST.DOC PRACTICE.DOC SAMPLE1.DOC SAMPLE2.DOC

Just like menus in a restaurant, computer **menus** list options from which you can choose. The options are

functions, like create, print and copy. To select an option, you simply press a key or 2. For example, to select the function, "print a document," all you have to do is type **p.**

Once you've made a selection, Newword may ask you for further information--a name, maybe, or more detailed instructions. The messages asking for the information are called "**prompts.** " In this example, after you typed **p**, Newword would ask you to name the document that you wanted to print.

Here in the Opening Menu, the options have to do with basic file maintenance, and the prompts usually ask for the name of the file.

You should respond to prompts by typing the necessary information, just as if you were typing the text of a document. Then you must usually press the **RETURN** key. For the example, you'd type the name of a document, then press **RETURN.**

Below the menu is a **directory** of the files that are stored on the disk you're currently working with. It looks like this ...

DIRECTORY Drive A
PRACTICE.DOC SAMPLE1.DOC SAMPLE2.DOC

3.3 OPEN FIRST.DOC

Let's type something right away. If your keyboard has auto repeat--see "REPEAT" in section 1.5--you'll have to type with a very light stroke. But you'll adapt pretty quickly, so let's go ahead. The worst you can do is not make mistakes!

The first step is **opening the document.**

There are 2 parts to this step. You have to select the function, then you have to name the document. So, to begin, type ...

>>> **d**

(Yes. Although the menu shows an upper case "D," you can type a lower case "d.")

OPENING

>>> Look at the top of the screen, in the far left corner. If you typed the **d** correctly, there will be a "D" there.

There will also be a prompt onscreen, beginning with these words ...

Specify the name of the document . . .

If you typed another letter, some other message may be onscreen. Don't try to figure out what happened. Don't try to correct your mistake. Just **GET OUT** by pressing your computer's **RESET** button. Then you'll have to load Newword again. Refer to section 2.12 for help.

>>> If you typed the "d" correctly, read the paragraphs on the screen.

After them is a line with just one word ...

Document?

The "d" you typed was a command. It stands for "Create or change a document." Newword knows what you want to do now, which is to open a document. But it needs a name for the document. So it asks, "Document?"

There's also a listing of commands on the screen. It begins with the heading, "WHILE ENTERING." Ignore it for now.

Type this name ...

>>> **first.doc**

>>> Read what you just typed. Does it look exactly like the name above? Is there a period between "first" and "doc"?

>>> If what you typed is okay, press **RETURN.** **If it's not correct,** use the **BACKSPACE** key to erase your mistakes, retype the line, and press **RETURN.**

Now Newword will look on the disk for a document named "first.doc."

But Newword won't find one. Not this time, because you haven't typed it yet.

So Newword has to ask you another question. It's displayed beneath the "WHILE ENTERING" message ...

```
Cannot find it. Create a new one (Y/N)? []
```

This question is a doublecheck against mistakes. You want to tell Newword, "Yes, you're right. There's no document called FIRST.DOC on this disk. And, yes, I want to create one."

So now you should enter ...

```
>>>          y
```

(Here, you do **not** have to enter <cr>. This is true for all prompts that require a **yes** or **no** response.)

If you entered this last command correctly, the screen will change. At the top will be a new menu, labeled, "Edit Menu."

If you made a mistake, another menu might be there, or maybe the screen won't change at all. It depends on what you actually did. But don't worry. Just **GET OUT,** by pressing the **RESET** button. Load NEWWORD, and begin again.

3.4 TYPING FIRST.DOC

Once you're safely at the Edit Menu, Newword is ready for you to type the document called FIRST.DOC.

You're probably curious about the menu. It lists some of the commands you'll be learning. But we think it's best if you not learn them right now. They'll make more sense **after** you've typed a document.

However, before you start, there's one important instruction we need to give. It's mostly for people who already know how to type, but everyone should know about it.

With Newword, **just as with a typewriter,** pressing the **SPACE BAR** will create a blank space in the text. That's how you make the spaces that separate words, and you should continue to use the **SPACE BAR** this way.

But, **unlike a typewriter,** do **not** press **RETURN** at the end of each line. Just keep on typing as if the line hasn't ended, and Newword will end the line for you.

The only time you press **RETURN** while typing in a document is to end the **last line of a paragraph** and to **add a blank line** (for example, between paragraphs).

>>> Okay, here's the document, FIRST.DOC. It has 5 paragraphs. Type it as best you can, but don't worry about mistakes. You'll correct them later ...

As you type FIRST.DOC, you're learning some of Newword's functions and commands. For example, did you remember that there's no shift lock on the keyboard? There may be a CAPS LOCK, but you still have to use the shift key to type the question mark. You must have noticed that when you reached the end of a line, the cursor automatically went to the next line. This is called "word wrapping." When you finish typing this sentence, press the RETURN key twice.

The first time you pressed RETURN, Newword ended the paragraph and sent the cursor to the beginning of the next line. The second time you pressed it, Newword created another line. To make a paragraph indent, use the TAB key or the SPACE BAR.

When the text reaches the bottom of the screen, Newword automatically rolls the entire screenful up underneath the menu, one line at a time.

(You're doing great!)

Now we're going back to the Opening Menu, to learn about the rest of the options listed there. Hold down the CONTROL key with one hand, and press k and then d with the other. The screen will change after you press the keys.

3.5 CLOSING FIRST.DOC

Pressing **CONTROL**, k and d has 2 results ...

The screen changes--FIRST.DOC and the Edit Menu are replaced by the Opening Menu. A third menu, the Blocking & Saving Menu, may flash briefly, too.

Newword stores FIRST.DOC on the disk, where it will stay for as long as you want it to. FIRST.DOC is now a file. If you knew how, you could read and edit the file, format and print it, copy it, move it to other disks or erase it.

Storing a document this way is the opposite of opening it, so we call the procedure, "closing." There are several ways to close a document; this one is called "saving."

If you haven't already done it, enter ...

>>> ^kd

If the Opening Menu doesn't come onscreen, it's because the keys you pressed weren't **CONTROL**, **k** and **d**. Here's what to do ...

Relax!

Ignore what's on the screen.

Find the **CONTROL** key on the keyboard. Hold it down while you look for the next key.

Find the **u** key. Press it once.

Let go of both keys.

Now find the **CONTROL** key again and hold it down. Keep holding it down.

Find the **k** and **d** keys.

Press the **k** first, and then the **d**.

Let go of all the keys.

This procedure will rescue you from most of the mistakes you could have made. If it didn't, there may be another procedure outlined in your other manuals. If not, you may have no other recourse than to press the **reset** button and start over.

For the time being, every time you close a document, use ^kd. Newword will go to the Opening Menu.

3.6 OKAY SO FAR?

Pause for a minute, stretch if you need to, and relax. At this point, it's easier to learn Newword if you take small steps.

Before typing FIRST.DOC, you looked at the Opening Menu. One of the documents listed in the directory there was SAMPLE1.DOC. It's actually a copy of FIRST.DOC that you can use in later exercises if your own copy isn't satisfactory. But **don't** try to look at SAMPLE1.DOC now.

You may want a longer break. If you do, press ...

>>> x

This will take you out of Newword completely; it's called "exiting." The operating system prompt, A>, should be at the bottom of the screen afterwards.

If you have floppy disks, it's a good idea to remove them--and store them properly.

If you have another kind of system, exit according to its procedures.

When you're ready to start again, load Newword. Refer to section 2.12 for help.

3.7 REVISITING THE OPENING MENU

Closing a document in **nw** always brings you to the Opening Menu, so that you can open another document, if you want to. As a reminder, here it is again ...

OPENING MENU

D get a document to change, or create a new document	**L** change logged drive
	E rename a document
N create or change non-document	**O** copy a document
P print a document	**Y** delete a document
M merge print a document	**F** turn directory off
C protect a document	**J** help
X all done with Newword (exit)	**R** run a program

DIRECTORY Drive A
FIRST.DOC PRACTICE.DOC SAMPLE1.DOC SAMPLE2.DOC

Look first at the directory, in the lower part of the screen. Take a bow. FIRST.DOC, your first document, is there. (It's okay if the names in your directory aren't arranged exactly like ours.)

Now, let's go back to the menu itself. It lists 13 functions. You've already used one of them to open FIRST.DOC. Let's check out the rest now.

>>> Recall that when you opened FIRST.DOC, you typed a "d," which is the letter to the left of the function description "get a document to change, or create a new document." This is always the way to choose the functions and commands listed in the menus, that is, just type the appropriate letters and keys.

Let's start with **help.** Press ...

>>> **j**

If you typed the "j" correctly, the first help message has appeared in the lower part of the screen. **If you typed something else,** you'll have to reset the computer and load **Newword** again. For the time being, resetting is the only way to recover from mistakes. We'll show you a better way in a few minutes.

Here's what the help message says ...

> **To get help with the menu above, type one of the letters that are shown to the left of each description.**
>
> **For menus that you might encounter later in your session with Newword, you may need to hold the control (CTRL) key down when you press the J key for help. On most terminals, you can also type the LINE FEED key and achieve the same results.**
>
> **To display and optionally change the help level, press the J key again.**

At this point, simply read the message. We'll learn about help levels later on. Although this particular message does not say so, when you finish reading a help message, you press the ESCAPE key to continue. So, press ...

>>> **ESC**

The Opening Menu will be back onscreen.

When you opened FIRST.DOC, we didn't want you to take the time to read the document prompt. Let's read it now, though. Type ...

OPENING

d

The document prompt explains a little bit about opening documents. It concludes with ...

Document?

This is a prompt, not a help message, because it asks you to respond by typing something. The last time, you typed "first.doc."

Notice the "WHILE ENTERING" message below "Document?." It shows how to correct typing errors made while responding to a prompt. You've been using the BACKSPACE key to erase mistakes. Well, in Newword, the DEL key works, too. You can also use ^h, that is, hold down the CONTROL key while typing h.

You can also cancel the whole operation by entering a different command ...

^u

This stops whatever's happening in mid-stream, and puts everything back the way it was.

We don't want to open a document now, so cancel the operation by pressing ...

>>> ^u

You're back at the Opening Menu. Let's get some more help with documents. Press ...

>>> jd

The same help message we saw before may flash briefly on the screen, then the document help message will be displayed. It begins ...

Newword is a word processor.

>>> Read the message, then press ESC.

In a minute, we want you to read the rest of the prompts and help messages on your own. Typing just a letter from the menu will bring the prompt; be sure to cancel the operation afterwards. Then type a j, followed by the letter in the menu, for the help message.

Try not to be confused by all these prompts and messages. In the first place, Newword displays the prompts automatically. So they'll come onscreen whenever you need them. As for the help messages, the main reason we want you to read them now is so you can find them later, when you really need them.

Don't worry about mistakes. Newword won't break. Most of the time, entering the unerase command, ^u, will fix things. Once in a while, though, Newword may flash an **error message** at you. It will tell you what's wrong and what to do about it. Since you're new to **Newword,** you may not understand the error messages. So try using ^u. If that doesn't help, you'll have to reset and load the program again.

One more thing, before you start reading. You may be wondering why, after you type, you sometimes have to press **RETURN** and sometimes don't. You'll catch on, after a while. But here are 2 general rules ...

Press **RETURN** after typing a response to a prompt. And press it at the end of a paragraph.

Do **not** press it after entering a **CONTROL** command.

>>> All right. Go ahead. Read the prompts and the help messages. Do it this way ...

First type one of the command letters in the menu.

Read the prompt that's displayed.

Then press ...

^u

This will stop the operation that pressing the command letter began.

Then type "j" followed by the same command letter.

Read all the help messages, pressing **ESC** at the end of each one.

OPENING

When the Opening Menu comes back onscreen, type the next command letter in the menu.
And so on.

When you finish, you should stay at the Opening Menu.

3.8 REVIEW

Opening Menu

* Computer menus and prompts--functions and information.
* Basic file maintenance.
* Directory of files.
* Files and documents.

Opening a Document

* Typing the command.
* Specifying the document.
* Answering the Y/N prompt.
* **GET OUT** by pressing the **reset** button.

FIRST.DOC

* Using the SPACE BAR and RETURN key.
* Typing upper case letters and indented lines.

Closing a Document

* Closing and saving.
* ^kd.

Revisiting the Opening Menu

* 13 functions in the menu.
* Getting help by typing j.
* Using the "WHILE ENTERING" commands to correct typing errors.
* Reading prompts and help messages.

4. EDITING

4.1 PREVIEW

In this chapter, we will:

DISCUSS THESE TOPICS and INTRODUCE THESE COMMANDS

4.2 EDIT MENU

Open FIRST.DOC. Use this sequence of entries. (Remember, <cr> means, "Press the **RETURN** key once.") ...

>>> d
first.doc<cr>

Every time you open a document, the **Edit Menu** will be onscreen as it is now.

```
........A:FIRST.DOC..P01.L01.C01.Insert.............................---  --

                              E D I T   M E N U

    CURSOR          SCROLL          ERASE       OTHER             EXTENSIONS
^E  up          ^W  up          ^G  char    ^J  help          ^O  on-screen forma
^X  down        ^Z  down        ^T  word    ^I  tab           ^K  saving & blocks
^S  left        ^R  up screen   ^Y  line    ^V  insert off    ^P  print controls
^D  right       ^C  down        DEL left    ^B  align parag   ^Q  quick function
^A  word left       screen      ^U  unerase ^N  paragraph line
^F  word right                              ^L  find/replace again

L----!----!----!----!----!----!----!----!----!----!-------------R
```

At the top of the screen is a row of periods, words and numbers. Then comes the menu itself. Then, just below the menu, there's a row of hyphens and exclamation marks.

4.3 STATUS LINE

>>> Look at the very top of the screen. That top line is called the **status line ...**

```
........A:FIRST.DOC..P01.L01.C01.Insert.....................--------
```

First, it shows the **document** you're working on, FIRST.DOC. It always shows the **drive,** as well. In this example, FIRST.DOC is on drive **A.**

Next comes ...

P01.L01.C01.

These letters and numbers are the **cursor indicator,** showing the cursor's exact position in the document, by Page, Line and Column.

Newword keeps track of pages for you, and displays the page numbers as P01, 02, 03, etc.

Newword counts the lines on each page, too, and displays them as L01, L02, L03, etc.

Finally, for each line, Newword also counts horizontal spaces. They're called **columns**, and they're displayed as C01, C02, C03, etc. Column **one** is at the far left of the line; there can be as many as 255 columns per line. When you first open a document, the cursor will always be on ...

Page 1

Line 1

Column 1

Next on the status line is the word "Insert."

When you type a character, Newword first makes room for the new character on the screen. It does this by "pushing" the existing text one space to the right. Then Newword displays the new character.

Insert may be turned **off**, causing Newword to replace the character under the cursor with the character you type. More on this later.

Next on the status line comes a string of periods and then 8 hyphens. The hyphens are the **fullness gauge.** Think of the fractions 1/8, 2/8, 3/8 and so on. Well, each of these hyphens represents one eighth. One eighth of what? One eighth of the computer's **memory.**

This is a little complicated, so bear with us.

When you're working on a document, Newword's working on it, too, but not at the terminal, like you. Instead, Newword works inside the computer's memory.

In **Newword**, documents (and non-documents, for that matter) can be any size. However, if they're larger than memory, editing can be inconvenient. So it's still best to keep them to about the same size as memory. Here's why.

This will be a little technical, so we'll go slowly. When your document fills up the computer's memory, Newword takes some of the overflow out of memory entirely. It stores the overflow on the disk, in a temporary file, leaving the part of the document you're currently working on in memory.

Now, if you want to go back to the part that's stored on the disk, Newword once again has to make room in memory, by moving what's there to a second temporary file. Then it has to move what's in the first temporary file back into memory.

All this shuttling back and forth takes time. Exactly how much time, depends on your computer, because it's actually your computer that does the work. In any case, editing a long document can be frustrating, because of the waiting that's involved.

So keep your files to a manageable size. When the fullness gauge shows that memory is full, close your document. Don't be concerned about breaking up a large document into several smaller files. In chapter 12, we'll show you how to put files together.

For example, if your document has **15,000** characters and there's space in memory for **30,000**, your document is **half** the size of the remaining space. The fullness gauge will show **4/8** full ("4/8" is the same as "half"). It will look like this ...

```
========
```

The first **4** hyphens will be replaced with equals signs, showing that your document represents **4/8** of the space available in memory.

When the gauge shows **7/8**, your document is plenty large enough. Save it, the way you saved FIRST.DOC, and start another document.

Other information appears on the status line from time to time. We will go over this later.

4.4 EDIT FUNCTIONS

Most of the Edit Menu is a list of functions and commands. You'll use them to move the cursor around the screen, scroll the screen (that is, roll it up and down), and make erasures.

In the menu, the functions are called "CURSOR," "SCROLL" and "ERASE."

In the OTHER column is a list of commonly used, related commands. The last column, EXTENSIONS, lists 4 additional menus.

4.5 RULER LINE

At the bottom of the menu (not the bottom of the screen), is the **ruler line**, showing left and right margins and tab stops (the exclamation marks). The line is 65 spaces long, and the tabs are set every 5 spaces, at 6, 11, 16, 21 and so forth.

You'll learn about the ruler line in another chapter.

4.6 HELP MESSAGES

There's one more thing to show you in this menu. In the OTHER column, the first command is ...

^J

You'll recall that the caret (^) stands for the CONTROL key. So hold the CONTROL key down while you type ...

>>> j

(Pressing just the **LINE FEED** key works, too.)

Every menu has a series of help messages like the ones you read in the Opening Menu. The first of this menu's help messages should be onscreen now ...

> **To get help with the Edit Menu above, type one of the keys that are shown to the left of each description in the menu at the top of the screen. (Remember that "^" means you should use the control key.)**
>
> **For a general explanation of the screen, type a question mark (?).**
> **For help with dot commands, type a dot (.).**
> **For help with saving your work, type ^KD.**
>
> **If you would like to change the help level, type ^J again.**

On your own, you're going to read the help messages that explain about the screen. They'll talk about the parts of the menu that we just went over, along with some that aren't onscreen right now. You'll have to

press the **ESCAPE** key to move from one message to another. To get started, type ...

>>> ?

>>> Read the message and then press **ESC**. Keep reading and pressing **ESC** until the messages are gone.

Then read about the commands in the menu. Here's how to do it ...

>>> Hold down the CONTROL key and type **j**.

Hold down the CONTROL key again and type **e**.

This brings the help message about the ^e command ...

You can move the cursor up one line with ^E.

Newword will try to position the cursor in the same column that it was in on the old line. However, if the cursor is to the right of the end of the line you're moving up to, the cursor will go to the end of that line.

>>> When you finish reading the message, press **ESC**.

Then press ...

>>> **^j**
^x

This brings the help message about the ^x command.

>>> Read all the messages. Press **ESC** after you finish each one. Then, to read the next message, enter **^j** and the command. Go through all 5 columns of commands.

When you finish, go on to the next section below.

EDITING

4.7 CORRECTING FIRST.DOC

You're going to practice the basic editing commands by proofreading and correcting FIRST.DOC.

Your version of FIRST.DOC probably isn't the same as ours. If there are typing mistakes here and there, that's good, because it means you'll have real errors to correct.

But if your document is, well, a mess, it might be better for you to work with a new one that's letter-perfect. There's one in SAMPLE1.DOC. However, we **don't** want you to open SAMPLE1.DOC. We want you to **copy** it, instead, and to work with the copy. By copying, you ensure that there'll always be a good version of SAMPLE1.DOC on your disk.

Copy SAMPLE1.DOC only if your FIRST.DOC is a mess.

To copy SAMPLE1.DOC, close FIRST.DOC with these keystrokes ...

^kd

At the Opening Menu, type ...

o

Newword will respond with a brief description of copying. This prompt will be beneath it ...

Document to be copied?

You should type the following. Proofread and correct the entry before pressing **RETURN** ...

>>> **sample1.doc**

(Proofread!)

<cr>

Newword will display a second message ...

Name a document to hold the new copy.

The new copy must have a different name. Type this ...

>>> **correct.doc**

(Proofread!)

<cr>

Newword will make the copy. While it does, it displays another message, beginning, "You can stop . . .". When the copy is made, the Opening Menu will come back onscreen.

Finally, open CORRECT.DOC with this sequence of entries. Don't forget to proofread ...

d
correct.doc
<cr>

4.8 MOVING THE CURSOR

>>> In the Edit Menu, look at the commands listed under CURSOR ...

CURSOR

^E up
^X down
^S left
^D right
^A word left
^F word right

>>> Now locate the corresponding keys on your keyboard. You'll notice that they form a **diamond.** You should also notice that the top key in the diamond is the one that moves the cursor up, and the bottom key, the one that moves it down. The keys that move the cursor to the left and right are arranged with the same symmetry ...

^e

^a ^s ^d ^f

^x

UP

LEFT WORD LEFT CHAR RIGHT CHAR RIGHT WORD

DOWN

>>> Practice the cursor commands for a while, on your own. The cursor diamond is arranged so that, on most keyboards, you can leave the little finger of the left hand on the CONTROL key, while using the other fingers of that hand to press the command letters.

Work slowly. If you hit the wrong key and something terrible seems about to happen, press ^u.

4.9 ERASING & INSERTING

Look at the Edit Menu again. Under **ERASE** are listed these commands ...

ERASE

^G char
^T word
^Y line
DEL left
^U unerase

There are two ways to make corrections with Newword: with **Insert on** and with **Insert off.** Which method you use depends in part on the kind of correction you need to make, but at this point, let's concentrate on making corrections with Insert on. This involves 2 operations ...

Erasing errors, **and then**

Inserting tne correct characters.

Let's walk through these operations. As we do, Newword will automatically scroll, that is, roll, FIRST.DOC (or CORRECT.DOC) whenever it has to.

Learn these basic commands, taking all the time you need. Don't worry if, when you add words, the text extends into the right edge of the screen. In the next part of this chapter, you'll learn how to take care of that.

Erasing

>>> Move the cursor to the **2nd line**, the one that begins, "functions." Move the cursor along the line, to the first letter of "remember." Be sure it is **on** the "r," like this ...

[r]emember

You can erase the whole word. Press ...

>>> **^t**

Easy, wasn't it? Notice that the word after "remember" has moved into the space where "remember" was.

Inserting

Now we have to put another word in the sentence: "forget." Without moving the cursor, type ...

>>> **forget**

You have to add a space between "forget" and "that," as well. Be sure the cursor is on the "t" in "that" ...

[t]hat

>>> Then press the **SPACE BAR** once. Pressing the **SPACE BAR** inserts a space.

Practice these commands by correcting your typing errors. Change the document a little, too, by adding a phrase here and there. (Don't worry about extra-long lines.) You have all the time in the world, so don't rush, and don't be anxious.

Keep in mind that Newword ...

erases to **e**ither side of the cursor; **but**

inserts characters only to the left of it.

Check out the difference between ^g and **DEL**. With ^g, the cursor stays in place, "scooping" characters and spaces in from the right, whereas with **DEL**, the cursor moves, "squashing" the characters and spaces to its left.

(There's another erase command, one that's not in the menu. It's ^h, and it works just like **DEL**.)

Practice using ^u, the **unerase command,** too. Erase a character with ^g, then press ^u. The erased character will reappear.

Do the same with the other erase commands--^u will restore whatever's been erased. The erased characters reappear **wherever the cursor is.** So if you've moved the cursor, when you press ^u the restored characters won't come onscreen in their original place.

4.10 ALIGNING PARAGRAPHS

Compare your document with the original version of FIRST.DOC in section 3.3. Look carefully at the right margin. Your margin isn't always the same as the original's. That's because you've been adding and removing characters from some of the lines.

As we explained before, some of your lines may even extend to the right edge of the screen. Here's how to fix that.

In the OTHER column of the Edit Menu is the **align paragraphs** command, ^b.

>>> Move the cursor to the beginning of the first line of the first paragraph. Put it on the "A" in "**[A]**s."

Next, check the cursor indicator. Is the cursor in column **one?** If it is, the indicator will display "**C01.**" If it isn't, move both the cursor and the word "As," to column one ...

To move to the **left,** use the **DELETE** key.
To move to the **right,** use the **SPACE BAR.**

Notice that the word "As" moves with the cursor. This is one way to move characters.

Once the cursor and the "A" are in column one, enter ...

>>> **^b**

If any of the lines in the first paragraph were too long or too short, Newword has just rearranged them. If the lines were okay, the paragraph hasn't changed at all.

>>> Align the rest of the paragraphs. Put the cursor **on the first letter** of each paragraph. Then move both to column one. For indented paragraphs, move them to column 6. Then press **^b**.

If you made only a few minor corrections to your document, Newword may not have to change the paragraphs at all. That's fine.

4.11 SCROLLING

There's one final set of commands in the Edit Menu that we haven't used, the scrolling commands.

Newword's done all the scrolling, so far. Now it's your turn. There are 4 commands listed in the menu under SCROLL ...

SCROLL
^W up
^Z down
^R up screen
^C down screen

>>> These commands move the text, not the cursor. ^W and ^Z move the text one line at a time; ^R and ^C move the text a screenful at a time.Practice the commands a few times; notice that, after ^R and ^C, the cursor is at the middle left of the screen.

And here's a wonderful variation on two of these commands: add ^q. Press ...

>>> **^qc**

(Hold down the CONTROL key while you press the "q" and the "c." All 3-keystroke commands can be entered this way.)

Newword went to the very end of the document. Now press ...

>>> **^qr**

You're at the document's beginning.

4.12 TIME OUT FOR PRACTICE

>>> Take time for more practice with all these commands. Type a few lines at the bottom of your document, then revise them. Pay special attention to where the cursor goes after each command.

When you feel you've had enough practice, proofread the document one last time, comparing it with the original version of FIRST.DOC in chapter 3.

Before you close the document, look for a minute at the extreme right of the screen. There are a few arrows there (<). They're one of the flag characters you read about when you reviewed the help messages earlier in this chapter. Newword put the arrows on the screen, to indicate where you entered carriage returns. Flag characters are explained in chapter 8.

Now, close the document by entering ...

>>> **^kd**

Don't be distracted by the changing menus at the top of the screen.

When the Opening Menu is onscreen, type ...

>>> **x**

This is the exit command; it takes you out of Newword, to the operating system. The prompt, A>, and the cursor should be at the bottom left of the screen.

Remove and store any floppy disks you have.

4.13 HOW ARE YOU DOING SO FAR?

The **Do It Yourself** lets you learn Newword at whatever pace suits you best. So if you're feeling a bit confused and overwhelmed, just slow down!

If it's been smooth going for you so far, terrific! But take an eye break, anyway. A 15-minute rest, every hour or 2, prevents eye strain and headaches.

4.14 REVIEW

Edit Menu	* Status line. * Functions and commands. * Ruler line. * Help messages.
Edit & Cursor Commands	* Correcting FIRST.DOC. * The cursor diamond. * Erasing characters, words and lines. * Aligning paragraphs. * Scrolling. * exiting from Newword to the operating system.

EDITING

5. FORMATTING

5.1 PREVIEW

In this chapter, we will:

DISCUSS THESE TOPICS and INTRODUCE THESE COMMANDS

CHANGING DRIVES. L
ON-SCREEN FORMAT MENU
BLOCKING AND SAVING MENU
SAVING . . . . . ^KD, ^KS, ^KQ, ^KX
MODIFYING THE RULER LINE
TABS . . . . . ^OI, ^ON
MARGINS. . . . . ^OL, ^OR
LINE SPACING . . . . . ^OS
CENTERING. . . . . ^OC
WORKING WITH EXTRA-LONG LINES, HORIZONTAL SCROLL

5.2 WHAT'S NEXT?

Word processing doesn't require extraordinary skills or knowledge. Finding your way to a document and then finding your way around the screen--that is, moving the cursor--are the hardest things to learn. And they're hard simply because they involve new concepts of what "writing" and "typing" are.

So, in some ways, the rest of the **Do It Yourself** will be easier. We'll just show you more of Newword's functions and commands. Learn them at your own pace, and don't be concerned when you make mistakes. It's important that you focus on what you're learning, not on your latest goof.

When you do make mistakes, figuring out what you did wrong probably isn't worth the effort. Just cancel a wrong command with ^u, or erase an incorrect entry, whichever applies, and do the task again.

You're going to learn about formatting next, by creating another document, a letter. You'll clear and set tabs and type indented lines. Afterwards, you'll learn a few more formatting commands.

If you don't already know how to type, some of the formatting terms will be as new to you as everything else in Newword. We won't explain them as carefully as we explain most other things, because, as we said in chapter 1, the **Do It Yourself** assumes that you know how to type. We had to draw the line on explaining things somewhere, and that seemed a good place to do it.

So, when you come to THIRD.DOC in this chapter, slow down. You may even have to go over this part of the chapter a few extra times, before everything's really clear.

5.3 CHANGING DRIVES

>>> You should be at the Opening Menu; if you're not, go there now. If you've forgotten how, here's a reminder ...

Newword goes to the Opening Menu every time you **save a document** with ^kd; **and**

it goes there every time you **load Newword** from the operating system.

How many floppy disk drives does your computer have? If it has more than one, let's put a disk in drive **B** and change the logged drive, just to learn how.

Keep the Newword working disk in drive **A**, and insert a properly formatted disk in drive **B**. Chapter 2 explains working disks and formatting. (If you have a disk in drive B already, that's great.)

(To format a new disk right now, exit from Newword, format the disk, and then come back to the Opening Menu. The new disk needn't have any files on it, neither the files supplied on our distribution disk nor any of your documents. It's okay if it does, though.)

At the Opening Menu, be sure the new disk is in drive **B.** Then enter this sequence of commands ...

```
>>>     l     (This is an "L," not a one.)
        b<cr>
```

Every time you put a new disk in a drive, you have to log it with this procedure, i.e., enter the logged drive command at the Opening Menu, and then enter the drive letter.

If you have one floppy drive and one hard drive, the command sequence above will work for you.

If you have only one drive, don't worry. One drive is fine. However, some single drive systems have **logical drives.** If yours does, you may be able to change to drive **B,** too. Check chapter 2 and your hardware manuals, first.

5.4 ON-SCREEN FORMAT MENU

Open the new document, now, by entering ...

```
>>>     d
        SECOND.DOC<cr>
        y
```

Look first at the status line. Notice that it says, "B:SECOND.DOC." Because you changed the logged drive, Newword opened the new document on drive B, not A.

Before typing SECOND.DOC, you'll change the ruler line, with commands in the On-screen Format Menu. The margins will stay the same, but not the tab stops.

You learned about the On-screen Format Menu when you were reading help messages in chapter 4. Refresh your memory by reading its help message again. Press ...

```
>>>     ^j
        ^o
```

The new menu is at the top of the screen now, and this help message is below it ...

You can change various things related to the format of your document using ^O. After ^O is typed, you must then type again to change something. If you don't remember what your options are and the help level is set to 2 or 3, you can wait a couple of seconds for the On-Screen Format Menu to appear. If you have typed ^O accidently, you can type the space bar to avoid changing anything.

Type one of the letters shown to the left of a description in the Format Menu at the top of the screen.

>>> Read the message, then press the **SPACE BAR**. The Edit Menu will return to the screen.

Now read about **tab stops**. Press these sequences and read the help messages as they appear ...

>>> **^j**
^o
i
ESC

>>> **^j**
^o
n
ESC

5.5 MODIFYING THE RULER LINE

Okay. Let's get going. First, find the ruler line. It shows ...

```
L----!----!----!----!----!----!----!----!----!----!--------R
```

L and **R**	Left and right margins
!	Tab stops
-	Columns

We will now clear some tab stops. Press ...

>>> **^on**

After flashing the On-screen Format Menu, Newword has

displayed another message about tab stops. This is **not** a help message; it's a prompt.

You learned, in the Opening Menu, that some commands require more information than the command alone. When that's the case, Newword shows you a prompt explaining how to enter the information. Sometimes the prompt has other information, too, like this one ...

> **Current tabs: 6 11 16 21 26 31 36 41 46 51 56**
>
> **Decimal tabs: None**
>
> **You can clear a tab stop by typing its column number (for example, "15"). For decimal tabs, precede the column number with "£" (for example, "£15").**
>
> **All the stops can be cleared if you type "A," instead of a number.**
>
> **Press the ESCAPE key (ESC) if the cursor was at the tab to be cleared.**
>
> **Tab stop to be cleared? []**
>
> **WHILE ENTERING: DEL or ^H erases a mistake. RETURN when done. ^U safely cancels in midstream.**

This is a long prompt, because there's quite a bit that needs to be said. Let's go through it carefully. You'll find that, along with new information, it also repeats some of what you read in the help messages.

The first thing it does is list all the existing tab stops. (Decimal tabs are special ones, for typing columns of numbers. If you do this sort of typing, read about tab stops in the **Newword Encyclopedia.**)

Next, the message explains how to clear a tab stop, that is, how to make it inactive. (You can think of "clearing" as erasing the tab stop, even though the stop isn't really erased.)

In a document, a line can have as many as 255 columns, or spaces, starting at the left edge of the screen and extending to the right. You can put a tab stop in any of these columns, and, of course, you can clear any tab stop.

Remember, too, that Newword displays the cursor's position, by Page, Line and Column, in the status line.

So, to clear a stop, you can ...

Enter its column number and <cr>; **or**

Enter **a** and <cr>, to clear **all** the tab stops; **or**

Press **ESC**, to clear the stop the cursor was positioned at when you entered ^on.

You can correct typing mistakes, too, with the commands list in the "WHILE ENTERING" message.

To change the ruler line for SECOND.DOC, you first need to clear **all** the tab stops, so type ...

>>> a<cr>

The Edit Menu has returned, and the ruler line should look like this ...

L----------R

The cursor is probably directly beneath the ruler line, at the far left of the screen. But it doesn't really matter where the cursor is.

Let's set 3 tab stops.

Press ...

>>> ^o

The On-screen Format Menu has returned. Press ...

>>> i (This is an "I," not an "L.")

This time, the "set tab stops" prompt appears. It's like the previous "clear tab stops" prompt. Read it. Then enter ...

>>> 6<cr>

The ruler line should have one exclamation mark on it, 5 spaces in from the left.

```
L----!-------------------------------------------------------R
```

Enter ...

```
>>>          ^oi
             11<cr>   (This is the number eleven.)
```

Another exclamation mark should be in the ruler line, 5 spaces to the right of the first one.

Now enter ...

```
>>>          ^oi
             31<cr>
```

You've made a new ruler line, with the same margins as before, but only 3 tabs. It should look like this ...

```
L----!----!-----------------!-------------------------------R
```

If yours doesn't, go back to the beginning of **MODIFYING THE RULER LINE** and start over.

When your ruler line has tabs at columns **6, 11** and **31,** congratulate yourself!

By the way, you don't always have to clear all tab stops in order to set new ones. We did it here only because that was the most convenient way to get rid of all the stops we didn't need.

Take a moment to relax . . . then go on.

5.6 HERE'S SECOND.DOC

We're going to show you the new document **twice.** The **first time,** just look it over. **Do not** try to type it.

The **second time** we show it, we will also give you instructions for the typing. Here's an example of an instruction ...

Press <cr> 3 times.

As you'll see, the typing instructions break up the document's visual continuity. So we thought it best to show you what the final product should look like, before we showed you the instructions.

We've also included a copy of SECOND.DOC on the Newword distribution disk. It's listed in the directory for drive A as SAMPLE2.DOC. SAMPLE2.DOC will give you a fresh copy of the following document, should you need it later.

Don't let all this scare you. We're just doing our best to anticipate problems ...

Oakenshield Hall
Gondor, Missouri
April 1, 1983

Samwise Gamgee
Mayor of the Shire
Bag End, Kentucky

Dear Sam:

I was more than pleased to visit 100 Aker Wood for you. I believe you're right. The spirit of the folk there is so much like our own, they must be from the Old Land.

I contacted Mr. Sanders, the leading businessperson there. He gave me this list of goods for trade:

Honey	40 barrels
(His honey really is scrumptious!)	
Pine cones	7 bushels
Drawings	Assorted pen-and-pencil sketches of life in 100 Aker Wood
Heffalump Traps	1

As you see, Sam, these people are quite innocent. Leg and I have agreed to keep their whereabouts a secret, from all but friends as trustworthy as you, til a better time. Let us all hope that time is soon.

Well, as always, I'm asked to send you warmest greetings from everyone here in Gondor.

Yours very sincerely,
in deepest friendship,

Gimli

[This letter is based on J.R.R. Tolkien's The Lord of the Rings fantasy novels (Ballantine Books, Inc., New York, 1965; copyright 1937, 1938, 1965, 1966 by J.R.R. Tolkien), and A.A. Milne's Winnie-the-Pooh (E.P. Dutton & Co., Inc., New York, 1961; copyright 1954 by A.A. Milne).]

5.7 TYPING SECOND.DOC

>>> Okay, let's do it. Typing instructions are underlined, as we said they'd be. Work slowly, correcting mistakes as you go along. (The cursor and edit commands are in the Edit Menu onscreen.) ...

Press **RETURN** only where you see the carriage return symbol <cr>.

Now, move the cursor by pressing the **TAB** key 3 times.

Oakenshield Hall<cr>
Gondor, Missouri<cr>
April 1, 1983<cr>
<cr>
<cr>
<cr>
<cr>
<cr>
<cr>
The 6 carriage returns made 6 blank lines.

Samwise Gamgee<cr>
Mayor of the Shire<cr>
Bag End, Kentucky<cr>
<cr>
Dear Friend Sam:<cr>
<cr>

Press the **TAB** key or ^i once, to make paragraph indents.

I was more than pleased to visit 100 Aker Wood for you. I believe you're right. The spirit of the folk there is so much like our own, they must be from the Old Land.<cr>
<cr>
I contacted Mr. Sanders, the leading businessperson there. He gave me this list of goods for trade:<cr>
<cr>

Use the first and third tab stops for the next line.

Honey 40 barrels<cr>
<cr>

Use the second tab stop for the next line.

FORMATTING

(His honey really is scrumptious!)<cr>
<cr>
Use the first and third tab stops for the next lines.

Pine cones 7 bushels<cr>
<cr>
Drawings Assorted pen-and-pencil sketches<cr>
of life in 100 Aker Wood<cr>
<cr>
Heffalump Traps 1<cr>
<cr>
As you see, Sam, these people are quite innocent. Leg and I have agreed to keep their whereabouts a secret, from all but friends as trustworthy as you, til a better time. Let us all hope that time is soon.
<cr>
Well, as always, I'm asked to send you warmest greetings from everyone here in Gondor.<cr>
<cr>
<cr>
Yours very sincerely,<cr>
in deepest friendship,<cr>
<cr>
<cr>
<cr>
Gimli<cr>
<cr>

5.8 HOW DID YOU DO?

>>> Proofread your version of the letter. Does it look like ours?

If it doesn't, take a deep breath, clear your mind, and work on it some more.

Work slowly and try to stay relaxed.

And don't forget, if worse comes to worst, there's a fresh copy of the letter in SAMPLE2.DOC.

5.9 BLOCKING AND SAVING MENU

In chapter 4, you read all of the Edit Menu's help messages, except the ones about saving documents and dot commands. Read the one about saving now. Press ...

>>> ^j
^kd

The **Blocking and Saving Menu** has come onscreen. At this point, we will discuss only the 4 commands listed under "SAVING" ...

S save & resume edit

D save document

X save & exit Newword

Q quit without saving current work

So far, you've used just one of these commands, ^kd.

^ks is a precautionary measure. When you're working on a document for a long while, it's wise to save it occasionally with ^ks. Think of it as keeping the saved version of the document up-to-date.

Until you're more used to word processing, you may accidentally erase a document now and then. If you've been updating the saved version with ^ks, erasing the document won't be a catastrophe. You'll have lost only what was created since the last update.

^ks also protects you against the computer's mistakes. Yes, computers goof, too. When they do, you can count on losing your current work. So update the saved version of your document . . . **often.**

How often is often? Well, how much of your current work do you want to retype, after it's been lost? If, in your situation, it's easy to retype 5 pages, then update with ^ks every 5 pages. If a page or 2 is more appropriate, update every couple of pages.

Because you use this command when you want to keep on working on your document, not when you're finished with it, Newword stays in the document--it doesn't go to the Opening Menu.

The other command, ^kq, is exactly the opposite of ^ks. Your current work is **not** saved, and Newword **does** go to the Opening Menu.

If the work you've been doing isn't worth saving, just **quit** it, with ^kq. Newword will close the file, but it won't save the current work. If the document's an old one, you'll still have the old version that's on the disk. If the document's new, Newword won't save anything at all. The document won't even appear in the directory. If you should type ^kq by accident, and have made changes to the file, Newword will ask you to verify that you actually want to quit.

In the past, when you have finished your editing session with Newword, you have saved your document with **^KD**, and then, at the Opening Menu, pressed **X** to get to the operating system. The command **^KX** is a shorthand combination of these two functions. After saving your document, Newword returns you to the operating system without displaying the Opening Menu.

Now save SECOND.DOC. Press **ESC** to clear the help message, and then enter **^kd.**

5.10 THIRD.DOC

We're going to go over more formatting functions now. Think of them as changing the **shape** of a document. (This is the part of the chapter that may be a little harder than usual for those of you who don't know how to type.)

You know one of the functions, tab stops. There are 3 others ...

MARGINS TYPING DISPLAY

We'll look at MARGINS and TYPING now. To do this, open a new document. Enter this sequence ...

>>> **d**
third.doc<cr>
y

Before you type the document below, read the help messages about margins, line spacing and centering. Use these sequences of commands ...

```
>>>     ^j
        ^o
        l    (This is an "L.")
        ESC

>>>     ^j
        ^o
        r
        ESC

>>>     ^j
        ^o
        x
        ESC

>>>     ^j
        ^o
        s
        ESC

>>>     ^j
        ^o
        c
        ESC
```

5.11 MARGINS

>>> Okay. Go ahead and type the following paragraphs. When you see a caret (^), type the symbol--**don't** press the CONTROL key ...

Changing margins is just like changing tab stops. You can ...

enter ^ol, the left column number and <cr>; and

enter ^or, the right column number and <cr>.

Or, you can ...

move the cursor to where you want the new left margin to be, and press ^ol and ESC; and

move the cursor to where you want the new right margin to be, and press ^or and ESC.

That's all there is to it. You can change both margins, or just one.

There's a margin release, too. It's ^ox. "Mar-Rel" is displayed in the status line when it's activated.

Use the margin release for typing beyond either margin. Once you move the cursor back inside the margin, Newword will turn it off automatically.

To be sure you understand these commands, let's practice a little. Enter this sequence ...

>>> ^ol (This is an "L.")

After flashing the On-screen Format Menu, Newword has displayed a prompt about margins. Read it; then enter ...

>>> **14<cr>**

>>> Look at the ruler line. It begins at column **14** now.

Next, enter this sequence. Newword will display the same margin prompt again. Read it before typing the new column number ...

>>> ^or

(Read the prompt!)

54
<cr>

>>> Look at the ruler line again. It ends at column **54.**

>>> Now add a blank line at the end of THIRD.DOC, by moving the cursor to the end of the document and pressing **RETURN** twice.

Then add this paragraph to the ones you just typed ...

>>> **Notice, too, that there are still tab stops within the margins. But the tabs are in the columns they were in before you changed the left and right margins.**

Tabs don't change unless you want them to.

>>> Move the cursor to the beginning of THIRD.DOC. Read the whole document. Make sure you understand everything before going on.

5.12 LINES

There are just 3 more things about formatting we want to show you at this time.

First, let's change the line spacing.

Enter this sequence ...

>>> ^os

Read the new prompt that's onscreen; then enter ...

>>> 2<cr>

Note that the status line now shows "Spacing-2," as a reminder.

Now, at the end of THIRD.DOC, add another blank line and this paragraph ...

>>> To center text on a line, move the cursor to the line. Anywhere on the line will do. Then press ^oc, and Newword will move all the text to the center of the page.

Let's practice centering.

>>> Move the cursor to the beginning of THIRD.DOC. Use the RETURN key to make 2 blank lines **before** the first paragraph. Then move the cursor to the beginning of the first blank line.

Type the following title, and then enter the command ...

>>> CHANGING MARGINS & LINES
^oc

The title's now centered between the left and right margins, which are presently at columns 14 and 54. To center the title between columns one and 65, you would have to change the margins to those numbers.

The last thing we'll show you is that Newword can scroll text **sideways** as easily as up and down.

Normally, columns 1-80 are displayed on the screen. But when you have to type lines more than 80 spaces long, the display automatically shifts to the right for you, **20** columns at a time. This allows you to see the line as you type it.

We could show you this by having you reset the right margin to, say, 150, and then type a long line. In fact, if you want to, you could do that right now. Just make up your own sentence and type it.

There's another way to show you, though, one that also teaches you something new about erasing.

>>> Move the cursor to the last paragraph. Put it at the end of the paragraph's first line, **after** the "e" in "the," like this ...

To center text on a line, move the []

You can connect the first line of text with the second line, now, by erasing the spaces that separate them. Press ...

>>> **^t**
^t
^t

See? The second line of text is now connected to the first line.

Notice, too, that Newword has put a plus sign (+) on the screen, at the extreme right. That's called a "flag character." It indicates that part of your text extends beyond the right margin. Flag characters are explained in chapter 8.

>>> To see Newword scroll the text sideways, move the cursor to the right, one word at a time. Use ^f. When the cursor passes column 80, the text display will shift.

>>> Keep moving the cursor, carefully, all the way to the end of the line. Press ^t 3 more times, so that the third line of text is connected to the second. Move the cursor to the end of the line again, and watch the display shift every time the cursor passes the 20th column.

That's all there is to long lines. Newword scrolls the text sideways, whether you're typing new text or joining several old lines together. Just remember that, should you have to type long lines, it's very easy to do--set your margins and type.

Let's restore the paragraph to the way it was. All you have to do is align it.

Make sure the cursor is on the long line. It doesn't matter which part it's on. Then enter ...

>>> ^b

>>> Now that the paragraph's back the way it was, let's restore the margins and line spacing, too. But let's do it the easy way, by saving THIRD.DOC and exiting from Newword.

When you return to Newword and open a document, the full, 65-space ruler line will be onscreen, and line spacing will be back to one.

5.13 REVIEW

Changing Drives	* Changing the logged drive to **B**.
On-screen Format Menu	* Four format functions. * Help messages and prompts. * Clearing tab stops. * Setting tab stops.
SECOND.DOC	* Typing SECOND.DOC. * Using <cr> to end short lines, and to make blank lines. * Typing indented lines with tab stops.
Saving Menu	* Three ways to save a document. * Saving SECOND.DOC with **^kd**.
THIRD.DOC	* Changing margins. * Changing line spacing. * Centering text on a line. * Managing long lines.

FORMATTING

6. FINDING

6.1 PREVIEW

In this chapter, we will:

DISCUSS THESE TOPICS and INTRODUCE THESE COMMANDS

FINDING WORDS.	^QF
FINDING AND REPLACING.	^QA
FINDING A PAGE	^QP
FINDING AGAIN.	^L
ERASING A DOCUMENT	Y
ERASING PARTIAL LINES.	^QY, ^QDEL

6.2 OPENING QUICK.DOC

There's a document called PRACTICE.DOC on your Newword working disk in drive A.

You're going to make a **copy** of PRACTICE.DOC on disk A, so you'll need space on the disk, space of at least 6k. Use your operating system's **status** command to check for the space. At the operating system's prompt, A>, enter ...

>>> **stat<cr>**

If there's enough space on the disk, use these commands to load **nw** and copy PRACTICE.DOC ...

>>> **nw<cr>**

(Wait for the Opening Menu.)

```
o
practice.doc<cr>
quick.doc<cr>
```

If there's not space on the disk, you can make space by erasing other documents, but **don't** erase our documents. They're listed in section 3.1.

To erase a document, load **nw** and then type **y**, the **erase a document** command. The prompt that follows the command asks you to name the document that's to be erased. Just type its name, followed by a carriage return.

But, if you have another disk drive, instead of erasing documents on disk A, why not put a formatted disk in the other drive and copy PRACTICE.DOC there? The following sequence logs the new disk and copies PRACTICE.DOC ...

```
l   (This is an "L.")
b<cr>
o
a:practice.doc<cr>
quick.doc<cr>
```

Remember, when you put a new disk in your computer, you have to log the disk **before** you do anything else. Once you log the disk, Newword assumes that all the files you name are on that disk.

In this example, if you hadn't typed "**a:**" when you named PRACTICE.DOC, Newword would have looked for it on drive B. But, you didn't have to name the drive for QUICK.DOC, because you were putting that document on drive B.

6.3 QUICK MENU

Now open QUICK.DOC ...

```
>>>     d
        quick.doc<cr>
```

In the Edit Menu's help messages, you read about the Quick Menu. We'll look at it closely now.

FINDING

Read the help messages first, by entering ...

>>> ^j
^q
ESC

>>> ^j
^q
r
ESC

>>> ^j
^q
c
ESC

>>> ^j
^q
y
ESC

>>> ^j
^q
DEL
ESC

[You may read the help messages for the other commands on this menu if you wish. We will not, however, discuss them at this time.]

Notice the **SCROLLING** commands, ^qr and ^qc. We showed you how to use these when you corrected FIRST.DOC, in chapter 4. But it won't hurt to practice them again.

Since you've just opened QUICK.DOC, the cursor is at the beginning of the document. So press ...

>>> ^qc

The cursor has moved directly to the end of QUICK.DOC.

>>> To see just how much work ^qc has saved you, move the cursor back to the beginning of the document with ^r.

You can practice the **ERASING** commands, ^qy and ^qDEL, later, on your own.

6.4 FINDING A WORD IN A DOCUMENT

In the Quick Menu, there are 3 commands in the **FINDING** column ...

F find

A find and replace

P find a page

You can't leaf through a document in the computer the way you can leaf through a stack of typed pages. However, Newword has some good substitutes for leafing. One of them is scrolling, which you know about. Another is **finding,** which is sometimes better than leafing and scrolling combined.

First, read the **find, find and replace** and **find a page** help messages in the Edit Menu. Enter these sequences and read the messages ...

>>> **^j**
^q
f

(Read the message.)

ESC

>>> **^j**
^q
a

(Read the message.)

ESC

>>> **^J**
^q
p

(Read the message.)

ESC

Now be sure the cursor is still at the beginning of the document. Then enter ...

FINDING

>>> **^qf**

Newword is displaying the **find prompt ...**

Find what?

There's also a "WHILE ENTERING" message.

Let's find the word "wood." Enter this, being careful not to leave any spaces after the word ...

>>> **wood<cr>**

Now Newword is displaying another prompt; it shows the 5 **find options.** You'll practice them later. For now, just enter another **<cr>** and watch the text.

Newword is reading QUICK.DOC, looking for the word, "wood." In fact, it's probably found the word by now.

6.5 FINDING IT AGAIN

Newword's found "wood" and positioned the cursor there. To find "wood" again, enter ...

>>> **^l** (This is an "L," not a one.)

Presto!

You can find every instance of "wood," if you continue to press ^l. After the last instance, Newword will tell you ...

End of search for: wood

PRESS ESCAPE (ESC) KEY TO CONTINUE.

This means Newword has searched to the end of the document, and there are no more instances of "wood."

>>> Keep looking for "wood," by pressing ^l, until Newword displays the end of search message. Then press **ESC.**

6.6 REPLACING A WORD

Leave the cursor at the **end** of QUICK.DOC, and press ...

>>> **^qa**

The **find and replace** sequence begins with the same prompt as before ...

Find what?

Enter this; be sure to leave no blank spaces after the word ...

>>> **computer<cr>**

The next prompt is different ...

Replace with?

Enter ...

>>> **microcomputer<cr>**

Now Newword's displaying its **find and replace options.**

Usually, Newword searches from the beginning of the document to the end. But you're already at the end, so a normal search won't work. Instead, you'll have to instruct Newword to search backwards, i.e., to start at the end of the document and go to the beginning.

You can do this with one of the options that are onscreen. Use the **backwards** option--enter ...

>>> **b<cr>**

Newword is moving through the document backwards, i.e., from the end to the beginning. By now, it's found "computer," and it's waiting for permission to replace it.

At the very top of the screen, in the righthand corner

of the status line, is a new prompt, **Replace?**. The cursor is bouncing from **Replace?** to the word "computer" in the text, and back again.

Newword is asking for permission to replace "computer" with "microcomputer." Give permission, by typing ...

>>> **y**

Presto again!

Find "computer" again, by pressing ...

>>> **^l** (This is an "L.")

Newword has found "computer" a second time and is waiting for your decision about replacing it. This time, enter ...

>>> **n**
^l

>>> Because you didn't give permission to replace "computer," Newword went directly to the next occurrence of the word. This time, let Newword replace it with "microcomputer" (Type **y.**) and search again (Enter **^l.**).

That's right, "computer" occurs only 3 times in QUICK.DOC.

Press **ESC**, to end the search.

6.7 NEWWORD DOES AS IT'S TOLD

Find "idea" by entering this sequence ...

>>> **^qf**

Find what? **idea**

ESC

Pressing ESC lets you bypass the options altogether.

>>> After Newword finds the first "idea," enter ^l to find it again.

>>> Look carefully at the second "idea."

It isn't "i-d-e-a;" it's "i-d-e-a-**s**," with an "s." Newword can find anything, but ...

It finds **only** what you tell it to; **and**

it finds **everything** you tell it to.

To keep Newword from stopping at "ideas," we'll have to use another one of the options. Enter this sequence ...

>>> **^qr**
^qf
Find what? **idea**<cr>

Options? **w**<cr>

>>> As you did before, after finding the first "idea," press **^l**, to find it again.

And presto again! By selecting **w**, the **whole word** option, you instructed Newword to find "idea" **only** when it's a word by itself, **not** when it's part of a longer word.

Press **ESC** to end this search.

6.8 AUTOMATIC REPLACEMENTS

Now we'll use 3 options at the same time. Enter this sequence, and watch the screen carefully afterwards ...

>>> **^q**
a

Find what? **wood**<cr>
Replace with? **plastic**<cr>

Options? **bng**<cr>

Okay, what happened? Well, Newword simply followed your instructions.

You chose the **b** option, so Newword searched backwards.

You chose n, too. This option tells Newword, "Replace the word **without asking me for permission** each time you find it. Just go ahead and make the replacement."

And you chose g, which means, "Don't wait for me to enter ^l. Go right from one instance of the word to the next, automatically."

If you had chosen **only n**, Newword still would have replaced "wood" with "plastic," without asking you for permission each time. But it would have waited for you to enter ^l before going to the next occurrence of the word. But, since you also selected g, Newword didn't wait.

g has another effect. Normally, Newword starts a search from wherever you are in the document. If you're in the middle of it, that's where Newword starts. However, g causes Newword to go the the beginning of the document **before** it starts the search. g has 2 results, then ...

Don't wait for ^l.
Start at the beginning.

If you combine g and b, Newword starts at the end, instead of the beginning.

Because, with g, Newword moves automatically, you won't be able to align paragraphs or do anything else. The only way you can stop the search is by entering ^u--it cancels the entire operation.

(r does half of what g does--it causes Newword to go from one occurrence of a word to the next, automatically. But the search starts wherever you are in the document. Newword doesn't first go to the beginning, or end.)

6.9 ALIGNING PARAGRAPHS AFTER REPLACING A WORD

When you replace a word with a longer word, you have to align the paragraph, with ^b. Since "microcomputer" is a longer word than "computer," and "plastic" is longer than "wood," you'll have to align the paragraphs they're in.

You can do it after each replacement, or wait until you've worked through the entire document.

If you do align after each replacement, here's a tip: move through the document **backwards.** Otherwise, align-

ing could result in skipping over words you want to replace.

One way to avoid this problem is to do all your replacements before you align, and then align every paragraph you've made replacements in.

Align QUICK.DOC now. Go to the end of the document and then find "microcomputer" and align its paragraphs. Do the same with "plastic."

6.10 SELECTING OPTIONS

Here's a table of all the options ...

b Search backwards. Use this to search the part of the document that comes before where you are now.

g This does 2 things. First, regardless of where in the document you are, **g** causes Newword to go to the very beginning before starting to look for the word you entered. (If it's combined with the **b** option, Newword starts at the end of the document.)

Second, it causes Newword to continue the operation automatically--you won't have to keep entering ^l. In fact, the only way to stop a **g** operation is by entering ^u.

n This works only during replace operations.

Normally, Newword "asks permission" before it makes each replacement. But if you select **n**, Newword makes the replacements automatically.

r Rest. This does half of what the **g** option does. Newword goes from one instance of the word to the next, automatically. But, instead of starting at the beginning or end of the document, it starts right at the cursor.

The only way to stop it is by entering ^u.

u Ignore case. With this option, you can find a word regardless of whether it has upper or lower case , i.e., capital or small, letters in it. Ordinarily, if you search for "help," you won't find "Help" or "HELP." But if you select the **u** option, you'll find them all.

w Whole word. Use this to prevent Newword from finding words that are part of longer words. For example, during a search for "help," use it to keep Newword from stopping at "helpful."

There are 3 different ways to select the options ...

If you press **RETURN once**, you can select as many options as you need.

If you press **RETURN twice**, Newword will select options for you. In **nw**, it selects **w** (whole word) and **u** (ignore case). These are called the "default options."

If you press **ESC**, you bypass the options all together.

6.11 A LITTLE ADVICE

Though it's exciting, replacing words can be tricky. At first, you'll probably make mistakes, but that's normal. Even after you have more experience, you'll still make mistakes. That's normal, too. The really important thing to remember is that you should test your commands as much as you can.

While you're a novice, it's best to **always** use the **^qf** command to look through your document after you've made a replacement with the **n**, **g** and **r** options. Even if the replacement word is the same length as the original one, if the original had been hyphenated because it fell at the end of a line, you'll have to align the paragraph again, now that you've replaced the hyphenated word with a non-hyphenated one.

Once you've found a word, you can do any typing and editing you want (unless, of course, you selected the **g** or **r** options). When you finish, enter **^l** and Newword will take you to the next occurrence of the word.

In fact, Newword remembers the last find, or find and replace, command even after you save a document. Newword remembers it **until** you exit to the operating system. If you don't exit, you can open another document and use **^l** to start the same search sequence again.

>>> Now save QUICK.DOC, using **^kd**.

6.12 REVIEW

Quick Menu

* Quick scroll commands.
* Quick erase.

Find Commands

* Finding a word.
* Continuing the search with ^l.
* Finding and replacing words.
* Using options to narrow the search.
* Aligning paragraphs after replacing words.
* Default options.

7. PRINTING

7.1 PREVIEW

In this chapter, we will:

DISCUSS THESE TOPICS and INTRODUCE THESE COMMANDS

7.2 GETTING READY TO PRINT

>>> We're going to show you how to print documents. So first hook up your printer according to the instructions in the hardware manuals.

>>> **Use fan-fold, pin-feed computer paper!** If you don't have that kind of paper, you **won't** be able to do the following exercises conveniently.

>>> Reset the computer by pushing the **reset** button. This restores the logged drive to **A.** Then load **nw** in drive **A,** as you have before.

At the Opening Menu, read the help message about printing; press ...

>>>
j
p
ESC

After reading the message, you'll be back at the Opening Menu. Make sure PRACTICE.DOC is listed in the directory. If it isn't, you're not using the right disk. The document is on the Newword working disk.

7.3 PRINTING PRACTICE.DOC

In the Opening Menu, find the command for printing ...

P print a document

Then type ...

>>> **p**

Newword's displaying the print prompt, ending with this question ...

Document to be printed?

There's a "WHILE ENTERING" message, too.

Let's print PRACTICE.DOC. Notice that, after you type the document's name, you press the **ESCAPE** key, **not** the **RETURN** key. You'll find out why, later on ...

>>> **practice.doc**
ESC

Did you get a **printout**, that is, a printed version, of PRACTICE.DOC?

If not, review your hardware manuals. Did you connect the computer and printer properly? Is the power on? Are all the printer's switches set correctly? Is the paper inserted the right way?

When you get everything straightened out, try to print PRACTICE.DOC again.

If you still do not get a printout, refer to the **Newword Installation Manual**.

Notice that, after a printout, Newword returns to the Opening Menu.

7.4 PRINT BUFFERS

While Newword was printing PRACTICE.DOC, the screen displayed a message listing these commands ...

^U printing ends in midstream

^P printing pauses in midstream

^C printing continues after pause

These are always available to you while Newword is printing.

The first command, ^U, stops the printing for good. You'd use this if you realized that you were printing the wrong document, or if something went wrong during the printout.

The second command, ^P, also stops the printing, but only temporarily. That's why it's called "pausing." You might use this to proofread part of a printout. Use the third command, ^C, to resume printing after a pause.

Let's practice these commands. Before you do that, though, we ought to tell you about **print buffers.**

When you use these commands to stop a printout, your printer may keep going a short while before it stops.

Don't worry about it. It's normal. The computer sends characters to the printer much more quickly than the printer can use them. So there's a waiting area, called a "buffer," where the characters "hang out" until the printer's ready for them. Sometimes the buffer's in the printer, sometimes it's in the computer. Sometimes there's a buffer in both.

The important point is this: because of buffering, after you've entered one of these stop print commands, the printer will keep on printing. Exactly how long it continues printing depends on how many characters are waiting in the buffer(s).

So don't worry when Newword doesn't stop printing immediately, okay?

(There's a way to get around the buffer(s), though. Just turn off the printer's power switch **and** reset the computer. This will stop the printout immediately and empty the buffers. However, you'll have to load **nw** again.)

Okay. Let's check your buffer(s).

Here's how you'll do it. At the Opening Menu, you'll press **p**. Then, when Newword asks for it, you'll type PRACTICE.DOC and then press the **ESCAPE** key. As soon as the title, **INTERVIEWS WITH ARTISANS,** is printed, press **^p**. Newword will probably flash a new message, "**Pausing.**" The printing, though, will go on for a short while.

Ready?

Go ahead. Here's the sequence; remember, don't use a carriage return ...

>>> **p**
practice.doc
ESC

(Look for **INTERVIEWS WITH ARTISANS.**)

^p

How much of PRACTICE.DOC was printed? In other words, how big is your buffer?

>>> Print the rest of the document, by pressing ^c.

7.5 PRINT CONTROLS MENU

Next, we'll show you some print commands that you put in the documents themselves. So open PRINT.DOC, a new document ...

>>> **d**
print.doc<cr>
y

First, read these help messages about printing ...

>>> **^j**
^p
b
ESC

```
>>>     ^j
        ^p
        s
        ESC

>>>     ^j
        ^p
        c
        ESC
```

Now call up the Print Controls Menu by pressing ...

```
>>>     ^p
```

(Yes, it does get confusing, when the same command has several meanings. A moment ago, ^p meant, "Printing pauses in midstream." Then it meant, "Read the help message about printing." Now it means, "Call up the Print Controls Menu." The problem stems from there being more commands in Newword than there are letters in the alphabet.)

Anyway, look at the new menu. It lists several commands, but at this time, we'll look only at ...

B **bold**

S **underline**

C **pause**

We'll look at the others later.

7.6 WHAT YOU SEE . . .

In section 1.1, we said that one of Newword's distinguishing features is **what you see is what you get.** Well, so far that's been true. But, with printing, it's no longer the case. What you see isn't what you print, not exactly.

Let's go right to a demonstration.

Enter this sequence ...

>>> ^pb
bold
^pb

Newword will display what you entered like this ...

^Bbold^B

You see, most terminals can't show special print on the screen. They show "print control characters," instead. That's what the 2 **^B**'s are called. There's a print control character for every print command.

You can see the print control characters onscreen, but they're **not** printed.

However, a few terminals can show special print. If yours can, not only will the 2 print control characters by displayed, but "bold" will be highlighted, as well.

7.7 BOLD PRINT & UNDERLINING

Erase the line you just typed (use **^y**), and then enter this ...

>>> ^pb
This is bold type.
^pb

The text onscreen should look like this ...

^BThis is bold type.^B

If it doesn't, erase the line (**^y**) and try again.

On terminals that display special print, the entire line will be highlighted, too.

>>> Close PRINT.DOC with **^kd**.

Then print it, following the same sequence of entries that you did with PRACTICE.DOC ...

>>> **p**
print.doc
ESC

Regardless of your terminal's capabilities, the printout should be a line of text in bold type, with **no** print control characters showing. It should look like this ...

This is bold type.

If it doesn't, go back to section 7.5, **WHAT YOU SEE . . .,** and try again.

Now, let's try underlining.

>>> Open PRINT.DOC again.

>>> Move the cursor **past** the end of "**^BThis is bold type.^B,**" and enter 2 <cr>.

Then enter this ...

>>> **^ps**
This is underlining.
^ps

The line should look like this ...

^SThis is underlining.^S

If your terminal displays special print, the line will also be underlined.

>>> Correct your entry, if you need to; save PRINT.DOC, and print it.

How did you do? The printout should look like this ...

This is bold type.

This is underlining.

>>> Open PRINT.DOC again.

>>> Move the cursor **past** the end of the **second** line. Enter 2 <cr>.

Then enter this paragraph. The print commands are shown in bold print--enter the commands, don't type them ...

>>> **You can combine start-stop print commands. For example, ^psunderlining^ps and ^pbbold^pb can be used ^pb^pstogether.^pb^ps**

With print control characters, your paragraph should look like this on the screen ...

> **You can combine start-stop print commands. For example, ^Sunderlining^S and ^Bbold^B can be used ^B^Stogether.^B^S**

If your terminal displays special print, "underlining" will also be underlined, "bold" will be **highlighted,** and "together" will be both underlined **and** highlighted.

Print control characters can make the screen display pretty confusing. They take up space on the screen, but Newword knows they're not really part of the text.

>>> To see how this works, use ^s or ^d to move the cursor to a print control character--Newword will put it on the caret, not on the letter. What column number is showing in the cursor indicator at the top of the screen? Note it, then move the cursor to the right using ^d once. Look at the cursor indicator again--the column number hasn't changed even though the cursor moved over two characters, because Newword knows the print control characters aren't part of your text.

You can clear print control characters from the screen by entering ...

>>> **^od**

This takes print control characters out of the display, so you can see your text better. They have **not** been erased, though, so your printout will be okay. In fact, you can even enter print control characters now. But it's not a good idea, because, of course, you can't see them.

The same command turns the display back on; enter ...

>>> **^od**

>>> Now close PRINT.DOC and print it.

The complete printout should look like this ...

This is bold type.

<u>This is underlining.</u>

You can combine print commands. For example, <u>underlining</u> and **bold can be used <u>together.</u>**

One final word--use **p**rint control characters **i**n **p**airs.

Newword keeps printing in the special way you've selected until you instruct it to stop. If you forget to enter the second print control character, your printout won't be right.

So proofread carefully. Then, just to be on the safe side, make a test printout of your document, before you print the final copy.

If you're eager to try out the other print control commands, they're described in chapter 10.14.

7.8 PAUSING

>>> Open PRINT.DOC one last time.

>>> Call up the Print Controls Menu again, by entering **^p**. Find the command ...

C pause

^pc inserts a pause in the printout, wherever you want one. If your printer uses interchangeable type wheels, use this command to stop the printout before switching the wheels.

So, now you know 3 ways to make the printing stop--press **^u** or **^p during the printout,** or put **^pc in the document.** But only **^pc** lets you stop the printout exactly where you want it.

>>> If you want to practice this, put a **^pc** in PRINT.DOC--on a blank line or anywhere in the text. Then print the document.

When the printing pauses, Newword will display this reminder ...

Pausing at top of page. ^C to continue.

Printing documents is complicated, because you have another machine to keep track of, in addition to the computer and your document. Maybe you need to pause, along with PRINT.DOC.

Enter ^c, to finish the printing, and then take a break. Continue when you're refreshed.

7.9 DOT COMMANDS & MESSAGES

There's one more set of commands that control printing. They're called **dot commands** because they begin with a dot, that is, a period, instead of the **CONTROL** key.

There are quite a few dot commands, but we'll show you only 3 now ...

.op Omit page number
.pa Page break
.po Page offset

You should be at the Opening Menu. Make a copy of SAMPLE2.DOC; call the copy SAM.DOC. Be sure there's room for it on the disk, though. You'll need at least 12k. (Turn to section 6.1 to refresh your memory about checking disk space and making room on a disk.)

Here's the sequence for making the copy on the same disk ...

>>> **o**
sample2.doc<cr>
sam.doc<cr>

Now print SAM.DOC, using this sequence ...

>>> **p**
sam.doc
ESC

Notice these features of the printout ...

Page numbers are centered at the bottom of the page.

The left margins are the same throughout the document.

Like the other printouts you've made, the left margin is about an inch from the edge of the paper.

Open SAM.DOC now ...

>>> **d**
sam.doc<cr>

>>> Put the cursor in **column one** of the very first line. Check the cursor indicator at the top of the screen--the number after "C." should be "01."

Enter ...

>>> **.op<cr>**

Be sure the period is in column one. Put the cursor on the period, then check the cursor indicator.

>>> Move the cursor down the screen, to the **fourth paragraph** in the body of the letter. It begins ...

Mr. Sanders, the leading businessperson there, ...

In the blank space made by the paragraph indent, put the cursor in **column one** and enter ...

>>> **.pa<cr>**
.po15<cr> (That's ".POfifteen")

Be sure the periods are in column one on both lines.

>>> Save SAM.DOC and print it.

Compare this printout with the first one you made. All the features mentioned above have changed ...

There are **no** page numbers.

The page break is in a **different** place.

The left margin on the second page is offset **more** than it was on the first page. But all the lines with paragraph indents are still indented.

What happened? Well, **.op** turned off the page numbers, **.pa** moved the page break, and **.po15** offset the second page by another inch.

Newword normally offsets pages on its own, by **8** spaces. That's why the left margins in all your printouts have been about an inch from the edge of the paper. But when you specified an offset of 15, you **increased** the offset that was already there, by 7 spaces (15 = 8 + 7).

Notice, too, that the dot commands themselves are **not** printed. Newword **never** prints dot commands, if you've entered them correctly. Newword doesn't print lines that begin with a period in column one.

>>> Open SAM.DOC one more time.

On one of the blank lines above Samwise Gamgee's name, type this. Be sure to start in **column one** ...

..Where's Frodo?

>>> Now erase the 3 dot commands: **.op**, **.pa** and **.po15**.

>>> Save and print SAM.DOC again.

It should look exactly like the first printout. **If it doesn't,** open it, and check to be sure you followed our instructions accurately.

But where's, "..Where's Frodo?"

Well, remember, Newword **never** prints lines that begin with a period in column one, the way "..Where's Frodo?" does.

"..Where's Frodo?" is a **dot message.** Use dot messages as reminders to yourself, or to other people who share your documents. The reminders will show onscreen, but they won't be printed.

For example, when you're creating a document with lots of format changes, you can describe the changes right in the document, with dot messages. You can refer to

the descriptions whenever you need to, using Newword's find command.

Dot messages are also useful as place markers. If your document has long passages, mark them with brief dot messages, so they'll be easy to find, later ...

```
..£1
..£2
..£3
```

Two final comments ...

Dot messages should begin with 2 periods, not one.

We showed you that, for purposes of the cursor indicator, Newword doesn't count print control characters. Well, it doesn't count the lines that dot commands and messages are on, either.

7.10 PRINT OPTIONS

You should be at the Opening Menu; enter ...

```
>>>     p
        sample1.doc<cr>
```

When you printed documents before, you ended the command sequence with ESC. This is the first time you've used a carriage return.

As with finding and replacing, a carriage return at this point causes Newword to display **options.** The first one is onscreen now ...

```
Number of copies?
```

Just as you can bypass the find options, you can bypass the print options, too. Just press the ESCAPE key, as you've been doing throughout this chapter. Once you've selected the option you want, you can bypass the rest.

Let's go over the options now, starting with "Number of copies?".

To make **one** copy, just press the **RETURN** key. But, to make more than one, type the number you want. This time, just press ...

>>> **RETURN**

Newword will display the next option right away ...

Pause between pages?

When you're printing on individual sheets of paper, such as stationery, you have to put them in the printer one at a time. This option lets you stop printing at the end of **every** page, so you can put in a fresh sheet.

Of course, putting ^pc in the document would have the same effect. However, if you sometimes printed on continuous, fan-fold paper, you wouldn't want the pauses to be in the document itself. And if you later edited the document, adding just one line to it would move the page breaks. You'd have to move the ^pc's, then, too, making extra work for yourself.

If you enter **y** here, the printing will pause at the end of each page. But, this time, just press **RETURN**, which means, "Don't pause."

The next option displayed is ...

Starting page?

A printout can start on any page. You can type a page number here, but if you want to start on page **one**, just enter a carriage return. Let's do that--enter a carriage return.

Next is displayed ...

Ending page?

You can also end a printout on any page. The page you specify will be the last page printed. However, since SAMPLE1.DOC is so short, let's print it all--enter another carriage return. This causes Newword to print to the end of the document.

The last option is ...

Name of printer?

Notice that Newword is now displaying a list of printers, too.

You probably won't use this option very often. When you installed Newword, you may have installed a printer, too. If you enter a carriage return here, Newword will assume you're using that printer.

You have to select a printer here **only** if you're **not** using the one you specified in the install program.

So, press **RETURN** here if you're using the installed printer. If you're not, select your printer before pressing **RETURN.**

Now Newword will print SAMPLE1.DOC.

7.11 STOPPING A PRINTOUT

Altogether, there are **6** ways to stop a printout. Here's a summary ...

IN THE DOCUMENT

^pc	Causes the printout to **pause** exactly where you put ^pc. Resume printing with ^c.

DURING THE PRINTOUT

^U	Stops the printout **completely,** but because of the print buffer, the exact spot is impossible to predict.
^P	Causes the printout to **pause.** Placement is inexact, because of the buffer. Resume printing with ^c.
Ending page?	Stops the printout **completely,** at the end of the page you specify in response to the prompt following entering the "p" command.

Pause between pages	Causes the printout to pause at the end of every page. Resume printing with ^c. This is also a printing command prompt.
Turn the power off	Be sure to reset the computer, too.

7.12 REVIEW

Printing PRACTICE.DOC	* The print commands in the Opening Menu: ^U, ^P and ^C. * Print buffers.
Print Controls Menu	* How Newword shows start-stop print commands on computer terminals. * Start-stop print commands: bold and underlining. * Always use print control characters in pairs. * Pausing the printout with ^pc.
Dot commands	* Changing SAM.DOC's format. * Newword's built-in page offset. * Non-printing messages and place markers.
Print options	* Starting and stopping the printout. * Table of ways to stop a printout.

PRINTING

8. WRAPPING UP BEFORE GOING ON

8.1 PREVIEW

In this chapter, we will:

DISCUSS THESE TOPICS and INTRODUCE THESE COMMANDS

FLAG CHARACTERS
JUSTIFYING . . . . . ^OJ

8.2 CONGRATULATIONS!

You're no longer a novice!

You're an **experienced** novice!

If you want to repeat some of the **Do It Yourself,** by all means, go ahead. Remember, we said you should move at your own pace.

The commands we've covered so far are all that you need for most documents.

And don't forget to use the help messages and job aids.

Once you're confident with **the basics,** however, there are 2 things you can try--the other features of Newword and the **Newword Encyclopedia.**

Encyclopedia This is one of the other manuals that come with Newword. It explains how every part of the program

works. **All** the commands are in it, with easy-to-read technical explanations, and tips on how to do word processing efficiently.

The **Encyclopedia** is a reference book, not a training manual, so it has a different format from the **Do It Yourself.** There are no chapters. Instead, the material is arranged in short sections that are listed alphabetically. It really is an encyclopedia.

The preceding notwithstanding, **if you don't want to, don't!** Don't use any feature of Newword until you feel ready. The basics are entirely adequate for most word processing work.

8.3 FLAG CHARACTERS

Before leaving you on your own, we want to tell you about 3 things. First, flag characters.

You read about flag characters in the Edit Menu's help messages. You don't type them--Newword does, at the extreme right of the screen. They're useful as reminders, especially when you're editing. Here's a complete list (Some flag characters have to do with dot commands and messages, which are explained fully in chapter 11.) ...

< You ended this line by entering a carriage return.

A blank space in the flag characters column means you did **not** use a carriage return on this line. Instead, Newword ended it with word wrap.

+ This is a long line that extends past the right edge of the screen.

^ The document ends **above** this line.

P This is a page break; the next line of text will begin on a new page. Page breaks are also shown by a line of hyphens extending all the way across the screen.

WRAPPING UP

? The line begins with a period in column one, but it's not a dot command or dot message. Still, the line won't be printed.

The question mark appears as soon as you type a period in column one. If you then type a dot message or command, the question mark is replaced by one of the next 3 characters in this list.

. The line begins with a dot command that changes the format of both the printout and the onscreen display.

: The dot command on this line changes the format of the printout only.

1 The dot command on this line works best when it's used at the beginning of the document.

- This is an overprint line--the next line will be printed over this one. (Overprinting is an advanced print function.)

J This line ends with a line-feed, instead of a carriage return. This is very rarely used.

8.4 JUSTIFYING

"Justifying" is an old printing term. In Newword, it means making the lines at the right margin as even as they are at the left. When you type on a typewriter, if 2 or more lines end exactly at the right margin, it's an accident. Typewritten text usually has a **ragged right margin.** Here in the **Do It Yourself,** the lines are justified, and, so far, so are the lines in your documents.

In order to make the lines even at the right margin, Newword has to insert extra spaces in them. It puts the spaces between words, adding only as many spaces as are needed, so that every line ends exactly at the right margin.

The only exceptions are lines on which you've entered a carriage return. Newword won't justify those lines.

(Aligning and justifying are independent of each other. Aligning re-arranges paragraphs. For instance, if you've erased a word or 2 from a paragraph, you'll have

to align it. If justifying is on, the paragraph will be justified; if justifying is off, it won't be.)

>>> At the Opening Menu, open SAM.DOC, or make another copy of SAMPLE2.DOC and open the new copy.

Then, at the Edit Menu, read the help message about justifying; enter ...

>>> **^j**
^o
j

(Read the message.)

ESC

Once you turn justifying off (^oj), everything you type will not be justified, until you turn it on (^oj again). Normally, you'll justify a document as you type it. However, it's easy to unjustify a document that was typed originally with justified lines and vice versa.

For the practice, unjustify SAM.DOC. First turn justifying off, then align the paragraphs with **^b**. Remember, the paragraph indent was at column 6. Remember, Newword won't change lines flagged with <, because you used a carriage return on those lines.

When you're done, close SAM.DOC with **^kd** and print it. Then exit from Newword by typing **x** at the Opening Menu.

8.5 HOW TO NAME YOUR DOCUMENTS

In Newword, file names have to follow rules appropriate to the operating system. For most operating systems, that means ...

File names can have **1 to 12 characters.**

After the **8th** character, there **must** be a period. The operating system will probably put one there, even if you don't.

The period counts as the **9th** character.

WRAPPING UP

After the period, there can be 3 more characters, but no more.

This is basically a title-and-subtitle format. But, in computer jargon, we use the term "file name" for the part of the name that comes before the period, and "file type" for the part that comes after.

There are some common file types. For example, in the **Do It Yourself,** you used ".DOC" for every document you created. Newword's program files have file types of ".OVR" or ".COM." (The files aren't shown in Newword's directory, because we don't want you to fool with them. But they are listed in the operating system directory.)

When we wrote the **Do It Yourself,** we used the file type to indicate which draft we were writing. .1DR meant "first draft," .2DR meant "second draft," and so on.

Here are a few common file types ...

.LTR	Letter
.ENV	Envelope
.PRO	Proposal
.CTR	Contract
.BLR	Boilerplate
.NON	Non-document
.BAS	Basic

Finally, you can use these 3 characters in Newword file names and types: /, - and £. However, word processing accessory programs, like spelling checkers and index-ers, sometimes reject names with one or another of these characters. If that happens to you, just rename the document without the character(s), and try again.

8.6 GOOD LUCK!

We're sure you'll enjoy Newword.

8.7 REVIEW OF NW

Introduction

* Word processing.
* Roadmap to the **Do It Yourself.**
* Get to know your hardware.
* Definitions of terms and keys.

Loading Newword

* Studying your computer and CP/M manuals.
* Newword and CP/M.
* 4 kinds of disks.
* Formatting a disk, with **format.**
* Putting CP/M on a disk, with **sysgen.**
* Copying files, with **pip.**
* Making a backup copy of the Newword distribution disk.
* Boot disks.
* Working disks.
* Making a **Newword** working disk.
* Installing **Newword,** with **ns.**
* Loading **Newword.**
* .BAK files.

Opening

* Opening Menu.
* FIRST.DOC.
* Opening, typing and closing a document.

Editing

* Edit Menu.
* FIRST.DOC, or a copy of it.
* Correcting a document ...
  - Moving the cursor.
  - Erasing and inserting characters.
  - Scrolling the text.

Formatting

* Onscreen Format Menu.
* Changing the logged drive.
* SECOND.DOC.
* Typing a letter with tab stops.
* Blocking and Saving Menu.
* THIRD.DOC.
* Changing margins and line spacing, centering titles, and handling long lines.

Finding

* Quick Menu.
* Quick scrolling and quick erasing.
* QUICK.DOC, a copy of PRACTICE.DOC.
* Finding and replacing words.

Printing

* Print commands in the Opening Menu.
* Print buffers.
* Print Controls Menu.

WRAPPING UP

* PRINT.DOC.
* Bold print and underlining.
* SAM.DOC, a copy of SAMPLE2.DOC.
* Dot commands and messages.
* Print options in the Opening Menu.

Wrapping Up

* Congratulations! You're experienced!
* Flag characters.
* SAM.DOC.
* Justifying.
* File names and types.

WRAPPING UP

9. ADVANCED NEWWORD

9.1 PREVIEW

In this chapter, we will:

DISCUSS THESE TOPICS and INTRODUCE THESE COMMANDS

OPENING MENU

Working with non-documents	N
Merge-printing	M
Renaming documents	E
Protecting documents	C
Running other programs	R
Turning off the directory display	F

EDIT MENU

Insert on/off	^V
Paragraph line	^N
Help levels	^JJ

QUICK MENU

Quick cursor diamond	^QR, ^QE, ^QS, ^QD ^QX, ^QC, ^QDEL
Quick aligning	^QB

ONSCREEN FORMAT MENU

Temporary indenting	^OG
Hyphen help	^OH

BLOCKING & SAVING MENU

Blocking	^KB, ^KK, ^KH, ^KC ^KV, ^KY, ^KW
Document functions	^KO, ^KJ, ^KL, ^KR

PRINT CONTROLS MENU

Changing pitch	^PA, ^PN
Special print functions.	^PD, ^PX, ^PV, ^PT ^PY, ^PH, ^PRET, ^PO ^PF
Custom print functions	^PQ, ^PW, ^PE, ^PR

9.2 GOOD FOR YOU!

You've advanced!

As you've probably already noticed, Newword has more functions and commands than we've discussed so far. These are the "icing on the cake" that make Newword an extremely versatile and flexiole word processing program. Section 9.1 showed you the list of the functions we will discuss in this part.

You may have noticed how long the list is. Well, it is long, but you really will learn it fairly quickly, because **you already know the basics.**

The **Do It Yourself** will be a little different from now on. Mostly, we'll just describe functions and commands, rather than teach you how to use them. So don't forget to use the help messages and job aids. The job aids may be more important than they have been, because of all the new functions and commands you're going to learn.

9.3 OPENING MENU

```
                    O P E N I N G   M E N U

D get a document to change,                L change logged disk drive
  or create a new document                 E rename a document
N create or change non-document            O copy a document
P print a document                         Y delete a document
M merge-print a document                   F directory off
C protect a document                       R run a program
X all done with Newword (exit)             J help
```

>>> Load Newword.

We have already discussed some of these functions, and you've used them. Here, we'll describe the ones you have not used, in order. First, though, you should read the help messages. Type j and each new function's letter.

In our list here, we'll only add a few comments to what you read in the help messages.

N create or change non-document

(Read the help message.)

For the most part, you'll create non-documents in conjunction with other software, or when you write your own programs.

Menus for non-documents are modified versions of the document menus.

M merge-print a document

You'll learn about this in chapter 11.

C protect a document

(Read the help message.)

A protected document is one you can't change--you can't add or erase a single character. But you do have access to it.

If you know CP/M's **read only** status, you'll understand the protect function right away. Newword does more than protect the document, though.

Just for the practice, protect SAMPLE1.DOC now, by entering ...

>>>
```
c
sample1.doc<cr>
y
```

>>> Then open the document in the usual way.

Notice that the Edit Menu is renamed **PROTECTED MENU**, and that there are fewer commands. You can't erase anything, for instance.

Some of Newword's other menus have "PROTECTED" versions, too, and there's no Print Controls Menu, at all.

However, close SAMPLE1.DOC, instead of looking at the other menus; you'll have to use **^kq**.

L change logged disk drive

(Read the help message.)

In the earlier chapters, we didn't talk about **user numbers**, unless your particular computer system called for it. So we'll explain them now.

If you have lots of files on a disk, the directory gets pretty crowded, and looking for a particular file can be frustrating.

However, you can reduce the clutter by giving your files user numbers. Then, the directory will display files of only one user number at a time.

For example, let's say you have 60 files on a disk. And let's say that you gave the first 20 files, user number 1, the second 20 files, user number 2, and the last 20 files, user number 3.

The directory won't show all 60 files together. Instead, when you log on as user 1, it will show only the 20 files with number 1. When you log on as user 2, it will show only the number 2 files. And so on.

User numbers are really a function of the operating system. Newword "borrows" them. In Newword, you can only assign a number to a **new** file, and you **cannot** switch a file from one number to another. Both documents and non-documents can have user numbers. There are 32 numbers, starting with 0 (zero), not one. (Using "0" as the first number in a series is common in the computer field.)

When you assign numbers, do it in a systematic way. For example, user numbers are commonly used when more than one person has access to the same disks. In this situation, you might assign user number **0** to files that everyone shares, and assign the rest of the numbers to individual operators.

Another common use is with computer systems that have hard disks. The disks can hold hundreds of files, so user numbers are a lifesaver. Assign them to different categories of files, e.g., letters, reports, proposals and contracts, or sales, customer support, training, and service. Develop your own categories, ones that are appropriate for your office routines.

To assign a user number, enter the logged drive command at the Opening Menu (L). Then, you **must** type both the disk drive's letter (A, B, etc.) and the user number, like this ...

```
L   (This is an "L.")
2B<cr>
```

In this example, all the files you open will be on drive B, and they'll be assigned user number **2**. The directory will display only the files with this number, and those are the only ones you'll have access to.

You don't have to assign user numbers, if you don't want to. Newword and the operating system will automatically give all your files user number **0**, but you'll never have to bother with it.

E rename a document

(Read the help message.)

Rename a document by typing its present name first, then its new one.

If the document's on another drive, just include the drive's letter when you type the present name. For instance, if you're on drive A and want to rename SECOND.DOC, which is on drive B, enter this sequence ...

```
e
b:second.doc<cr>
new name<cr>
```

You can only rename documents that have the current user number.

F directory off

(Read the help message.)

When a menu displays an **on/off** command like this one, selecting it will do what the menu says. In this case, typing **f** will turn the directory display **off**.

R run a program

(Read the help message.)

Many programs can be "run" without having to exit from Newword. That is, you won't have to exit to the operating system to use them. This is handy for checking disk space with the operating system's **status** command, and for things like spelling and style checkers.

However, if you have CP/M, you can**not** run its built-in commands from Newword. You can **only** run the transient commands, that is, the ones in their own .COM files, like STAT.COM and PIP.COM.

J help

(Read the help message.)

In your work this far, the help level was fixed at **3**. This caused the menus to be onscreen all the time; prompts were fully displayed, too.

However, you can tailor the onscreen display of menus, status and ruler lines, and prompts.

You'll try out the different help levels later.

9.4 EDIT MENU

E D I T M E N U

CURSOR	SCROLL	ERASE	OTHER	EXTENSIONS
^E up	^W up	^G char	^J help	^O on-screen format
^X down	^Z down	^T word	^I tab	^K saving & blocks
^S left	^R up screen	^Y line	^V insert off	^P print controls
^D right	^C down screen	DEL left	^B align parag	^Q quick functions
^A word left		^U unerase	^N paragraph line	
^F word right			^L find/replace again	

>>> Open QUICK.DOC (or another copy of PRACTICE.DOC).

We've covered most of the Edit Menu already, except for 3 commands in the **OTHER** column. Read the help messages, if you want to, before going on.

OTHER

^v insert off

In earlier chapters, when you typed new characters, they were **inserted** into the text. Now, you can turn the insert function on and off at will. ^v is the command.

With insert **off**, new characters are typed right over old ones.

When insert's **off**, the word "Insert" is no longer displayed in the status line.

^n paragraph line

When you type **non**-documents, you **must** end every line with a carriage return, because there's no word wrap function. ^n lets you convert a non-document to a document, by clearing the carriage returns.

It works with documents that have been typed without word wrap, too.

HELP LEVELS

But now you can change the help level whenever you want to. Enter ...

>>> ^j^j
2<cr>

There's **no** menu at the top of the screen, now, only the status line and ruler line. You'll probably come to prefer this full-screen display.

Don't worry, though. At level 2, you can call up a menu any time you want to--hold down the **CONTRO** key and type the menu's letter.

To call up the Edit Menu, enter **^j**; the other menu letters are the ones you already know: ^k, ^o, ^p, ^q. And to clear a menu without having to enter a command, simply press the **SPACE BAR**.

The other help levels have minimal displays. At level **1**, you can call up the menus only by entering ^j. However, Newword will display all its prompts during operations like finding, along with an expanded **WHILE ENTERING** message.

At help level **0**, only the ruler line and prompts are displayed.

Regardless of the help level, you can **always** get the help messages, by entering ^j. So, even though you're at help level 2 now, you can continue reading help messages.

Now we'll review the other menus, starting with the Quick Menu.

9.5 QUICK MENU

Q U I C K M E N U

CURSOR
E up to upper left
X down to lower right
S left end of line
D right end of line

SCROLLING
R top of document
C bottom of document

ERASING
Y line to the right
DEL line to the left

FINDING
F find
A find and replace
P find a page

ALIGNING
B align paragraphs

QUICK CURSOR

You'll recall that the keys for moving the cursor are arranged in a diamond. Well, the quick cursor and quick scroll commands are laid out in a diamond, too ...

^qr

^qe

^qs ^qd

^qx

TOP OF DOCUMENT

UP TO UPPER LEFT

LEFT END OF LINE RIGHT END OF LINE

DOWN TO LOWER RIGHT

BOTTOM OF DOCUMENT

You already know ^qr and ^qc. They move the cursor to the beginning and end of the document. The new commands move the cursor, too, but only in the text that's currently onscreen.

QUICK ERASE

^qDEL complements ^qy, the quick erase command. ("**DEL**" stands for the **DELETE** key.)

Both commands erase from the cursor to the end of the line. **^qy** erases to the **right** end of the line, and **^qDEL** erases to the **left** end.

FINDING

^qp is a command for finding a page in your document. You can find any page, by typing its number.

Newword will put the cursor at the beginning or end of the page, depending on whether it's moving forwards or backwards though the file.

QUICK ALIGNING

So far, to align a document, you entered the align command at each paragraph. Well, now you can align an entire document with just one command, **^qb**. Hyphen help works with this command, too.

Newword will pause briefly at the end of every paragraph, to give you the opportunity to stop the aligning--just enter ^u. But if you don't stop it, Newword will go on to the next paragraph.

9.6 ONSCREEN FORMAT MENU

O N - S C R E E N F O R M A T

MARGINS	TYPING	DISPLAY
L set left	W word wrap off	D print controls off
R set right	J right justify off	H hyphen help on
X release	E soft hyphen	
T ruler off	G temporary indent	TABS
F ruler from text	S set line spacing	I set tab stop
O ruler to text	C center line at cursor	N clear tab stop

This menu has several new functions and commands, so we'll spend more time on it.

MARGINS L----!----!----!----!----!----!----!-------R

^ot ruler off

You can turn the display of the ruler line on and off with ^ot. This gives you one more line for display of text.

^of ruler from text

So far, you used margin and tab stop commands to change the ruler line. Now, you can **type** a new ruler line--right in your document--and replace the current ruler with the typed one. Just put the cursor anywhere on the typed ruler and enter ^of.

After you've moved the typed ruler to the menu, erase it, or make it a dot message. You'll learn about dot messages in chapter 10.

Entering ^of while the cursor is in a line of regular text (rather than a typed ruler line) will change the left and right ruler margins to match those of the text. Tab stops within the new margins are unchanged.

^oo ruler to text

^oo does the opposite of ^of. It causes Newword to insert a copy of the current ruler in your document.

TYPING

^ow word wrap off

When you typed FIRST.DOC, you learned about word wrap. Now you can turn it on and off, to suit you. Remember, entering the command will do what the command description in the menu says. In the example shown here, entering ^ow will turn word wrap **off.**

When word wrap is off, you have to end each line yourself, by entering a carriage return. This is just like using a normal typewriter. In fact, on some terminals, Newword will sound a beep when the cursor reaches the right margin.

Because of the carriage returns, you won't be able to align the document very well, unless you first clear the returns with ^n.

^oe soft hyphen

^oe makes what are called "soft hyphens." You'll find out about them soon.

^og temporary indent

^og is a way of temporarily changing the left margin. It works during both typing and aligning. We'll show you how to use it in a moment.

DISPLAY

^oh hyphen help off

^oh turns hyphen help on and off. Hyphenating is pretty complicated, so we'll go over it with you carefully.

9.7 INDENTING LINES AUTOMATICALLY

Sometimes you'll want a few paragraphs to be indented more deeply than the others. Here's a quick way to do it.

>>> Move the cursor to a blank line at the end of QUICK.DOC.

>>> Then enter **^og**.

Notice that there's a "V" in the ruler line, now, at the first tab stop.

>>> Enter another **^og**.

Now the "V" has moved to the second tab stop.

From now on, every line you type will be indented to the second tab stop, until you press **RETURN**, enter an align command (**^b** or **^qb**), or change the left margin.

>>> Type a few lines, to see how temporary indenting works. Be sure to end your paragraph with a carriage return.

9.8 HYPHEN HELP

Think about aligning. So far, when a word extended beyond the right margin, it was moved to the next line. Well, if you want to, you can hyphenate the word, leaving part of it on the current line, and moving only the second part to the next line. This has the advantage of consolidating the text, especially when you also justify the lines.

The Newword function that does this is hyphen help. It's turned on and off with **^oh**. But it works **only** when you align paragraphs.

Furthermore, hyphen help uses a special kind of hyphen, called a "soft hyphen." A soft hyphen is one that's printed **only** when it's at the **end** of a line of text.

Where else would a soft hyphen be? Well, suppose you align a document and hyphenate a few words. All the soft hyphens will be at the ends of lines. But if you edit the document and then align it again, some of those hyphenated words will be moved. They'll no longer be at the ends of lines.

When soft hyphens are moved away from the ends of lines, Newword changes them to equals signs (=), so you can recognize them. Because they're soft, they won't be printed.

Most of the time, you'll want to leave hyphen help **on.** Even when it's on, the hyphens you type by pressing the hyphen key, as in "happy-go-lucky," will be printed. (You guessed it; they're called "hard hyphens.")

Once in a while, though, you may want to type a soft hyphen yourself. There's a command for that--enter **^oe,** and Newword will type a soft hyphen for you.

9.9 HYPHENATING QUICK.DOC BY ALIGNING IT

Before we start, make sure hyphen help is turned **on.**

>>> Call up the On-screen Format Menu, if it isn't onscreen. Look under **TYPING.** The command description for **^oh** should read "**hyphen help off.**" If it does, press the **SPACE BAR.** If it reads "**hyphen help on,**" type **h.**

Make sure justifying is turned **on,** too.

>>> The command description should read, "**justify off.**" If it does, press the **SPACE BAR;** if it doesn't, type **j.**

(Aligning and justifying don't have to be used together. We're combining them here for convenience.)

>>> Now set your left margin at **1** and your right margin at **65.**

Finally, move the cursor to column one, on the first line of the first paragraph, like this ...

>>> "[A] friend asked me . . .

Then enter ...

>>> **^b**

When Newword finds a word that's too long for the line it's on, it displays the hyphen help prompt, which explains that you have 3 choices ...

> Type a hyphen immediately. This will cause Newword to divide the word, using a soft hyphen, and to move the second half of the word to the next line. The letter the cursor's on will begin the second half of the word; **or**
>
> Move the cursor, and then type a hyphen. You can move the cursor only within the word it's in. Do this when the original cursor position isn't appropriate; **or**
>
> Don't hyphenate at all. Just enter **^b**. This causes Newword to move the entire word to the next line.

After you make your choice, Newword will execute it and then continue aligning.

Right now, unless you've altered QUICK.DOC, Newword wants to hyphenate "writer." Notice that the cursor's on the comma immediately after the word. Newword doesn't suggest a way of hyphenating the word--it just puts the cursor at the last possible column space in the line.

>>> Move the cursor to the "t" in "writer" (wri**[t]**er), and then type a hyphen (-).

>>> Now Newword's stopped at "subtle." Move the cursor to the "t" (sub**[t]**le), then type a hyphen.

>>> Newword's stopped once again, at "yourself." Move the cursor to the "s" (your**[s]**elf), then type a hyphen.

That finishes the paragraph.

Here are the rest of our hyphenations in QUICK.DOC. You may want to hyphenate differently. That's fine, so long as you follow accepted usage ...

2nd paragraph

bookkeep-
ing

ampli-
fies

3rd paragraph

NONE

4th paragraph

hun-
dred

5th paragraph

immeasur-
ably

remind (Didn't hyphenate it.)

designs (Didn't hyphenate it.)

You could have started the aligning with ^qb, instead of ^b, but everything would have happened too quickly for us to tell you about it.

You can also combine temporary indenting (^og) with aligning. It's easy. Enter as many indenting commands as you need, followed by the appropriate aligning command (^b or ^qb). Newword will do the rest.

You can stop hyphen help and aligning any time by entering ^u.

9.10 SOFT HYPHENS

All the hyphens Newword just typed were soft hyphens--they'll be printed only if they stay at the ends of lines.

Let's change the margins of one of the paragraphs, align it, and then see what happens to the soft hyphens.

>>> Change the right margin to 50.

>>> Now choose a paragraph you hyphenated a word in. Move the cursor to column one on the paragraph's first line, and enter **^b**.

What's happened to the soft hyphens in the paragraph? They've become equals signs (=). They're soft, so they won't be printed, because they're not at the ends of lines.

>>> Change the right margin back to **65**, and align the paragraph again.

The soft hyphens are back at the ends of lines, and they're displayed as hyphens. If you were to print QUICK.DOC now, they'd be printed.

9.11 BLOCKING & SAVING MENU

B L O C K I N G & S A V I N G M E N U

SAVING	BLOCKING	DOCUMENT
S save & resume edit	B mark start of block	O copy
D save document	K mark end of block	J delete
X save & exit Newword	H hide/show markers	L change logged drive
Q quit without saving	C copy block	R insert a document from disk
	V move block	
	Y delete block	
	W store block on disk	

9.12 BLOCKING

Do you know what's meant by the term "cut-and-paste"? If you've revised typewritten documents, you've done your share of cut-and-paste editing. It's when you **cut** 2 sentences from page 13, and 3 paragraphs from page 15, and **paste** them all on page 21. It's a nuisance.

Well, say goodbye to cutting-and-pasting!

Newword presents ...

***** THE BLOCK *****

A block is a piece of text. It can be practically any size, even an entire document. To make a block, all you have to do is put the right markers in the right

places. Then everything between the markers becomes a block.

It's as simple as that. And it's as terrific as it is simple.

>>> Let's get started. Move the cursor to QUICK.DOC's title, "INTERVIEWS WITH ARTISANS." Put it on the first "I" in "INTERVIEWS" (**[I]**NTERVIEWS).

To mark the **beginning** of the block, enter ...

>>> **^kb**

In the document, the marker <B> is now in front of "INTERVIEWS." The marker always means, "A block begins here." Also, the title has moved over to make room for the marker. Don't worry about its taking up space on the line, though. Newword knows the marker isn't part of the text itself.

Now move the cursor to the end of the title, after the last word, and enter ...

>>> **^kk**

The end of the block is now marked with <K>. On some terminals, the whole block is now highlighted, too.

You can do all sorts of things, now, with "INTERVIEWS WITH ARTISANS." We'll show you a couple, and explain the others.

9.13 MOVING & HIDING A BLOCK

>>> Make a blank line below the last line of QUICK.DOC, and put the cursor in column one there.

Enter ...

>>> **^kv**

Presto!

Notice 3 things ...

The block is down at the end of the document now. If you were to look at the beginning of the document, you'd find the title's been erased.

The block starts in column one, where the cursor was when you entered ^kv.

The block is still marked as a block, that is, the <B> and <K> markers are still displayed.

It's very important that you "hide" the block markers immediately. Don't worry about where the cursor is--it doesn't have to be in the block. Just enter ...

>>> ^kh

Hiding the markers doesn't erase the text or the markers. For example, if you were to enter another ^kh now, the markers would reappear.

But you should usually hide the markers right away. There's a command for erasing blocks, but it only erases them if their markers are displayed. By hiding the markers, you protect the block from being erased accidentally.

9.14 OTHER BLOCK COMMANDS

Here are some comments on the rest of the block commands ...

^kc copy a block

This makes a copy of the block, in the same document the block's in.

Copying blocks is different from moving them. When you copy, the original block stays where it is, instead of being erased. Of course, the copy goes wherever you put it.

Be sure to move the cursor to where you want the copy to be, **before** you enter ^kc.

^ky erase a block

This erases both markers **and** text. Like hiding, it works even if the cursor's nowhere near the block.

^kw store block on disk

You can store a block on the disk, as if it were another document. In fact, once you store it, it is a document.

Why store blocks? Well, to make the block into a separate document, the way SECOND.DOC became SAMPLE2.DOC. Or to use the block as a boilerplate in other documents.

You can also move a block from one document to another, by first storing the block.

9.15 DOCUMENT COMMANDS

Most of these commands are just like ones in the Opening Menu.

^ko copy

With this command, you can make a copy of any document, **except** the one you're currently working on. (The only way to copy your current document is to mark the whole thing as a block; then use **^kw** to store it on the disk.)

The copy will be stored on whatever drive you name.

^kj delete

This command erases any document (or non-document), on any drive, **except** you can't erase the document you're currently working in.

^kl change logged drive

Changing the logged drive with ^kl has the same effect as changing it at the Opening Menu.

You can change user numbers, too, but **not** for the current document.

^kr **insert a document from disk**

You can also bring a copy of another document into the document you're currently working on.

Newword will **insert** the new document into the text, at the cursor, so be sure to position the cursor **before** entering the command.

To move a block from your current document into another one, first store the block on the disk, using ^kw. Then open the other document and insert the block, using ^kr.

Like all copy commands, this one leaves the original document unchanged.

9.16 PRINT CONTROLS MENU

P R I N T C O N T R O L S

START-STOP	CHANGES	SPECIAL	CUSTOM
B bold	A alternate pitch	H overprint char	Q custom 1
D doublestrike	N standard pitch	RET overprint line	W custom 2
S underline	C pause	O binding space	E custom 3
X strikeout	Y change color	F phantom space	R custom 4
V subscript			
T superscript			

START-STOP

^pb **bold**

This is bold print.

^pd doublestrike

This is doublestrike type. Test it on your printer, to see if it's significantly different from bold print.

^ps underline

This is underlining. Blank spaces won't be underlined, unless you type underlines yourself.

^px strikeout

~~This is strikeout print--hyphens are printed over the text. It's for legal documents.~~

^pv/^pt subscript/superscript

This is super script and this is super script.
sub
They're for mathematical and chemical formulas. However, some printers can't reproduce them properly, so test yours.

CHANGES

"Pitch" means how many characters are printed in an inch of text.

Many printers let you select 2 or 3 different pitches. **Pica** is a common one--it has **10** characters to the inch. **Elite** is another--it has **12** characters per inch.

Using the install program, you designate one of these 2 pitches as your standard pitch, and the other, as your alternate pitch. Then, every time you load Newword, the standard pitch is selected automatically.

Therefore, to print in standard pitch, you needn't enter any pitch commands on your own. To print in alternate pitch, though, you have to use a pitch command.

(You can alter the pitch designations with a special installation of Newword.)

When you change pitch on dot-matrix printers, the size of the characters themselves changes, too,

automatically. But on printers with removable typing elements, only the pitch changes. The characters themselves stay the same, unless you put on a new typing element.

To change the typing element in the middle of a document, enter ^pc right after the change pitch command. This will stop the printer exactly where you want it.

^pa alternate pitch

This changes pitch from your standard one to whatever has been installed as the alternate.

^pn standard pitch

This switches back to standard pitch.

^py change color

Some printers use 2-color ribbons. This command switches from one color to the other. To go back to the original color, enter **^py** again.

On dot-matrix printers, some people use this command to select doublewide pitch, instead of ribbon color. But doing so requires a special installation of Newword. Check both your printer manual and Newword's install manual.

SPECIAL

^ph overprint char(acter)

^ph overprints **one character** with any other character you choose, like these accent marks in **é**cran ("screen," in French) and b**ł**ąd ("bug," in Polish).

Here's how to type "écran" ...

Type	Display	Printout
e^ph'cran	e^H'cran	écran

^pRET overprint line

^pRET overprints **an entire line,** with any character you choose, like this ...

This/is/overprinted/with/a/slash

To use the command, type the first line. At the end of the line, enter **^p** and then press **RETURN.** Then type the line you want to be printed over the first one. End it with a carriage return, but no ^p, to be sure it stays the same length during later alignings.

^po binding space

^po binds 2 characters (or words) together with a single blank space that Newword can't change when it justifies and aligns the document. Here in the **Do It Yourself,** we've used **^po** to bind the ellipsis to the preceding word. If we hadn't bound them, Newword might have put the ellipsis on a new line, like so
...

^pf phantom space

The typing elements of wheel printers sometimes have a character that's not on the computer keyboard. It's often the symbol for "cents." **^pf** prints that character.

Otherwise, entering **^pf** will cause the printer to insert a blank space in the line.

CUSTOM

Some printers have extra features not covered by the common print control commands. If you have a printer like that, use the commands here to activate the features. The install manual tells you how to do it.

If you don't install these commands, don't use them. Like **^pf,** they'll cause the printer to insert blank spaces in your text.

ADVANCED NEWWORD

10. DOT COMMANDS

In chapter 6, you learned how to use 4 dot commands ...

```
.op     Omit page number
.pa     Page break
.po     Page offset
..      Dot message
```

In this chapter, we'll describe most of the rest of the dot commands and explain how to use them. The few remaining ones are explained in chapter 12.

Dot commands are for changing the format of a **printout,** controlling margins, placement of page numbers, etc. Most of them don't affect the onscreen display at all. And, although they can be seen onscreen, the commands themselves are **not** printed.

Instead of formatting the printout, a few dot commands affect the way the printer itself operates. And, if you use word processing accessory programs (spelling checkers, indexers, etc.), you'll find they sometimes use dot commands, too.

10.1 HOW TO TYPE DOT COMMANDS

Dot commands **must** begin with a period in column **one,** even if your left margin is further to the right. After typing the period, type the rest of the command and its related characters. Then end the line with a carriage return.

(Ordinarily, when your left margin is to the right of column one, e.g., at column 6 or 21, you'd have to release the margin, with ^ox, before you could type something in column one. But with dot commands, you don't have to use the margin release.)

Don't put dot commands in the middle of a paragraph, unless there's no other choice. During subsequent alignings, the commands will be drawn into the text of the paragraph.

If you enter several commands at once, put them in the sequence that you want Newword to follow during the printout.

When you type a period in column one, Newword displays a question mark in the flag characters column. This is replaced by a colon, period or one, **if** you type a real dot command. If you mistakenly type something else, the question mark stays. But the line still won't be printed.

Flag characters are explained in section 8.3.

Here in the **Do It Yourself,** dot commands are typed in lower case, and there are no spaces between the commands themselves and their related characters, e.g., **.po12.** You can use the same style, but it's okay, too, if you type in upper case, or if you put **one** space between the commands and the related characters. All of these variations work ...

.po12	.po 12
.Po12	.Po 12
.pO12	.pO 12
.PO12	.PO 12

Sometimes, we use "**N**" to represent additional, related characters that you have to add to the commands. The characters may be the number of lines in a margin, the text of a header, etc.

Numbers are often used in the commands; type whole numbers (4, 16, 66, etc.), **not** fractions (9/48, 5 1/2, etc.) or decimals (58.6, 12.5, etc.).

Many of the commands have **default values,** i.e., settings that are automatically chosen every time you load Newword. When you enter a dot command for which there's a default setting, the dot command overrides the default. You can change the default values themselves with the install program.

At the end of the chapter are **examples** of dot commands as they're used in documents. There's also a **diagram** showing relationships between page format and some of the commands.

10.2 FORMATTING THE PRINTOUT ONLY

Most dot commands affect **only** the printout--they have **no** effect on the onscreen display, other than appearing there when you type them.

Newword flags these commands with a colon (**:**), except for one command, which it flags with a one (**1**).

Here are the commands ...

.bp	Bidirectional print
.cw	Character width
.fo	Footer
.fm	Footer margin
.he	Header
.hm	Header margin
.pc	Print default page numbers in column N
.pg	Restore default page numbers
.pn	Page N
.po	Page offset
.op	Omit default page numbers
.uj	Microjustification

.bpoff/.bpon Bidirectional print

This forces printers to print while the head is travelling in just **one** direction, usually left-to-right. It sometimes improves the quality of the printing, but it doesn't work on every printer.

.bp0 (That's a zero.) also turns bidirectional print **off**, and **.bp1** (That's a one.) also turns it **on**.

When you load Newword, bidirectional print is turned **on**.

(See example **3**.)

.cwN Character width

Use this command when you change pitch on a printer that's capable of **microspacing** (also called "microincrementing"). It tells the printer how much space to allow for a character.

The width of a printed character is measured in 120th's of an inch. Pica type has a character width of 12/120, and elite, 10/120.

To change character width, type the command and then just the first part of the fraction (**.cw12**). Do **not** type the entire fraction (~~.cw12/120~~).

Newword's as-delivered default character width is **12,** for pica type.

Character width has a direct effect on the number of characters you can fit in a line. The wider the characters, the fewer you can fit in a line of any given length.

(See example **5.**)

.foNNNNNN Footers

Footers are lines repeated at the **bottom** of **every** page. Here in the **Do It Yourself,** we printed chapter titles as footers. Footers can have any combination of words, numbers and symbols.

You can print as many as **3** footers ...

.fo or **.f1**	First footer
.f2	Second
.f3	Third

The numbers in "**.f1**," "**.f2**" and "**.f3**" are part of the commands, so you must type them as they're shown, without extra spaces. But you can put a space **after** the numbers, before you type the footer's text.

To turn a footer **off,** enter the same dot command that began it (that is "**.f1**," "**.f2**" or "**.f3**"), and enter a carriage return immediately after. To separate 2 footer lines by a blank line, you **must** use **.f1**NNNNNN for the first footer, **.f2**<cr> for the blank line, and **.f3**NNNNNN for the second footer.

If you put no spaces between the footer command itself and the text, or one space, in the printout the footer will begin at the left margin. If you put additional spaces, i.e., more than one, between the command and the text, the footer will be indented by the same number of spaces. The rule for this is that Newword ignores the first space, but implements all others. For example, if you put 5 spaces between command and text, the footer will be indented 4 spaces from the left margin.

Normally, Newword prints the page number as a footer. This is called the "default page number." Once you enter your own footer, the default page number is turned **off.** Therefore, to print the numbers, you'll have to include them in one of your own footers, or headers.

To do this, type "£" in the footer, at the column position where you want the numbers to be printed. It marks the position of the numbers' **first** digit.

If you want the "£" symbol itself to be printed in the footer, you have to precede it with another symbol, \.

To make Newword print "£" in a footer (or header), type **\£.** To print "\," type **\\.**

Footers can have different print features (**^pa, ^pb,** etc.) from those in the document's text. Just enter the appropriate commands from the Print Controls Menu.

(See example **2.**)

.fmN Footer margin

The footer margin is the number of blank lines Newword leaves between the **last line of text** and the **first line of the footer;** the as-delivered default is **2** lines.

The footer margin **plus** all the footers must be **less than** the bottom margin (**.mb**). If they're greater than the bottom margin, Newword will print all the footers, anyway, making a mess of your printout.

(See example **4.**)

.heNNNNNN Headers

Headers are lines repeated at the **top** of every page.

Everything that applies to footers also applies to headers, **except** that Newword normally uses **no** header.

(See examples 2 and 4.)

.hmN Header margin

The header margin is the blank lines between the **last header line** and the **first line of text;** it's preset at time of delivery to 2 lines, like the footer margin.

Newword will print all the headers you create, even if the top margin is too small.

(See examples 2 and 4.)

.fo^pk/.he^pk Alternate corners

Page numbers, titles and section names are often printed in the **outside corners** of book and magazine pages. On even-numbered pages, they're in the left corner; on odd-numbered pages, in the right corner.

This command lets you do the same thing, through the use of a unique print command, ^pk, that works **only** when it's part of a footer or header command.

On **odd**-numbered pages (1, 3, 5, etc.), ^pk has **no** effect. A footer that looks like this onscreen ...

```
.fo^K                                   Annual Report
```

... will look like this in the **printout** ...

```
                                        Annual Report
```

But on **even**-numbered pages (2, 4, 6, etc.), Newword ignores the blank spaces after ^pk. So the same footer will print like this ...

```
Annual Report
```

For page numbers, you **must** use "£" to indicate the numbers' position. It will mark the numbers' **first** digit.

(See example 2.)

.pcN Print default page numbers in column N

Newword usually **centers** the default page numbers on their footer line. However, they're centered within the default ruler line, even if you're not using that ruler. So, when you change the margins, you may have to move the page numbers' print column, so the numbers are centered within the new ruler. This command moves them to the column specified by N.

You don't have to use **.pcN** only for centering. It moves the page numbers to any column you want.

.pcN works **only** with the default page numbers, not with ones you put in your own footers or headers. **£** positions these.

(See example 4.)

.pg Restore default page numbers

If you've turned off the default page number--with another footer or with the **omit default page number** command (**.op**)--you can start it again by entering **.pg.**

(See examples 2 and 4.)

.pnN Page N

Normally, Newword numbers pages from the beginning of the file. This command lets you reset page numbers when you need to--simply type the command and the new number.

.pnN also restores the printing of default page numbers, if they've been turned off with **.op.**

As a special precaution, this command is flagged with a one (1), instead of a colon. Newword reads dot commands whenever it encounters them in a file, whether it's moving forwards or backwards, i.e., from beginning to end, or from end to beginning.

When it reads **.pnN**, it changes the page numbers, as it's commanded to do. That's fine, if Newword is moving forwards.

But, sometimes, if it's moving backwards, the numbers it changes won't be the right ones. It depends on how many times you've used the command.

Newword can keep track 10 **.pn** commands at a time. If you use more than that, page numbers will be correct only when Newword moves forwards.

(See example 2.)

.poN Page offset

In a printout, text seldom begins at the left edge of the paper. Instead, all the lines are uniformly indented a few spaces. The indented space is called the "offset."

Offset is defined as the space between the **left edge of the printed page** and the screen's **column one.**

Newword's delivered with a default offset of **8** spaces. If the left margin on the screen is at column **one,** the printout's left margin will be 8 spaces. If you change the page offset to, say 18, the printout's left margin will be 18 spaces; but the onscreen left margin will remain at column **one.** Page offset determines where Newword prints text with respect to the left edge of the paper in the printer. Be careful. It is possible to make Newword print text off the right edge of the printout, if you increase the page offset without decreasing the **onscreen** right margin.
To print text at the left edge of the paper, enter **.po0.**

(See example **3.**)

.op Omit default page numbers

This tells Newword **not** to print the default page numbers, which are normally printed in a footer.

It does **not** turn off numbers printed by the other dot commands.

In the install program, you can select **.op** as a default.

(See example **3.**)

.ujoff/.ujon Microjustification

Sometimes, printers that do microspacing align columns of text improperly. If you have this problem, turn microjustification **off** by entering **.ujoff before** typing a column. But be sure to turn it back **on** after typing the column, by entering **.ujon**.

.uj0 (That's a zero.) also turns it **off**, and **.uj1** (That's a one.) also turns it back **on**.

Microjustification is always turned **on** when you load Newword, unless you change it with the install program.

(See example **5**.)

10.3 FORMATTING BOTH PRINTOUT & ONSCREEN DISPLAY

Dot commands that change the length of the page affect both the onscreen display **and** the printout. So does the automatic ruler command.

Newword flags most of these commands with a period (.). It flags 2 of them with a one (**1**).

Here are the commands ...

..	Dot message
.cp	Conditional page break
.lh	Line height
.mb	Bottom margin
.mt	Top margin
.pa	Page break
.pl	Page length
.rr	Automatic ruler line

..NNNNNN Dot message

Put notes and reminders in your files with this command. If a message has several lines, be sure you begin **every** line with 2 periods.

You can get away with using just one period, instead of 2, but it's not the accepted style. You can also use **.ig**.

(See examples **3** and **4**.)

.cpN Conditional page

With this command you can prevent Newword from putting a page break where you don't want one.

When it comes across a **.cpN**, Newword checks to see if the current page will end within N lines. If it will, Newword ends the page early, and the line right **after** the **.cpN** is made the first line of the next page.

(See example **5**.)

.lhN Line height

Some printers can change the height of each line **without** changing the height of the printed characters. This allows for more subtle changes in line spacing than ^os provides. Line height is measured in 48th's of an inch.

Newword's as-delivered line height is **8**, meaning 8/48 of an inch. This produces 6 lines per inch, which is fine for most typefaces. If you set line height to **10**, the space between the lines of text will be 2/48 of an inch greater (8 + 2 = 10).

This will change the number of lines that can fit on a page. The top and bottom margins will stay the same, so the difference will be seen in the text itself.

.lhN is flagged with a one (**1**), instead of a period. Newword keeps track of 10 line height changes at a time. If there are more than that in a single file, they'll be implemented correctly only when you move forwards through the file.

Some printers can't change line height.

(See example **1**.)

.mbN Bottom margin

This margin is the number of lines between the **end of the text** and the **bottom of the page.** Newword is preset at time of delivery for a bottom margin of **8** lines. The footers and the footer margin fit **within** the bottom margin.

To increase the number of printed lines on a page, **decrease** the bottom (or top) margin.

(See example **3**.)

.mtN Top margin

This is the number of lines between the **top of the page** and the **first line of text**; it's preset at time of delivery at **3**.

(See examples 2 and 3.)

.pa Page break

Use this command to make Newword insert a page break where it normally wouldn't, for example, at the end of chapters and major sections.

To keep tables, paragraphs, etc., on a single page, it's best to use **.cpN**, instead of **.pa**. Otherwise, you'll have to check page breaks every time you align your document.

(See example 2.)

.plN Page length

This sets the **total** number of lines on the page, **not** just the number of printed lines or text lines. Newword is preset at delivery for a default page of **66** lines. This is based on what's called the "vertical motion increment," or VMI.

The VMI is a standard distance the printer's paper carriage rolls for each new line. The distance is 8/48 of an inch, which works out to 6 lines per inch, and 66 lines on a page 11 inches long.

The only way you can change the VMI is with a special patch in the install program.

Headers, footers, top and bottom margins, and the text itself must fit **within** the page length. Newword will print everything, even if it doesn't fit.

Change the page length when you switch to paper that's longer or shorter than 11 inches, e.g., forms and envelopes. To calculate the new length, use this formula ...

.plN = 6 x length of paper in inches

When you change page length, Newword adjusts all page breaks to the new length.

.plN is flagged with a one (**1**) to warn you that, if there are more than 10 page length changes in a single file, they'll be implemented reliably only when you move forwards through the file.

(See example **1.**)

.rrNNNNNN **Automatic ruler line**

When it encounters this in a document, Newword changes the ruler line to the one you've typed (**NNNNNN**).

If you begin the new ruler with ...

.rr---- Left margin at column **one.**

.rrL---- Left margin at column **3.**

.rr L---- Left margin at the **L's** column position.

Of course, the right margin will be at the **R's** column position.

As delivered, Newword allows a maximum of **6** ruler lines in a file. You can change this with the install program.

An alternative way to enter automatic rulers is with the overprint line command, **^pRET.** Type the **.rr** on the first line, then type the ruler on the second, overprint, line. The lines will **not** be printed.

(See example **3.**)

10.4 A LAST WORD OF ADVICE

Interactions between dot commands can be tricky, so be sure to **test your printouts.** If something goes wrong, look at the commands onscreen. Proofread them; change their sequence; erase the troublesome command and figure out some other way to format the printout.

Normally, Newword won't print a line that has a period in column one, but there's a way to get around that. Use the overprint character command, **^ph**, like this ...

Keystrokes	Display	Printout
^ph.	^H.	.

Be sure to start in column one.

And, remember, there's no law that says you have to use lots of dot commands in every document.

10.5 EXAMPLES

EXAMPLE 1:

```
.pl84
.lh12
```

Let's suppose you're printing a booklet for vision-impaired readers, and that you're going to use doublewide pica typeface on paper that's 14 inches long. You're also going to increase line height, in order to increase the spacing between lines.

First, calculate the new page length. Six times 14 inches equals 84, so the new page length is **84**.

Next, through trial and error, select a line height that looks right. Make test printouts of several different heights, and choose the best. In this example, we liked **12**.

That's all there is to it.

EXAMPLE 2: .op

Letters of Gimli Gloin

Collected and Published by
The Workers of the Minas Tirith Archives

.pa
----------(Page break)----------P

Table of Contents

.pa
----------(Page break)----------P

.h1^K ^BMinas Tirith Archives^B
.pn1
.fo^K ^Y^B£^Y^B

[First page of the text]

----------(Page break)----------P

.h2<cr>
.h3^K ^A^BLetters of Gimli Gloin^N^B
.hm4
.mt7

These are the opening pages of a book. Neither the title page nor the table of contents page have page numbers, headers or footers. Page breaks were forced, with .pa.

Page numbering starts with the first page of the text (.pn1); note that it isn't necessary to use .pg. The numbers are printed in the footer, at £, and the alternate corners command (^pk) is used in the footer, too.

On the text's following pages, second and third header lines are added. .h2 is a blank line, and .h3 is "Letters of Gimli Gloin." To make room for the added header lines, both the header margin (.hm4) and top margin (.mt7) are reset from their default values.

Notice, too, the alternate corners (^K) and other print control characters in the headers and footer (^B, ^Y, ^A and ^N).

("Minas Tirith," "Gimli Gloin" and "The Quest for the Ring" are from **The Lord of the Rings** novels by

J.R.R. Tolkien, Ballantine Books, Inc., New York, 1966; copyright 1937, 1938, 1965, 1966 by J.R.R. Tolkien.)

EXAMPLE 3:

```
..THIS IS STATIONERY FORMAT:
..
.rr--!----!----!----!----!--------£----£---------R
..
..FIRST PAGE:
.bpoff
.mt10
.mb5
.po14
.op
```

This is the format for the first page of an organization's correspondence on business stationery.

In the ruler line, the left margin will be at column one (**.rr**). Bidirectional print is turned off, to insure that the printing is sharp and clear (**.bpoff**). The first page of the letter has a pre-printed letterhead, so the top margin is fairly large (**.mt10**), and there's no header. The bottom margin is only 5 lines (**.mb5**). Finally, the page offset is reset (**.po14**), and the default page number is turned off (**.op**).

EXAMPLE 4:

```
.. FOLLOWING PAGES:
.. FOR THE PRINTOUT, BE SURE TO SELECT THE "PAUSE AT
.. PAGE BREAKS" PRINT OPTION.
..
.h1 [Date]
.h2 [Addressee]
.hm4
.mt6
.pg
.pc1
.fm3
```

This is the format for STATIONERY's following pages. Letterhead paper isn't used, so there's room for headers, one for the date and one for the addressee. Notice that the headers (**.h1** and **.h2**) **plus** the header margin (**.hm4**) **equal** the top margin (**.mt6**) (2 + 4 = 6).

Default page numbers are restored by **.pg**, and positioned in column one by **.pc1**.

Instead of using the default footer margin of 2 lines, a 3-line margin is set (.fm3). Here, too, it's important to be sure the bottom margin (5 lines) has room for the footer (one line) and footer margin (3 lines). Five is more than one plus 3, so this is fine.

Formatting commands and reminders like these can be stored on disk, as a document. Every time you type a letter, use ^kr to insert the document into the letter.

EXAMPLE 5:

```
.cp6
.ujoff
     ^C^A038F          42   55   47
     054D              02   01   14
     0555              02
     03D8              5B
     03DC              5D
     03AE              20^C^N
.ujon
```

These columns of numbers are from a document about programming.

.cp6 will keep the 6 lines of text from being broken up by a page break, and turning microjustification off (**.ujoff**) will keep the columns from being misaligned by the printer.

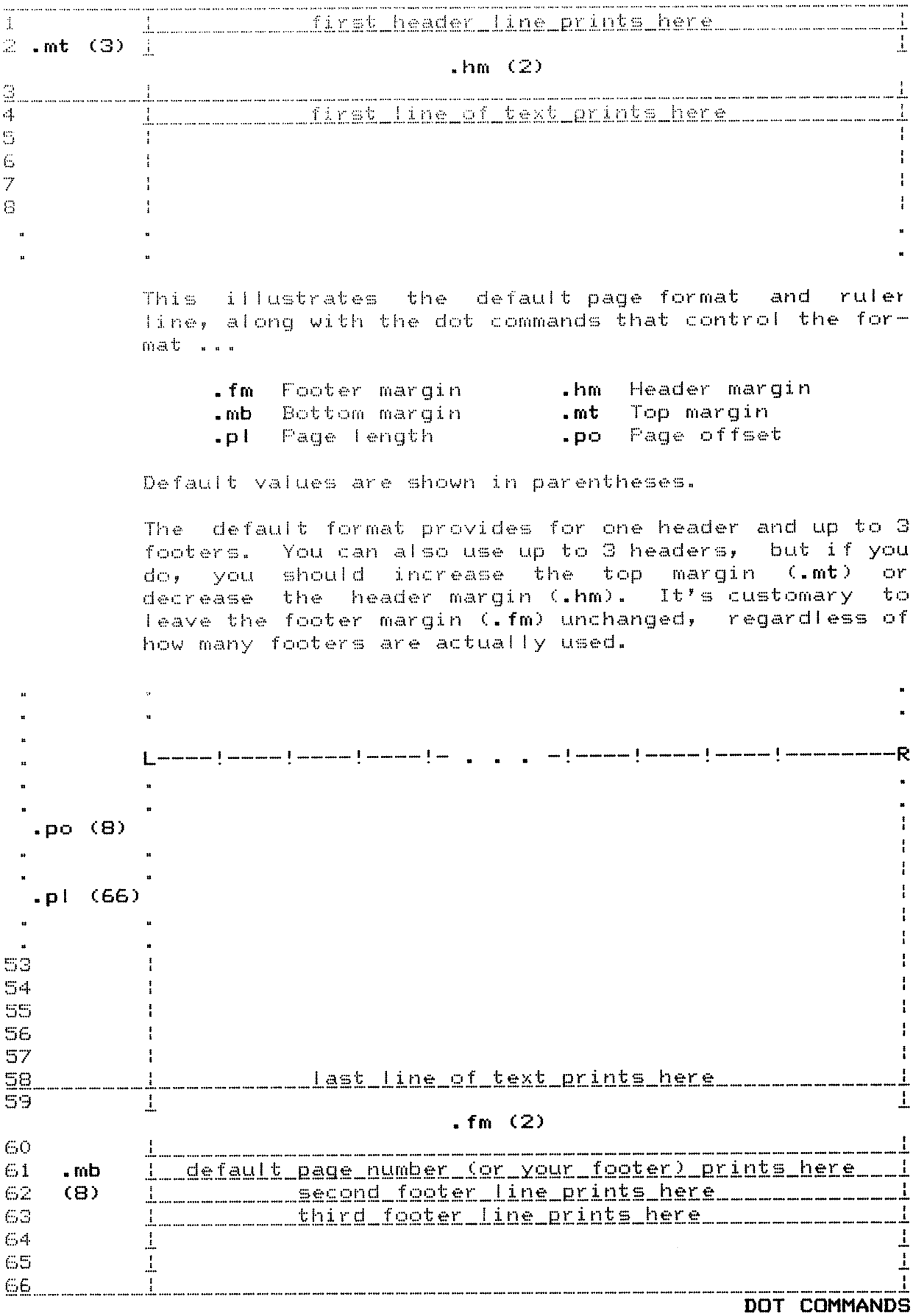
The printer will pause ("^C") just before shifting to the new typeface ("^A"), so its typing wheel can be changed.

After the list, microjustification and typeface ("^N") are restored to their previous values.

EXAMPLE 6:

On the next page is a diagram of Newword's customary default page format, i.e., the format Newword uses if you don't change it. It's for an 8 1/2 x 11-inch page.

The diagram also illustrates how some of the dot commands affect page format. It shows the position of headers and footers, margins, page offset and the default page number.



This illustrates the default page format and ruler line, along with the dot commands that control the format ...

.fm	Footer margin	**.hm**	Header margin
.mb	Bottom margin	**.mt**	Top margin
.pl	Page length	**.po**	Page offset

Default values are shown in parentheses.

The default format provides for one header and up to 3 footers. You can also use up to 3 headers, but if you do, you should increase the top margin (**.mt**) or decrease the header margin (**.hm**). It's customary to leave the footer margin (**.fm**) unchanged, regardless of how many footers are actually used.

DOT COMMANDS

11. MERGE-PRINTING, CHAINING AND NESTING

11.1 WHAT ARE THEY?

This chapter talks about some of the things that makes word processing so much more useful than ordinary typing. Merge-printing, chaining and nesting are all printing functions

Chaining and **nesting** are ways of combining files when they're being printed, rather than combining them beforehand, on the disk.

Merge-printing is a way of individualizing printouts of things like form letters and mailing lists. Newword can draw information from special data files you've created, and insert the information in the printout, wherever you want it.

By typing a few new dot commands, you can cause Newword to add one file to another, making a long chain of files. You can put file A in the middle of file B, and file C in the middle of A, and file D in the middle of C, and then tack on files E and F at the end. The dot commands can go in the files that are being chained together, or, sometimes, in a separate file of dot commands alone.

This kind of printing is called "chaining," when one file is tacked on the end of another, and "nesting," when one file is inserted in the middle of another. There are 2 major advantages to chaining and nesting ...

Documents that are too big to be in one file, whether they're 80,000 characters long or 800,000, can be prepared as a series of files, and then strung together like links in a chain during the printout.

Documents with boilerplates can be printed more conveniently. A "boilerplate" is text that's used, unchanged, in different documents. It may be a few paragraphs describing your organization, a list of products and prices, a standard part of a contract, or whatever. With a typewriter, you have to type boilerplates over and over again, or do cut-and-paste editing. With Newword, you just print them with dot commands to automatically insert the boilerplate.

Merge-printing is used with form letters, envelopes, mailing lists, etc. Those unsolicited letters you get in the mail, the ones that repeat your name 2 or 3 times, but always misspell it, are created by merging a form letter with a separate file containing the names and addresses of thousands of people. Now that you have Newword, you can misspell the names of thousands of innocent people, too.

In the Opening Menu, M is the command for merge-printing.

The different kinds of printing can be mix-and-matched, making documents that are chains of individualized, nested boilerplates! This is called "very advanced!"

11.2 PREPARATIONS

>>> For this chapter, be sure Newword and the 3 Newword documents, SAMPLE1.DOC, SAMPLE2.DOC and PRACTICE.DOC, are on the disk in drive **A**. You'll need an additional **24k** space, as well. Use the operating system's **status** command to check (for CP/M, the command is **stat**). If you don't have enough room on the disk, erase or move some of the files to make room.

>>> Load Newword.

>>, Open a new document, CHAIN1.DOC, on drive **A**.

Then, enter this ...

>>> ^kr
sample1.doc<cr>

11.3 CHAIN1.DOC

CHAIN1.DOC is now a copy of SAMPLE1.DOC.

>>> Move the cursor to the blank line between the first and second paragraphs, and enter a carriage return. This will add another blank line.

Type the following merge-print dot command. Be sure to start in column **one**, and end the line with yet another carriage return ...

>>> .fi sample1.doc<cr>

Now do the same thing at the very end of the document. Be sure there's one blank line between the existing text and the new dot command, like this ...

>>> . . . screen will change when you do.

.fi sample1.doc<cr>

>>> Did you put **.fi sample1.doc<cr>** in the document in 2 places? Good; save CHAIN1.DOC and print it.

>>> Look over the printout. Compare it with example 1 at the end of this chapter. You'll find that SAMPLE1.DOC has been printed 3 times, and that the second and third printings start right where you entered **.fi** commands.

The second printing, the one that starts right after the first paragraph, is **nested** within the first printing. (Documents inserted anywhere before the end of a file are nested.)

The third printing is an example of **chaining,** because it's inserted at the very end of the file.

11.4 CHAIN2.DOC

>>> Now let's get fancy. Open another new document, CHAIN2.DOC.

This file will be just a list of .fi commands. This is another way of chaining documents together. With it, you can merge-print documents without putting the .fi commands in the documents themselves.

As you type the file, notice these things ...

Newword flags all the dot commands with a **colon**, to indicate that they affect only the printout.

When you type **.pa**, a page break appears in the file. There'll be a break in the printout, too.

Last--and this is very important--, there **must** be a carriage return after every dot command.

Here's the command file, CHAIN2.DOC. **Don't** type the comments that are underlined, and **don't** leave any blank lines ...

>>>
```
.pfon<cr>
.rm55<cr>
.ojoff<cr>
.fi sample1.doc<cr>
.pa<cr>
.ojon<cr>
.ls2<cr>        That's ".LS2."
.fi sample1.doc<cr>
.pa<cr>
.ls1<cr>        That's ".LSone."
.rm65<cr>
.fi practice.doc<cr>
```

>>> Now save CHAIN2.DOC, and then print it, with the merge-print command. Compare your printout with example 2 at the end of the chapter.

What did all those commands do? ...

.pfon

This turns on a print function called "print formatting." It is equivalent to ^B used to align paragraphs while entering text with

Newword. **.pfon** will cause every paragraph to be aligned while printing.

.rm55

As you can see in the printout, this changes the right margin to **55,** just as if you had typed in the document originally with a right margin of 55.

.ojoff

This turns justifying **off.** (Paragraphs which are aligned during printing will have a ragged right margin.)
.fi sample1.doc

You know what this means, "Insert SAMPLE1.DOC here."

It was printed with the right margin at **55,** and the lines were justified.

.pa

As we said it would be, the page break is in the printout.

.ojon

Turns justifying **on.** (Paragraphs which are aligned during printing will have the right margin justified, even if they were originally entered with a ragged-right margin.

.ls2

This changed line spacing to **2.**

.fi sample1.doc

In this printing, the lines are double-spaced, but not justified.

.pa

Another page break.

.ls1

Line spacing changes back to one.
.rm65

The right margin is changed to **65.**

.fi practice.doc

With this command, PRACTICE.DOC is chained to the file.

It's single-spaced and justified, and the right margin's at **65.**

Creating a command file like this one is like writing a program--a very simple one, of course. But it teaches one important feature of computers. On this level of computing, they tend to be pretty linear. They do only one thing at a time, finishing the first before going on to the second. So should you.

Think first about how you how want the printout to look. Then review the various documents that are going into the printout. What's the correct sequence of documents? Are the documents all formatted alike? If not, what changes have to be made, in which documents? Remember to put the formatting commands **before** the document they're going to change.

To work correctly, Newword needs a carriage return at the end of every file and every dot command. Ruining a large printout is a frustrating way to discover you left one out somewhere. So make a habit of **always** ending your documents with a carriage return.

11.5 CHAINING FILES ON SEVERAL DISKS

Documents like the **Do It Yourself** are so big that they won't fit on one disk, let alone in just one file. It's easy to print them, however, if your computer has 2 drives.

(The following procedure may also work with one-drive computers having virtual drives. Check your hardware manuals. Hard-disk computers usually don't need this procedure, however.)

print

Let's say you have 3 documents, A, B and C, on one disk, and 3 more, D, E and F, on another. At the end of all but the last document, you'll type a .fi command, but the one that ends document C will be a little different. Here are the **.fi** commands for chaining these documents. Don't type them, though--this is just an explanation ...

MERGE-PRINTING

On first disk ...

End A with .fi B<cr>.
End B with .fi C<cr>.
End C with .fi D change<cr>.

On second disk ...

End D with .fi E<cr>.
End E with .fi F<cr>.

The "**change**" in **.fi D.DOC change<cr>** tells Newword to stop the printing so you can change disks. Newword will display appropriate prompts to guide you.

When you use this procedure, put the Newword working disk in drive **A. Don't** remove this disk.

Put the disk with the first 3 documents in the second drive, and start the printing. You'll only be able to change the disks in the second drive. You can't change disks in the drive that contains Newword.

11.6 CREATING A DATA FILE

Chaining and nesting documents is only part of the advanced-printing story. Merging documents and data files is the other part.

To teach you merging, we'll first show you how to create what's probably your first non-document, a data file listing just names and addresses. Next, you'll create a letter, a form letter that will use the names and addresses from the data file.

Go to the Opening Menu. Then open a **non**-document, by typing ...

>>> n

Since this is probably your first non-document, let's go over some of the differences between documents and non-documents.

>>> First, set the help level to **3**, if it's not there already, so you can see the Non-Document Edit Menu.

Notice the tab commands in the **OTHER** column. Then, look at the **EXTENSIONS** column. **^O** is missing, because there's no Onscreen Format Menu in non-documents. That means there are no ruler lines, margins, word wrap, justifying, etc.

In non-documents, you have to end every line yourself, with a carriage return, just as though you were using an ordinary typewriter.

The status line is different, too. It shows line and column, but not page. And the line number can go as high as 99,999.

The other menus are pretty much like the document versions (except that the Onscreen Format Menu is missing, as we said.)

>>> Reset the help level to whatever you prefer.

Let's create your data file, now. Follow these general instructions carefully ...

Start each line in column **one.**

Put in all the commas that we have, including the extra one in the third line.

Don't enter a carriage return until you see the symbol <cr>.

Okay, here's the file. Go ahead and type it ...

>>> **Fremore,Threngen,Farmore,Folde,Eastfold,Rohan<cr>**
Bombur,Heavyfoot,Commissary,The Mountain,Erebor,Dain<cr>
Sam,Gamgee,Mayor of the Shire,,Hobbiton,the Shire<cr>

Let's go over what you just did ...

First, some terminology. Each data item is separated by a comma or carriage return. For example, in the third line, or record, "Sam" is a data item; so is "Mayor of the Shire."

Did you type 3 lines, ending each one with a carriage return? Did you leave no spaces between data items?

Did you put the double comma (,,) in the third line? There's one less data item in this record than in the other 2. When this happens, you **must** put an extra comma in its place.

MERGE-PRINTING

>>> Okay. Proofread FORM1.DTA and then save it. You can print it, if you like, using the ordinary print command, but it isn't necessary.

11.7 CREATING A FORM LETTER

Now you'll type the letter that will use the names and addresses from the data file.

>>> Create a new document, FORM1.LTR.

>>> The letter we've typed on the next page was created with justifying on, and then aligned with hyphen help on, but you can format your version any way you like. **Don't** type the underlined comments, though ...

```
.op
.df form1.dta
.rv FNAME, LNAME, ADDRESS1, ADDRESS2, TOWN, COUNTRY
```

Oakenshield Hall
Minas Tirith, Ithilien
Rethe 11, 1464 S.R.

```
&FNAME& &LNAME&
&ADDRESS1&
&ADDRESS2/0&   That's a capital "oh," not a zero.
&TOWN&, &COUNTRY&
```

Dear Friend &FNAME&:

I've been asked to serve as agent for a small, rural cooperative producing specialty foodstuffs and crafts items. Currently, they're offering the most delicious honey I've ever tasted, packaged in a whimsical, bear-like ceramic pot.

You're sure to enjoy the sample I've asked them to send you. It should be arriving very soon.

Do write to me, &FNAME&, if you'd like to establish a trade in the honey, or if you're interested in their other wares. You won't be disappointed.

Yours very sincerely,
in friendship and trade,

Gimli Gloin, Elf-friend
Lord of the Glittering Caves
Agent in Trade

Finish the letter with .pa.

```
.pa
```

("Oakenshield Hall," "Minas Tirith, Ithilien," "Rethe," "1464 S.R.," "Sam Gamgee," "Hobbiton, the Shire," "Folde, Eastfold, Rohan," "Bombur," The Mountain, Erebor, Dain," and "Gimli Gloin, Elf-friend, Lord of the Glittering Caves" are from **The Lord of the Rings** novels by J.R.R. Tolkien, Ballantine Books, Inc., New York, 1966; copyright 1937, 1938, 1965, 1966 by J.R.R. Tolkien.)

Okay. What did you do in this letter? Well, you began with 3 dot commands ...

.op just turns off page numbering.

.df form1.dta is an instruction to Newword; it means, "Read the data file named 'form1.dta.'" (That's the one you typed a few minutes ago.)

.rv is another instruction; it means, "In the data file, read these variables: FNAME1, LNAME1, etc." In the form letter, the names of the variables **must** be enclosed in ampersands, i.e., the "and" symbol (&) whenever you want the data item to be inserted for a variable. Otherwise, Newword won't individualize the printout.

Variables are the names of storage spots for data items. With the .rv command, you're naming each data item: **FNAME**, **LNAME**, **ADDRESS1**, etc. By naming them, you enable Newword to keep track of where the data items go in the document.

Notice how we numbered the 2 **&ADDRESS&**'s. The variables' names must be different from each other. When they refer to similar categories, e.g., 2 lines of an address, numbers are an easy way to make the names different. But the numbers aren't necessary--we could just as easily have used different words. Later, you'll see another situation where numbers come in handy.

Then you typed the letter ...

You put the names of the variables where you wanted Newword to print information from FORM1.DTA, the data file.

Notice that you can use a variable more than once; **FNAME** is used 3 times.

When you merge-print this letter with the data file, Newword will put the right variable in the right place, every time. It will keep printing copies until it's used all the information in FORM1.DTA.

>>> Proofread FORM1.LTR, then save it and merge-print it, using the **M** command in the Opening Menu.

How did you do? Compare your printout with example 3 at the end of the chapter. You should have made 3 individualized copies of the form letter.

If something went wrong, proofread your two files care-

fully. When it comes to merging documents and data files, Newword is very fussy. You probably left out a comma, or something similarly hard to notice. (If you print FORM1.LTR with the ordinary print command, it will look exactly as it does on the previous page, and proofreading it will be easier.)

Notice that, in the letter to Sam Gamgee, there are only 3 lines in the addressee section, whereas there are 4 in the other letters.

The missing line corresponds to the missing data item you typed a double comma for (,,), in the data file. In the printout, Newword would have left a blank line, if you hadn't typed the "/O" in "&ADDRESS2/O." You might want to try removing the "/O" to see what happens.

11.8 MORE ABOUT MERGING

>>> Once you have a good printout of 3 letters, each one individualized appropriately, open FORM1.LTR again.

On the blank line **after** the .rv command, enter this ...

>>>
```
.dm This letter is to &FNAME& &LNAME&.
.av "Type the date, then press RETURN. ", DATE<cr>
```

Yes, put a blank space after the period.

Now erase the date in the letter itself ("Rethe 11, 1464 S.R."), and type this in its place ...

>>>
```
&DATE&
```

.dm means, "Display message." Before Newword prints each letter, it will display this message onscreen, using the appropriate data items from each line.

.av means, "Ask for variable." You can put variables in the document that aren't named in **.rv** commands. But you **must** provide the data some other way--otherwise, the printout will be ruined. **.av** is one way to do it.

The next time you merge-print FORM1.LTR, Newword will display the part of the **.av** command that's between quotes, before printing each letter. You should do what the message says, i.e., type today's date and then press RETURN.

Let's proof the command ...

The message must be enclosed in quotation marks, and there must be a comma **after** the second set of marks.

Note that the blank space after the period will show in the onscreen display during the merge-printing.

After the comma, you should type the name of the variable (**DATE**), so Newword knows where the information goes in the letter. Don't use ampersands (&).

>>> Okay. Save FORM1.LTR and print it. Each time Newword displays the .av message, type today's date and press **RETURN.**

How did you do? The dates in the letters should be today's. Is it?

Actually, you don't have to include a message with .av, if you don't want to. You can just type the variable's name, like this ...

.av DATE

During the merge-printing, Newword will display the name, followed by a question mark, like this ...

DATE?

However, because .av variables have to be entered for every document, they're not always convenient. So there's another way to define variables that aren't named in a .rv command. It's the .sv command.

.sv means, "Set variable." Instead of typing the date for each document, you can set the **DATE** variable once, and Newword will use it for every printing.

However, you have to enter the .sv in the document, **before** you merge-print. Here's an example--**don't** type it ...

.sv DATE, July 12, 1983

Notice there are no ampersands, just commas. And the variable's name begins the command, instead of ending it.

You can use .av and .sv more than once.

11.9 MAKING ENVELOPES FOR FORM LETTERS

After all the letters have been printed, you can use the same data file to print the envelopes.

>>> Create a file called FORM1.ENV.

```
.dm Put an envelope in the printer now.<cr>
^pc   Yes, enter the ^pc command again.
.pl25<cr>   That's ".PL25."
.mt12<cr>   That's ".MTtwelve."
.mb0<cr>
.po40<cr>
&FNAME& &LNAME&<cr>
&ADDRESS1&<cr>
&ADDRESS2/0<cr>   That's a capital "0."
&TOWN&, &COUNTRY&<cr>
.pa
```

These commands pause the printing to allow you to put in new envelopes, and also format the envelope printout. .pl25 changes page length--the standard 9 1/2 X 4-inch envelope is 25 lines long. .mt12 sets a top margin of 12, so the printing begins halfway down the envelope. .mb0 sets the bottom margin at zero. .po40 sets the page offset at 40, so the printing is well over to the right.

>>> Okay. Proofread FORM1.ENV, save it and print it. Newword will print envelopes, just like it printed the form letters, pausing between each one. You can use real envelopes, or ordinary paper. If you use paper, advance the paper carriage to the next page, after each envelope address has been printed.

Afterwards, compare your printout envelopes with those in example 4 at the end of the chapter.

11.10 PRINT FORMATTING

Print formatting is a little more complicated than it appears. Here's why.

When Newword is merging data files and form documents, every time it encounters a variable's name surrounded by ampersands (&) in a document, it inserts text from the data file. Now, the number of characters in the text and in the variable's name are usually different. Ordinarily, this would change the length of the line the variable's in.

But Newword takes care of that. Every time it replaces a variable with text from a data file, it aligns the paragraph. Normally, it uses the default margins, justification and line spacing.

Newword does this aligning **if** you don't put any **.pf** command in the form document, **or** if you put in **.pf dis.**

"**dis**" stands for "discretionary;" it means simply that the paragraph is to be aligned **only** if data is inserted for a variable.

However, if you put **.pf off** in the document, Newword won't do any aligning. Using **.pf on** means "align every paragraph."

In CHAIN2.DOC, you learned how to change a printout's margins, line spacing or justifying with dot commands--**.lm**, **.rm** and **.oj**. They have an effect only during aligning, though.

Suppose you're printing a long form document made up of a number of separate files. To be sure the printout will have uniform margins, you could use print formatting dot commands, along with **.pfon** ...

```
.pfon
.lm6
.rm55
```

Then suppose, for some reason, that you wanted one of the files to be printed with its original margins.

You'd have to use **.lm** and **.rm** to set the margins back to the originals, and **.pf off** to turn aligning back off.

Here's a table outlining the **.pf** command ...

No .pf The default is **.pf dis**. In other words, not using a **.pf** command is the same as using **.pf dis**.

.pfdis Aligns paragraphs only if data has been inserted into the text from a data file.

.pfon Always aligns paragraphs when printing.

.pfoff Turns off aligning.

11.11 MULTI-COLUMN MAILING LABELS

For large mailings, people don't usually print directly on envelopes. Instead, they use self-adhesive mailing labels.

The labels come in 2 styles. One style is a roll of labels. The roll is one-label wide. You could merge-print this kind of label now, using the commands and techniques we've shown you.

However, the second style is tougher. That's one in which there are 2, 3 or more columns of labels, usually on sheets, instead of rolls.

We're not going to lead you through this, step by step. We'll just explain how to do it.

To merge-print multi-column labels, you have to use the overprint a line command (^pRET), very long lines, and the TAB key.

Here's how you would set up a document for sheets with 3 columns of mailing labels. Don't type this--it's only an example ...

```
.op<cr>          (This turns off page numbering.)
.po2<cr>         (Choose an offset appropriate for
                  your own labels.)
.mt1<cr>         (Again, choose what's appropriate.)
.mb1<cr>         (Ditto.)
.rr---//-----!-----//-----!-----//-----!-----R<cr>
                 (Make an automatic ruler line [see
                   chapter 11]. Make the right margin
                  3 times the width of the paper.

                     Put tabs at the actual begin-
                     ning of the 2nd and 3rd
                     labels.)
```

```
.df NNNNNN<cr> (Specify the data file.)
.rv NAME1, STREET1, CITY1, STATEZIP1<cr>
.rv NAME2, STREET2, CITY2, STATEZIP2<cr>
.rv NAME3, STREET3, CITY3, STATEZIP3<cr>
                    (There are 3 .rv commands and 3
                     sets of variables, because there
                     are 3 columns of labels.
                         If your sheet had 2 columns,
                     you would use 2 .rv commands and 2
                     sets of variables.)
&NAME1&^pRET        (^pRET is the overprint command.
                     The cursor will move to the next
                     line after you enter it.)
TAB                 (Pressing the TAB key moves the
                     cursor to the second label.)
&NAME2&^pRET        (The cursor will move to the next
                     line again.)
TAB
TAB                 (Pressing the TAB key twice will
                     move the cursor to the third
                     label.)
&NAME3&<cr>         (Don't overprint--use normal car-
                     riage return.)
&STREET1&^pRET      (Use overprint here.)
TAB
&STREET2&^pRET      (Overprint here, too.)
TAB
TAB
&STREET3&<cr>       (Use normal carriage return.)

                    (And so on. After &STATEZIP3&,
                     start over with &NAME1&.)
```

On the screen, the variables' names won't be lined up as they are in the example above. Instead, they'll be spread across the screen, but staggered, like this (Notice that we also show the flag characters Newword will display.) ...

```
&NAME1&                                          -
                  &NAME2&                        -
                                    &NAME3&      <
&STREET1&                                        -
                  &STREET2&                      -
                                    &STREET3&    <
&CITY1&                                          -
                  &CITY2&                        -
                                    &CITY3&      <
&STATEZIP1&                                      -
                  &STATEZIP2&                    -
                                    &STATEZIP3&  <

(And so on.)
```

Type the mailing list as a data file, just like the one you did for FORM1.DTA. In the first line, type the first name, street, city, state and ZIP. And so on.

11.12 SUMMARY OF MERGE-PRINTING

Merge-printing can be hard to get the knack of. Here are some rules and tips that we hope will smooth the way for you ...

"Chaining" is adding a file at the end of another file. "Nesting" is adding a file anywhere before the end. "Merge-printing" is combining form documents with data files.

Keep in mind that, in merging documents with data files, everything has to be just right. Take your time, and proofread carefully.

Test your printouts. There's a way to do this without actually printing. Make a duplicate file of your document's merge-print commands, but don't put any text in the file. (Use ^k_ commands to store all the dot commands in the duplicate file.)

In the duplicate, add a **.dm** command exactly like every **.rv** command, except that the variables' names in the **.dm** command must be enclosed with ampersands.

Then print the duplicate. Since there's no text in the duplicate, it won't actually print, but Newword will display every data item and variable on the screen, in the appropriate sequence.

If you've made an error somewhere, the display should show it. Depending on your error, the display may stop at the mistake, or it may skip the mistake and start scrambling the data on the document.
screen.

If you're only chaining and nesting files, you don't have to use the special merge-print command in the Opening Menu. The ordinary print command will work. But if you're using variables and data files, you have to use the merge-print command (M).

Data files contain the information used to indi-

vidualize form letters (and other form documents). When you use Newword to type a data file, it's more convenient to type as a **non-document.** But you can also use BASIC, as well as some database management programs to make data files..

Each line of a data file must end with a carriage return.
Data items are the individual pieces of data to be read with .rv.Each must be separated by commas or carriage returns. Insert an extra comma for each missing data item.

If a comma is used in the data item itself, like this ...

Silicon Valley, California,

... enclose it in quotes and put the separating comma outside the ending quotes ...

"Silicon Valley, California",
Either single (') or double quotes (") will work.

Data items can be up to 255 characters long.

Variables with ampersands show places in the form letter where Newword inserts information from the data file.

There must be a data item for every variable named in a .rv command. The sequence of variables must be the same as the sequence of data items in the data file.

Variables' names can be up to 80 characters long.

In a document, "/O" (That's a capital "Oh.") at the end of a variable's name means, "Omit this line if it's completely empty after inserting the data item."

Typing variables' names in upper case isn't necessary, but most people do it.

The .df command must be before the .rv commands, but not necessarily before the .av and .sv commands.

The .av command causes Newword to display the variable's name onscreen while printing, followed by a question mark. You can then type the appropriate data. However, you can also write a message for Newword to display, rather then the variable's name.

When you type your own .av message, it must be typed before the variable's name. The message must be enclosed in quotes, and there must be a comma after the closing quotes.

The .av display, including both message (yours or Newword's) and response, can't be longer than the screen is wide. (Most screens are 80 columns wide.)

If you're displaying messages (using .dm) during merge-printing, the messages can fill the screen. A dot command, .cs, will clear the screen. Put it in the document as often as you need to.

Merge-print dot commands can be used in any file. They affect only the printout.

When you specify **filenames** in .df and .fi commands, you can specify a file on another drive. Just add the drive letter to the name ...

.df b:proposal.dta

In fact, it's a good idea to include the drive loaded in the computer. Just add the word "change" to the command ...

.df b:proposal.dta change

During the printing, Newword will stop when it reaches this command, and display a message telling you to insert the next disk.

Data files don't have to be named .dta.

Merge-print Dot Commands
with examples

.av ask for variable
```
.av PRODUCT
.av  "Type product  name, then press RETURN. ",PRODUCT
```

.cs clear screen
```
.cs
```

.df read data file
```
.df b:products.poo
```

.dm display message
```
.dm This letter is to &FNAME& &LNAME&.
```
.fi insert file
```
.fi a:honey.blr
```

.lm set left margin
```
.lm11*
```

.ls set line spacing
.ls3*

.oj justifying
.ojon/.ojoff

.pf print formatting
.pfdis/.pfon/.pfoff

.rm set right margin
.rm40*

.rv read variable
.rv FNAME, LNAME

.sv set variable
.sv PRODUCT, honey

11.13 EXAMPLES

EXAMPLE 1: CHAIN1.DOC

As you type FIRST.DOC, you're learning some of Newword's functions and commands. For example, did you remember that there's no shift lock on the keyboard? There may be a CAPS LOCK, but you still have to use the shift key to type the question mark. You must have noticed that when you reached the end of a line, the cursor automatically went to the next line. This is called "word wrapping." When you finish typing this sentence, press the RETURN key twice.

As you type FIRST.DOC, you're learning some of Newword's functions and commands. For example, did you remember that there's no shift lock on the keyboard? There may be a CAPS LOCK, but you still have to use the shift key to type the question mark. You must have noticed that when you reached the end of a line, the cursor automatically went to the next line. This is called "word wrapping." When you finish typing this sentence, press the RETURN key twice.

The first time you pressed RETURN, Newword ended the paragraph and sent the cursor to the beginning of the next line. The second time you pressed it, Newword created another line. To make a paragraph indent, use the TAB key or the SPACE BAR.

When the text reaches the bottom of the screen, Newword automatically rolls the entire screenful up underneath the menu, one line at a time.

(You're doing great!)

Now we're going back to the Opening Menu, to learn about the rest of the options listed there. Hold down the CONTROL key with one hand, and press K and then D with the other. The screen will change when you do.

The first time you pressed RETURN, Newword ended the paragraph and sent the cursor to the beginning of the next line. The second time you pressed it, Newword created another line. To make a paragraph indent, use the TAB key or the SPACE BAR.

When the text reaches the bottom of the screen, Newword automatically rolls the entire screenful up underneath the menu, one line at a time.

(You're doing great!)

Now we're going back to the Opening Menu, to learn about the rest of the options listed there. Hold down the CONTROL key with one hand, and press K and then D with the other. The screen will change when you do.

MERGE-PRINTING

As you type FIRST.DOC, you're learning some of Newword's functions and commands. For example, did you remember that there's no shift lock on the keyboard? There may be a CAPS LOCK, but you still have to use the shift key to type the question mark. You must have noticed that when you reached the end of a line, the cursor automatically went to the next line. This is called "word wrapping." When you finish typing this sentence, press the RETURN key twice.

The first time you pressed RETURN, Newword ended the paragraph and sent the cursor to the beginning of the next line. The second time you pressed it, Newword created another line. To make a paragraph indent, use the TAB key or the SPACE BAR.

When the text reaches the bottom of the screen, Newword automatically rolls the entire screenful up underneath the menu, one line at a time.

(You're doing great!)

Now we're going back to the Opening Menu, to learn about the rest of the options listed there. Hold down the CONTROL key with one hand, and press K and then D with the other. The screen will change when you do.

EXAMPLE 2: CHAIN2.DOC

As you type FIRST.DOC, you're learning some of Newword's functions and commands. For example, did you remember that there's no shift lock on the keyboard? There may be a CAPS LOCK, but you still have to use the shift key to type the question mark. You must have noticed that when you reached the end of a line, the cursor automatically went to the next line. This is called "word wrapping." When you finish typing this sentence, press the RETURN key twice.

The first time you pressed RETURN, Newword ended the paragraph and sent the cursor to the beginning of the next line. The second time you pressed it, Newword created another line. To make a paragraph indent, use the TAB key or the SPACE BAR.

When the text reaches the bottom of the screen, Newword automatically rolls the entire screenful up underneath the menu, one line at a time.

(You're doing great!)

Now we're going back to the Opening Menu, to learn about the rest of the options listed there.

Hold down the CONTROL key with one hand, and press K and then D with the other. The screen will change when you do.

--------Page break--------

As you type FIRST.DOC, you're learning some of Newword's functions and commands. For example, did you remember that there's no shift lock on the keyboard? There may be a CAPS LOCK, but you still have to use the shift key to type the question mark. You must have noticed that when you reached the end of a line, the cursor automatically went to the next line. This is called "word wrapping." When you finish typing this sentence, press the RETURN key twice.

The first time you pressed RETURN, Newword ended the paragraph and sent the cursor to the beginning of the next line. The second time you pressed it, Newword created another line. To make a paragraph indent, use the TAB key or the SPACE BAR.

When the text reaches the bottom of the screen, Newword automatically rolls the entire screenful up underneath the menu, one line at a time.

(You're doing great!)

Now we're going back to the Opening Menu, to learn about the rest of the options listed there. Hold down the CONTROL key with one hand, and press K and then D with the other. The screen will change when you do.

MERGE-PRINTING

INTERVIEWS WITH ARTISANS

JOHN HAULE, WOODWORKER

"A friend asked me to do a dictionary stand. He was a writer, you see. The last thing I'd done for him was a beautiful desk, one of my favorite things to do. So many possibilities for turning something functional into something beautiful. You know, the intersection of drawer front and desk frame can be very subtle and suggestive. Just rounding the edge of the drawer front, for instance, instead of leaving a sharp, right angle, changes the way you use the desk, changes the way you think about yourself while you work. I've had customers tell me that my desks have transformed their careers. It doesn't surprise me. I feel the same way. Maybe that's why I do what I do, because I can feel it in the wood.

"Everything's all of a piece in this kind of work. I'm a business man. I make things and sell them. I do my own bookkeeping--got a computer last year for that, and I'm learning now to do some of my design work on it. Stress analysis, things like that. It surprised me, the computer. A friend talked me into borrowing hers while she was off on a trip somewhere. It's a tool, just like any other tool. Every piece of equipment here in the shop has its good points and its bad, its personality. I found myself learning about the computer just the way I learn about any other new tool. Use it for a while, watch how it amplifies my . . . oh, how it amplifies my input, I guess. If I feed the wood this way, the tool cuts and shapes it that way. Okay, so I remember that, for next time. And I have to maintain the shop, take on apprentices and let them learn and develop while still protecting the integrity of my work. And, after all these years, and with my great reputation and everything, I still have to hustle for work sometimes. I still have to make a buck.

--------And so on--------

EXAMPLE 3: FORM1.LTR

Oakenshield Hall
Minas Tirith, Ithilien
Rethe 11, 1464 S.R.

Fremore Threngen
Farmore
Folde
Eastfold, Rohan

Dear Friend Fremore:

I've been asked to serve as agent for a small, rural cooperative producing specialty foodstuffs and crafts items. Currently, they're offering the most delicious honey I've ever tasted, packaged in a whimsical, bear-like ceramic pot.

You're sure to enjoy the sample I've asked them to send you. It should be arriving very soon.

Do write to me, Fremore, if you'd like to establish a trade in the honey, or if you're interested in their other wares. You won't be disappointed.

Yours very sincerely,
in friendship and trade,

Gimli Gloin, Elf-friend
Lord of the Glittering Caves
Agent in Trade

MERGE-PRINTING

Oakenshield Hall
Minas Tirith, Ithilien
Rethe 11, 1464 S.R.

Bombur Heavyfoot
Commissary
The Mountain
Erebor, Dain

Dear Friend Bombur:

I've been asked to serve as agent for a small, rural cooperative producing specialty foodstuffs and crafts items. Currently, they're offering the most delicious honey I've ever tasted, packaged in a whimsical, bear-like ceramic pot.

You're sure to enjoy the sample I've asked them to send you. It should be arriving very soon.

Do write to me, Bombur, if you'd like to establish a trade in the honey, or if you're interested in their other wares. You won't be disappointed.

Yours very sincerely,
in friendship and trade,

Gimli Gloin, Elf-friend
Lord of the Glittering Caves
Agent in Trade

Oakenshield Hall
Minas Tirith, Ithilien
Rethe 11, 1464 S.R.

Sam Gamgee
Mayor of the Shire
Hobbiton, the Shire

Dear Friend Sam:

I've been asked to serve as agent for a small, rural cooperative producing specialty foodstuffs and crafts items. Currently, they're offering the most delicious honey I've ever tasted, packaged in a whimsical, bear-like ceramic pot.

You're sure to enjoy the sample I've asked them to send you. It should be arriving very soon.

Do write to me, Sam, if you'd like to establish a trade in the honey, or if you're interested in their other wares. You won't be disappointed.

Yours very sincerely,
in friendship and trade,

Gimli Gloin, Elf-friend
Lord of the Glittering Caves
Agent in Trade

EXAMPLE 4: ENVELOPES FOR FORM1.LTR

Fremore Threngen
Farmore
Folde
Eastfold, Rohan

Bombur Heavyfoot
Commissary
The Mountain
Erebor, Dain

Sam Gamgee
Mayor of the Shire
Hobbiton, the Shire

MERGE-PRINTING

12. SHORTCUTS

12.1 CONGRATULATIONS AGAIN!

You probably don't feel that you really know Newword yet. That's very understandable, especially if word processing is a new experience for you.

But don't worry--you're doing just great!

Take your time, adding one or two commands at a time to your bag of tricks. When you first try new ones, keep the situation simple, if you can. You'll be able to see more clearly how the new commands work.

The index here in the **Do It Yourself** lists every section and document, as well as a few key topics, mostly from chapters 9-13. It isn't a complete index, because once you've finished the **Do It Yourself**, you should rely on the job aids and the **Encyclopedia.** Most people use them quite a bit while they're becoming proficient with Newword.

12.2 TIPS FROM THE EXPERTS

Here are a few suggestions experienced word processing operators have made. Some of them are more appropriate for people who do word processing all day long, than for those of you who do it only occasionally.

In any case, we think trying these out can prove to be helpful ...

In general, learn to use the least number of keystrokes. There are often several ways to do something--get in the habit of looking for the way that involves the least keystrokes.

For example, ^f and ^a often move the cursor between paragraphs with fewer strokes than ^x and ^e. And, to put the cursor on a word near the end of a line, it's easier to first move the cursor **below** the line, and then move it backwards to the word, with ^a. ^h can be a handier way to erase characters than the DELETE key, too.

Leave the insert function **on.** When you're making corrections, insert the new characters, then erase the old ones. Turn insert off only when it's clearly more convenient.

Use the lowest help level that's practical. This is a long-term goal, of course.

Learn all the uses of ^u. It cancels commands, including ones like ^qf with the g option, and ^qb.

But it also "unerases" typing mistakes. And it can move text, because the unerased word reappears **wherever the cursor is.** For example, if you erase a word and then press ^u **immediately afterwards,** the word will reappear where it was originally typed. But if you erase the word and then **move the cursor** before pressing ^u, the erased word will reappear at the new cursor position.

When you type a new ruler line, type it as an automatic ruler (.rr) or a dot message (..). If, instead, you just change the existing ruler line, with the margin or tab commands, use ^oo to bring the ruler into the document, and then change it to an automatic ruler or dot message.

Some people make ruler files, too. In a file named RULERS, type the 3 or 4, or however many, ruler lines you frequently use. When you need one, ^kr will bring RULERS into your document. Use ^of to activate the new line and ^oo to put it in the document as an automatic ruler or dot message. Then mark RULERS as a block and erase it.

If you do word processing for hours at a time, you may find that some commands can be entered more conveniently with the CONTROL key than with your keyboard's extra function keys. The CONTROL key

lets you keep your hands in the **ASDF-JKL;** position.

During **find** and **find and replace** operations, mark your place with a dot message, so you can get back to it easily.

It's often more convenient to search backwards through a document, rather than forwards.

When you're typing a file's name, e.g., at the Opening Menu, if you enter ^r instead of actually typing the name, Newword will write the name you used the last time. Newword remembers the previous file name, so it can make backup files, etc., and ^r writes the name.

Finally, you'll sometimes find it convenient to "beat" Newword, that is, to enter a command before Newword asks you for it.

For example, suppose you save the document you're working one and open another. When you enter ^kd, Newword takes a little while to store the document. Before Newword displays the Opening Menu, you can type **d**, the command for opening another document.

Be careful with this. You have to know the right command. And don't get so far ahead of Newword that you can't proofread your entries.

CONGRATULATIONS!AGAIN!CONGRATULATIONS!AGAIN!CONGRATULATIONS!AGAIN!
CONGRATULATIONS!AGAIN!CONGRATULATIONS!AGAIN!CONGRATULATIONS!AGAIN!
CONGRATULATIONS!AGAIN!CONGRATULATIONS!AGAIN!CONGRATULATIONS!AGAIN!

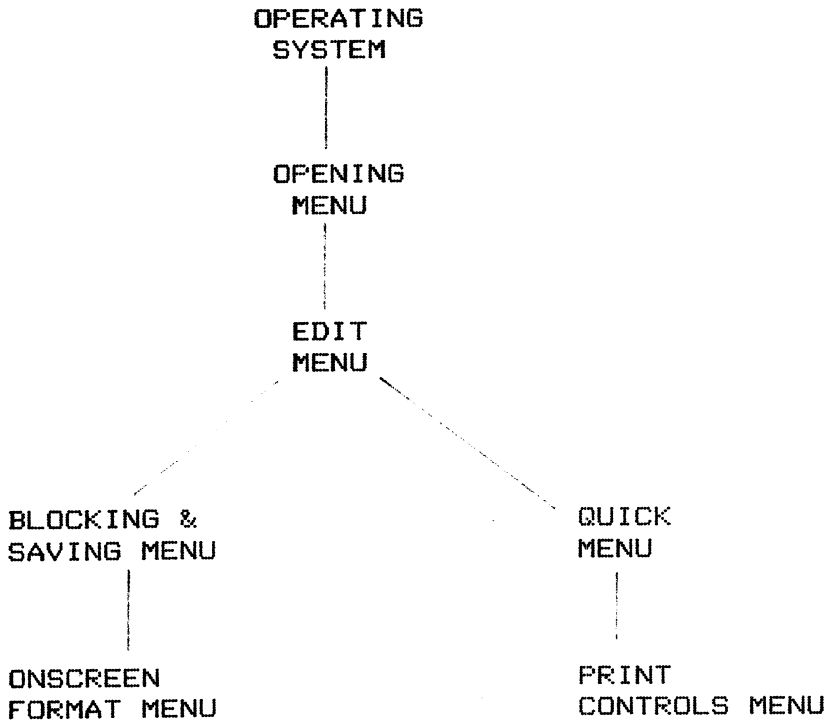
SHORTCUTS

13. JOB AIDS

On the following pages, we've typed each of Newword's job aids. You may also have been provided one-page, typeset copies of them.

13.1 FLOWCHART

The flowchart's being prepared. It will show that you first load Newword at the operating system, then select a command at the Opening Menu, then use the other menus to create and edit documents. Here's a very rough approximation ...



13.2 FUNCTION CARD

OPENING MENU FUNCTIONS

copy file	O
change logged drive	L
display directory	F
erase file	Y
exit	X
help	J
merge-print file	M
open document	D
open non-document	N
print file	P
protect file	C
rename file	E
run a program	R

OTHER FUNCTIONS

BLOCKING

copy	^kc
erase	^ky
hide/show markers	^kh
mark end	^kk
mark start	^kb
move	^kv
store on disk	^kw

CURSOR

char left	^s
char right	^d
down to lower right	^qx
line down	^x
line up	^e
end of line, left	^qd
end of line, right	^qs
tab	^i
up to upper left	^qe
word left	^a
word right	^f

ERASING

block	^ky
char left	DEL
char left	^h
char right	^g
file	^kj
line	^y
to left end line	^qDEL
to right end line	^qy
word right	^t

FILE OPERATIONS

change logged drive	^kl
copy file	^ko
erase file	^kj
insert file	^kr

FINDING

find	^qf
find & replace	^qa
find page	^qp
find/replace again	^l

FOOTERS & HEADERS

alternate corners	^pk
footer, 1st line	.fo/.f1
footer, 2nd line	.f2
footer, 3rd line	.f3
header, 1st line	.he/.h1
header, 2nd line	.h2
header, 3rd line	.h3
literal character	\
page number here	£

FORMATTING

Onscreen, Dot Commands

conditional page	.cp
dot message	..
line height	.lh
page break	.pa
page length	.pl

Onscreen, Typing

hyphen help	^oh
justifying	^oj
soft hyphen	^oe
word wrap	^ow

Printer

bidirec print	.bp
character width	.cw
microjustification	.uj

Printout

def'lt pg. no. col.	.pc
justifying	.oj*
omit def'lt pg. no.	.op
page N	.pn
page offset	.po
pg. no., foot/head	£
print formatting	.pf
restore def'lt no.	.pg

* Activated by .pfon

HELP

help levels	^j^j
help messages	^j

MARGINS, LINES & TABS

Onscreen

center line	^oc
line spacing	^os
margin release	^ox
margin, set left	^ol
margin, set right	^or
ruler display	^ot
ruler from text	^of
ruler to text	^oo
ruler, automatic	.rr
tab, cursor to	^i
tab stop, clear	^on
tab stop, set	^oi
tab stop, set (non-document)	^o
temporary indent	^og

Printout

bottom	.mb
footer	.fm
header	.hm
left	.lm*
line spacing	.ls*
right	.rm*
top	.mt

* Activated by .pfon

MERGE-PRINTING

clear screen	.cs
data file	.df
display message	.dm
insert file	.fi
omit null line	/O
variable, ask for	.av
variable, begin/end	&
variable, read	.rv
variable, set	.sv

PRINT CONTROLS

binding space	^po
bold print	^pb
custom 1	^pq
custom 2	^pw
custom 3	^pe
custom 4	^pr
doublestrike print	^pd
overprint char	^ph
overprint line	^pRET
pause	^pc
phantom char	^pf
pitch, alternate	^pa
pitch, standard	^pn
ribbon	^py
strikeout	^px
subscript	^pv
superscript	^pt
underline	^ps

SAVING

quit file	^kq
save & exit	^kx
save & resume edit	^ks
save file	^kd

SCROLLING

bottom of document	^qc
line down	^z
line up	^w
screen down	^c
screen up	^r
top of document	^qr

13.3 COMMAND CARD

OPENING MENU COMMANDS

C protect file
D open document
E rename file
F display directory
J help
L change logged drive
N open non-document
O copy file
P print file
R run a program
T transmit file
X exit to operating system
Y erase file

OTHER COMMANDS

^a cursor word left

^b align paragraph

^c scroll down screen
(resume, when printing)

^d cursor char right

^e cursor up line

^f cursor word right

^g erase char right

^h erase char left

^i cursor to tab
(column tab, non-doc)

^j help

^k Blocking & Saving Menu
^kb start block
^kd save file
^ke rename file
^kh hide/show block markers
^kj erase file
^kk end block
^kl change logged drive
^ko copy block
^kq quit file
^kr insert file from disk
^ks save & resume edit
^kv move block
^kw store block on disk
^kx save & exit
^ky erase block

^l find/replace again

^m carriage return

^n paragraph line
(auto indent, non-doc)

^o Onscreen Format Menu
(set tab stops, non-doc
^oc center line at cursor
^od display control chars
^oe soft hyphen
^of ruler from text
^og temporary indent
^oh hyphen help
^oi set tab stop
^oj right justify
^ol set left margin
^on clear tab stop
^oo ruler to text
^or set right margin

JOB AIDS

^os set line spacing
^ot display ruler
^ow word wrap
^ox margin release

^p Print Controls Menu
(pause, when printing)
^pa alternate pitch
^pb bold print
^pc print pause in file
^pd doublestrike print
^pe custom print control 3
^pf phantom character
^ph overprint character
^pn standard pitch
^po binding space
^pq custom print control 1
^pr custom print control 4
^pRET overprint line
^ps underlining
^pt superscript
^pv subscript
^pw custom print control 2
^px strikeout
^py ribbon color

^q Quick Menu
^qa find & replace
^qb align document
^qc bottom of document
^qd cursor right end of line
^qDEL erase to left end
of line
^qe up to upper left
^qf find
^qp find page
^qr top of document
^qs cursor left end of line
^qy erase to right end
of line

^r scroll up screen

^s cursor char left

^t erase word right

^u unerase

^v insert

^w scroll up line

^x cursor down line

^y erase line

^z scroll down line

DOT COMMANDS

.av ask for variable

.bpon/.bpoff bidirec print

.cp conditional page
.cs clear screen
.cw set character width

.df read data file
.dm display message

.fi insert file
.fm set footer margin
.fo footer, 1st line
.f1 footer, 1st line
.f2 footer, 2nd line
.f3 footer, 3rd line
.fo^pk alternate corner
(ok to use .f1, etc.)

.he header, 1st line
.h1 header, 1st line
.h2 header, 2nd line
.h3 header, 3rd line
.he^pk alternate corner
(ok to use .h1, etc.)

.lh set line height
.lm set left margin*
.ls set line spacing*

.mb set bottom margin
.mt set top margin

.ojon/.ojoff justify on/off*
.op omit default page number

.pa page break
.pc set default page no. col.

.pfon/.pfoff/.pfdis print format
.pg restore default page nos.
.pl set page length
.pn set page number
.po set page offset

.rm set right margin*
.rp print "N" times
.rr automatic ruler
.rv read variable

.sv set variable

.ujon/.ujoff microjustification on/off

* Activated by **.pfon**

SYMBOLS

^ CONTROL key
£ print page no. here (footers & headers)
<B> begin block
<K> end block

**** literal character (footers & headers)
/O omit null line (merging)
& begin/end variable, in text

JOB AIDS

JOB AIDS

14. INDEX

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

Abridged Terms & Conditions (Downloads)

Disclaimer

www.primrosebank.net, (***the website***) is provided by Dave Stevenson as a service to the public, is provided "as is" and carries no warranties, expressed or implied, of any kind.

Dave Stevenson is not responsible for, and expressly disclaims all liability for, damages of any kind arising out of use, reference to, or reliance on any information contained within the website or made available for download. Whilst the information contained within the website site is periodically updated, no guarantee is given that the information provided on the website is correct, complete, and up-to-date.

A number of articles on the website contain technical data and practical guidance which may be of use in testing and maintaining various items of vintage computer and electronics hardware. Such articles are not intended to cover all aspects of the tasks involved and may omit essential information, including necessary safety precautions. Performance of the tasks described may risk damage to equipment and/or people. The reader is responsible for ensuring that he/she is capable of performing the tasks described and well as assessing the inherent risks involved and taking appropriate measures to mitigate such risks.

Dave Stevenson expressly disclaims all liability for, damages to equipment or injury of any kind arising out of use of such technical data and guidance.

As far as possible, information included on the website from other sources has been credited to the respective author and/or publisher. The majority of content on the website is derived from material first published in the 1980s. ***This material is likely still under copyright of the original author and/or publishers.*** The authors and/or publishers may not have given express permission to copy, transmit or make this information available for download, but I believe that they would have no objection to this archive information being placed into the public domain.

However, should the author and/or publisher of the original material find any content on the website for which they wish to assert their rights, they should notify Dave Stevenson (by e-mail to: webmaster@primrosebank.net) who would be pleased to enter into a dialogue to agree a satisfactory resolution of their concerns.

If you obtained this file as part of a paid-for package, you have been scammed! I suggest that you request a refund from the seller, please also advise Dave stevenson at the e-mail address above.