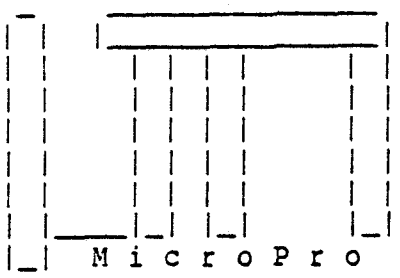


MicroPro International Corporation

WORD-MASTER(tm)

OPERATORS GUIDE

MicroPro International Corporation
1299 Fourth Street
San Rafael, California 94901
(415) 457-8990
TELEX 340388 MICROPRO SRFL

TABLE OF CONTENTS

W O R D M A S T E R

O P E R A T O R S G U I D E

INTRODUCTION

Transcribing text has been with us since the days of the scribe, that literate person with great skill in hand lettering. But, while literacy became more widespread, skill at hand lettering still belonged to a minority. As society became more complex and the amount of written communication increased, it was no longer workable for a scribe to spend years copying a book. In the constant quest for more efficient methods, several developments took place. Shorthand for oral transcription, mechanical letter printers for uniform legibility, followed by dictation equipment, manual typewriters, automatic typewriters, and finally, the computer.

With the computer, we have the beginning of a new type of improvement not only in productivity but in the very nature of secretarial practice. Document transcription (transcribing documents), has, at best, been a tedious, repetitive, exhausting task for the secretary. Power typing, as exemplified by the magnetic card storage typewriter, was a real breakthrough in that the step from rough draft to final copy was eased considerably. Repetitive typing of letters, for example, was made much easier, since only the names and addresses had to be retyped.

WORD-MASTER, a software product developed and licenced for use to others by MicroPro International Corporation represents an important contribution to low-cost yet very capabable text editing and document preparation technology. Designed for use by office personnel familiar with a standard typewriter keyboard, this Operators Guide will provide an operator with information to prepare and manipulate documents with significant ease and efficiency.

GETTING STARTED

We will begin by doing some "cookbook" exercises to show off the system, dazzle your mind and quicken your curiosity for reading the rest of this guide.

Turn on your computer; slip in a diskette with **WORDMASTER** on it; and bring up the CP/M system. If you don't know how to do this, better refer to the turn-on procedure for your particular computer before proceeding further. When the A> appears, type:

A>WM EXAMPLE.RSC (and then press the RETURN button)

After a few seconds, you'll see the MicroPro copyright notice, a message that says "NEW FILE", after which the screen blanks and a white rectangular symbol, called the cursor, appears in the upper left hand corner of the screen. **WORDMASTER** is now ready for use. Use the keyboard illustrations in the back of this manual or, preferably, have your terminal ready for use.

Look at your keyboard. Look at the key to the left of the keyboard labeled CTRL. This key is the "key" to the power of **WORDMASTER.** Whenever you see an up-arrow ^ in this manual, it means that you press the CTRL key. Thus to enter ^R, you press CTRL and, without releasing the CTRL key, you press R.

Now, look at the right side of the keyboard and locate the RETURN key. Wherever you see a <CR> in this manual, it means that you press the RETURN key.

Now look at the left side of the keyboard and locate the ESCAPE key (sometimes labeled ESC). Whenever you see a dollar sign $ in this manual, it means that you press the ESCAPE key.

Let's try something to make sure we understand each other. Type the following regardless of what happens while you are typing:

$QLThe quick brown fox jumped over the lazy dog.^N<CR>
12QGBV<CR>

If you typed EXACTLY what is shown, regardless of what went on while you were typing, you should now see twelve lines of the famous nimble fox. If you don't, go back and read this section from the beginning and start over. I'm sure you'll succeed this time.

Now type:

```
$<sbrown$blue$><CR>
BV<CR>
```

The BROWN fox is now BLUE!

Now type:

```
^F^T
```

You're really getting into it if the first fox is no longer speedy.

Now type:

```
^B^B^A^Obrown ^O
```

If you now know the color of the first lazy dog, you're set for the rest of this book.

Now type:

```
$Q<CR>Y<CR>
```

In addition to seeing the message ABORT (Y/N)?, the familiar A> will appear when you finish typing this.

Don't worry if you feel confused at this point. By the time you finish this users guide, everything that just happened should be totally clear to you.

WORD-MASTER MODES

There are three different operating "modes" in **WORDMASTER.** This means that certain keys on the keyboard will cause different actions depending on the current mode you are using. **WORDMASTER**'s three modes are the Video Mode, the Command Mode and the Insert Mode.

The differences between modes may be summarized as follows:

- In Video Mode, you have the opportunity to see the appearance of the text as you make alterations. (If you are familiar with CP/M's editor, it may help you to know that the cursor position on the screen is a true representation of where the "pointer" is.) You can position the cursor in all four directions, page through text back and forth, insert and delete lines or part of lines and input or delete characters and words.

- In Command Mode, you can do everything you can in Video Mode, but any changes in the text are only displayed on your command. In addition, other facilities are possible including searching, substituting, looping, matching and moving around blocks of text. You can also combine and/or separate text from/to different files on your diskettes.

- Use Insert Mode for all original entry and for large insertions. In this mode, a backspace also erases the last character typed. You cannot backspace to a previous line.

We leave and enter the different **WORDMASTER** modes as follows:

- **WORDMASTER** always starts in Video Mode.
- To enter Command Mode from Video Mode or Insert Mode, press the ESC key (indicated by a "$")
- To enter Video Mode from Command Mode type: V<CR>
- To enter Insert Mode, first enter Command Mode and type: I<CR>

Let's perform some exercises to try out our new capabilities. Begin by typing:

```
A>WM EXAMPLE.RSC<CR>
```

What we just did was to call up the **WORDMASTER** program (WM) and tell it we want to do some work on a file called EXAMPLE.RSC. Since this file did not exist on the disk, **WORDMASTER** tells us that it is a new file. The screen will now be clear (we are in the Video Mode) and you may now enter text. Go ahead and type in the following paragraph. For the sake of this exercise, press RETURN at the end of each line of text as indicated by the "<CR>".

The Video Mode in **WORDMASTER** provides a new level of ease and<CR>
convenience in inspecting and updating text files when using a<CR>
CRT terminal or video display. Control keys permit moving to<CR>
any point in the file without leaving Video Mode. The contin-<CR>
uously updated display speeds editing and reduces mistakes.<CR>

After you've typed the passage and pressed the final carriage return, type:

$H<CR>

After a few seconds, the screen will clear and the text you typed will reappear. You just used **WORDMASTER** to enter text into the computer's memory, exited the Video Mode and entered the Command Mode by pressing "$" (ESCAPE) and then used the "H" command. The "H" command is usually used during a long edit so that the file on the disk is up to date. This ends the edit, updates the disk with the text in the memory and re-enters the Video Mode so you can do some more work on the text you just entered. Whenever you use the "H" command, the previous version of the file (if there is one) will be renamed NAME.BAK for BAcKup file (where "NAME" is the file name). Imagine after spending two or three hours typing something into the computer there is a power failure or you accidently delete half of the text you were working on! All of the work you just did would be lost! However, if you had used the "H" command every twenty minutes or so, you would only have to re-do those last twenty minutes instead of those last 2-3 hours.

So, there is now a file on the disk called EXAMPLE.RSC with the above passage stored in it.

```
**************************
*                        *
*   E X E R C I S E S    *
*                        *
**************************
```

It is very IMPORTANT to WATCH THE SCREEN as you do these exercises so you observe what happens with each of your keystrokes!

It is also important to follow the "recipes" exactly as indicated. You will probably want to try one of the commands a few extra times. Restrain yourself if you can. But, if you can't resist and happen to make a mistake which you don't know how to correct, you can always bring the file back up from the disk by typing:

$O<CR>Y<CR>	to cancel current **WORDMASTER** operations and bring it back up again with your saved file.

I. THE VIDEO MODE

A. WORD TAB. While holding the CTRL key down, strike the letter F several times. Note that the cursor jumps to the first letter of each successive word. Still holding the CTRL key down, strike the letter A. Note that the cursor moves to the first letter of each word; i.e. it's a back tab operation. Thus:

^F is word tab forward and ^A is word tab back

B. 4-WAY CURSOR MOVEMENT. Holding the CTRL key down, alternately strike the letters S, D, E and X.

Note that:
S will move the cursor left one character,
D is cursor right one character,
E is cursor up one line and
X is cursor down one line.

Moving the cursor to any position does not alter the text in any way.

C. SCREEN TAB. Holding the CTRL key down, strike the key HOME. The cursor will move to the upper left hand corner of the screen. Still holding the CTRL down, strike the key HOME again. The cursor will then move to the first letter on the last displayed line. Thus:

^HOME(^^) is screen tab toggle

A "toggle" key is a single key that alternates between two functions. In this case, it first positions the cursor at the top of the screen, and then at the bottom of the displayed text.

D. LINE TAB. Type ^HOME(^^) to bring the cursor in the upper left hand corner.

Then type:

<CR><CR><CR>^P^P^P^P

The cursor is now positioned over the "p" in the word "point" in line four. Holding the CTRL key down, strike the letter B. The cursor will move to the first character of the line (over the "a" in "any"). Type ^B again and the cursor moves just to the right of the last character on the line.
Thus:

^B is the line tab toggle

E. LINE INSERT and DELETE. First type:
^HOME<CR><CR> to place the cursor over the "C" of "CRT"
then: ^N^N to insert two blank lines
and: one<CR>two^E

What should have happened is that you inserted two new lines which now occupy lines three and four on the screen with the words "one" and "two", respectively. The cursor position should be just to the right of the word "one" on line three. So:

^N is line insert

Now type:

^Y^Y

The inserted lines have disappeared and the succeeding lines have moved up to fill the gap.

^Y is line delete regardless of cursor position on the line.

The cursor should now be over the "C" in "CRT". Now type:

^F^F^F^N

All the words on the line to the right of the word "or" have moved down one line and over to the left margin, pushing all succeeding lines down one line. Thus:

^N is also line insert in mid-line.

F. CHARACTER INSERT and DELETE. You can turn the character insertion toggle on and off with ^O. You know when the insertion toggle is on (i.e. active), because the the cursor is shown overlaying a < sign regardless of cursor position; the character occupying the cursor position is not visible unless you move the cursor.

Let's try it. First type:

^X^X^F

The cursor is now over the "f" in the word "file". Next type:

^Othis phrase is now being inserted

One of the first things you will notice as you typed is that the characters on the far right of the line were pushed over the edge of the screen and "wrapped around" to the far left of the screen on the next line down. The characters ">>" also appeared on the left to remind you that this line is one continuous line; that is, the RETURN key was not pressed at the end of the line. What this means is that the line of text is longer than the screen's width.

All functions, including cursor movement, line insert/delete, forward/back word-tab, word and character delete, and paging commands operate the same regardless of whether the insertion toggle is on or not. The RETURN key, however, behaves somewhat like ^N when the insertion toggle is on. Using ^N for line inserts leaves the cursor where it was. With the insertion toggle on, the RETURN key inserts a new line at the point of the cursor and moves the cursor to the first character of the new line.

Now type:

<CR><CR>^O

The first RETURN you typed moved the text to the right of the cursor down to the next line, the second RETURN inserted a blank line, and the ^O turned off the insertion toggle.

^O is the insertion toggle

Now let's try deleting text. Type:

^E^E^F^F^F^F

The cursor should now be over the "t" in the word "this"

Now type:

^G^G^G^G^G

You just deleted all four letters of the word "this" and the space following it. Use ^G to delete the character under the cursor. When you use ^G in this way, all the characters to the right of the vacated position move to the left one position.

Next type:

^E^E

The cursor will now be two spaces to the right of the word "or".

Now type:

^D^S^D^S

Well, you just moved the cursor over the invisible RETURN character which we inserted with a ^N earlier. Picture the RETURN as a dam which breaks up a series of words into lines. Watch what happens when we delete this dam.

Type:

^G

As you already know, ^G deletes the character it is over, in this case, a RETURN. The dam has been broken and the next line down merges with the first line.

Now try typing:

^F^X^D

The cursor is now over the space between the words "phrase" and "is". Look at your keyboard and find the key with DEL on it (RUB OUT on some keyboards). Character deletion may also occur to the left. Now press the DEL key three times and notice that the "ase" has been deleted.

To summarize:

^G is delete character under the cursor
DEL (or RUB) is delete character left

G. FULL or PARTIAL WORD and PARTIAL LINE DELETION.

First type:

^A^T^T

The cursor was first placed on the "p" of what was the word "phrase". Then "phr" was deleted when you typed the first ^T and " is" was deleted with the second ^T.

Use ^T to delete from the cursor position to the first character of the next word. When your words contain special characters such as (), ',", #, $, or &, etc., the definition of what is a word changes. Please refer to the **WORD-MASTER** Users Guide for detailed information.

Now type:

^F^F^\^\

First the cursor was placed over the first character of the word "inserted", then the two previous words were deleted. Make sure you used the backslash \ character, not the slash /.

^T is delete word right
^\ is delete word left (NOTE: SHIFT^L on some terminals)

Next type:

^U

You may now notice that everything to the left of the cursor has VANISHED. The word "inserted" is all that is left of the line.

Now see if you can place the cursor over the "V" in the word "Video" two lines down. When you have done so, type:

^K

And notice that everything to the right of the cursor is gone. Therefore:

^U is delete line left
^K is delete line right

H. REPEAT COMMAND. Try typing the following:

^HOME^@^D

First you placed the cursor at the upper left corner of the screen, then the cursor moved over four spaces.

Now try this:

^@^F

This time the cursor moved over four words. The command ^@ will cause the command which follows it to repeat FOUR times.

Now type:

^@^@^F

And the cursor moved ahead 16 words. It did not move it ahead 8 words. That's because the first ^@ caused the second ^@ to do its thing 4 times, or 4 times 4 equals 16.

Just to make sure, type:

^@^@^@^F

The cursor zoomed ahead 64 times (4 times 4 times 4 equals 64). However, we have now reached our limit: ^@^@^@^@ will not give us 256 times but only 4 times.

I. SCREEN SCROLLING. First, check to see how many lines of text your screen can display. It will probably be either 16 or 24 lines.

Depending on that number, type one of the following:

16 lines ^HOME^HOME^@^@<CR>This is page 2<CR>^@^@<CR>This is page 3<CR>

24 lines ^HOME^HOME^@^@<CR>^@<CR>^@<CR>This is page 2<CR>
^@^@<CR>^@<CR>^@<CR>This is page 3<CR>

First we placed the cursor at the bottom of the text. Then we used enough ^@'s with RETURNs <CR> to make the text scroll off the top of the screen:

4 X 4 = 16 for 16-line screens
4 X 4 + 4 + 4 = 24 for 24-line screens

Next we typed a phrase indicating the page number, added another 16 or 24 RETURNs and another page number.

We now have three "screenfuls" of text. Now, how do we get to see them? We use the SCROLLING commands! Let's try them - type:

^R^R

Each time you type ^R, the file of text moves DOWN one screenful, that is, we move towards the beginning of the file.

Now type:

^C^C

Each time you type ^C, the file moves UP one screenful, towards the end of the file. If you reach the end or the beginning of the file, nothing will happen (where could it go?!).

Now try typing:

^W^W^W^W^W

So, ^W moves the file DOWN one line, so we just moved the file DOWN 5 lines.

And now type:

^@^X

And the file moves back UP four lines.

To summarize:

^R is file DOWN one screenful
^C is file UP one screenful
^W is file DOWN one line
^X is file UP one line

II. THE COMMAND MODE

We will now go back to those original examples and take them apart so they make sense.

A. JUMP to BEGINNING/END of TEXT.

First type:

$ (remember, that is the ESCAPE key)

When in Video Mode or Insert Mode, pressing the ESCAPE key will always put us into the Command Mode.

Next type:

B<CR>

Beginning of the file. It's there even though you can't see it. In Command Mode, there are some powerful commands, but to actually have them displayed, we must be in the Video Mode. So type:

V<CR>

Now we are in the Video Mode with the cursor over the first character of the file.

Now type:

$-BV<CR>

We are now back in the Video Mode with the cursor over the last screenful of text. Typing a "-B" will move you to the end of the file. If "B" means the Beginning of the file, then of course "-B" would mean the opposite of the Beginning, or the End.

Before we go on to the next command, lets get rid of all the garbage we now have in the file and start over again. Since we previously saved the paragraph you typed, let's get it back. Type:

$O<CR>Y<CR>

The "O" command gets us back the original version of the file (or the version of the file as it was after the last "E" command). Since this could be dangerous if typed by accident (you could lose any new corrections or updates to the text), the question "CONFIRM(Y/N)?" appeared to give you a second chance. Typing "Y"<CR> carried out the command. Typing anything else would have returned you to the Command Mode. You should now have the paragraph displayed on the screen as it was when you finished typing it.

NOTE: The commands which write the file to the disk or exit **WORDMASTER** (so far, H,O,Q) can be the only commands on the line. That is, typing "HV" will give you an error (???). You must type "H"<CR>"V"<CR> instead.

B. SCRATCHPAD COMMANDS. The Scratchpad is a special box where we can temporarily store text which could be commands to be executed from the Scratchpad or text that we want to move to a different place in the file. All Scratchpad commands begin with "Q" or "/Q".

Type:

$3QP<CR>

What we just did was to take the next 3 lines starting at the cursor and Put them in the Scratchpad. Type:

V<CR>

We are now back in Video Mode. Notice that the first three lines of the paragraph are gone. When you use the "QP" command, the text you Put in the Scratchpad is deleted from the file and anything that was in the Scratchpad box before is now gone.

Now type:

^HOME^HOME to move the cursor to the bottom of the paragraph

and: $QGV<CR>

Notice that it's alright to type more than one command before hitting RETURN while using Scratchpad commands. (This is also true of Command Mode commands, but more on that later!).

The 3 lines which we Put in the Scratchpad have been copied into the text at the position of the cursor. "QG" means Get the contents of the Scratchpad and insert them at the cursor. The cursor is moved to the end of the inserted text.

Next type:

$7QGV<CR>

By typing the "7" before the "QG" command, we copied the contents of the Scratchpad into the text 7 times. Use the scrolling commands to see them all.

Type:

$QT<CR>

The three lines that are still in the Scratchpad are Typed on the screen. "QT" means Type (display) the contents of the Scratchpad.

Now type:

B2QP<CR>

As before, the cursor went to the beginning of the file (even though you didn't see it) and the next 2 lines were Put into the Scratchpad.

To check it, type:

QT<CR>

Notice that the 3 lines that were in the Scratchpad before have been replaced by the two new lines.

Next type:

3/QP<CR>

The command "/QP" means the same as "QP" but this time, the new lines are added on to the end of whatever was in the Scratchpad without emptying it (appended).

Check it by typing:

QT<CR>

There are now five lines in the Scratchpad.

Type:

QKQT<CR>

Nothing happens! That's because "QK" means Kill the Scratchpad (that is, empty it out). So when you typed the "QT" command, it Typed out the now empty Scratchpad.

Do you remember those exercises we did in the beginning of this guide? Do you remember when I said that they would all be totally clear to you? We are now to the point where you can understand what happened then. First, let's start with a clean slate. Type:

Q<CR>Y<CR>

Typing a "Q" by itself stands for the word "Quit". This means leave **WORDMASTER** without saving the current edited text on the disk. Again, this could be dangerous if typed by accident, so the question "CONFIRM(Y/N)?" appeared to give you the chance to change your mind. When you typed Y<CR>, you left WM and returned to CP/M (A>). Typing anything else would have returned you to the Command Mode.

Now let's re-enter **WORDMASTER** and open a new file on the disk. Type:

WM EXAMPLE2.RSC<CR>

A new file named EXAMPLE2.RSC has been created.

One of the first exercises you did in this guide used the Scratchpad. Let's try it again now that you can understand how it works. First, place the cursor at the end of the file. If you don't remember how to do this, check back to section II.A.

Next, while in the Command Mode, type:

QLThe quick brown fox jumped over the lazy dog.^N<CR>

The command "QL" means Load the following text into the Scratchpad. The "^N" at the end of the sentence means to include a RETURN as part of the text.

Check the Scratchpad by typing:

QT<CR>

The line, "The quick brown fox..." should be displayed.

Now type:

12QGBV<CR>

See if you can figure out what happened before I explain it to you.

The first command, "QG" was executed 12 times. That is, we Got the contents of the Scratchpad 12 times and inserted them into the text. Next the cursor was placed at the Beginning of the file and we entered the Video Mode. We should now have 12 lines of the brown fox jumping.

Another Scratchpad command is "/QL". It does the same thing as "QL" but, the contents of the Scratchpad are not emptied out first, and the series of characters (called a "string") which follow the command are appended to the contents of the Scratchpad.

Let's summarize the Scratchpad commands. The "n" below represents any number:

nQP is Put n lines (counting from the cursor down) into the Scratchpad
n/QP is the same as QP but append to current contents of Scratchpad
nQG is Get contents of the Scratchpad and insert into text n times
QT is Type (display) contents of the Scratchpad
QK is Kill (empty) contents of the Scratchpad
QLstring$ is load string into Scratchpad, replacing current contents
n/QLstring$ is append string to current Scratchpad contents n times

C. TEXT SEARCHING. Let's say you wanted to locate a specific portion of text in a large file. One way would be to start at the beginning of the file and page through it. But **WORDMASTER** has a special SEARCH feature. In fact, it has two different Search features. First, place the cursor at the beginning of the file and make sure you are in the Command Mode. Then type:

Fbrown$<CR>

The "F" command says Find the following string ("brown" in this case). Notice that there is a "$" (ESCAPE key) at the end of the word to be searched for. This is a "terminator" for the string. That is, the ESCAPE key is pressed to tell the computer that this is the end of the string to be searched for. Now go into the Video Mode. Notice that the cursor has been placed just after the first word "brown".

Now return to the Command Mode and type:

3Fquick$V<CR>

The cursor should now be immediately after the third "quick" following the previous cursor position. A number (n) preceding the "F" command means find the nth occurence of the string. Next type:

-2Ffox$V<CR>

The cursor is now just after the second "fox" counting backwards from the last cursor position. A negative number (-n) means search backwards from the cursor and find the nth occurence of the string.

If the "nth" match cannot be located, the cursor's position will remain unchanged, and you will be informed of this failure by the message "## Fstring". In fact, whenever you are in Command Mode and you type in a command which cannot be carried out (for whatever reason), you will see the message "## command" where command is what you just typed in.

I mentioned that there are two different types of Search commands. The kind we just worked with, "F", is called a "short search". A short search will look for a matching string within 2000 characters of the cursor or all of the text stored in the computers internal memory (RAM), whichever is greater. A short search (F) is usually used to find nearby text.

The command "N" initiates a "long search". "N" is used exactly as "F" is used. The only differences are that "N" searches forward or backward to the end of the file (depending on whether or not you use "n" or "-n") and if no match is found, the cursor will be left at the end/beginning of the file.

To sumarize:

(-)nFstring$ is Find the nth occurence of string on short search
(-)nNstring$ is find the nth occurence of string on long search

The (-) means that you can use the minus sign for a backward Search.

D. SEARCH and REPLACE. Not only can **WORDMASTER** locate words or phrases, but it can also REPLACE those words with other words or phrases.

Place the cursor at the beginning of the text and enter the Command Mode. Next type:

Sbrown$light green$<CR>

The command "S" says Search for the first string ("brown") and replace it with the second string ("light green"). Enter the Video Mode to check it.

As in the Search commands, there are both short and long searches for Search and Replace commands. "S" is the Short Search and "R" is the long search (Replace). You can check the above section (C.) for the differences between short and long searches.

The Search and Replace commands can also be used for Search and Delete. First enter Command Mode and then type:

-Slight $$V<CR>

The minus (-) sign tells **WORDMASTER** to search backwards. Notice there is nothing between the two $'s (ESCAPE's). If you replace a string ("light ") with nothing, you are in fact deleting it!

Actually,

Sstring$$
Sstring$<CR>
Sstring<CR>

will all do the same thing, that is, delete the string.

To summarize:

(-)nSold string$new string$ is short Search and replace n times
(-)nRold string$new string$ is long search and Replace n times

The (-) means that you can use the minus sign for a backward Search and Replace.

E. LOOPS. You already know that more than one command can be typed on a line before hitting RETURN. **WORDMASTER** also allows the repetition of a series of commands (also called "Command string") a specific number of times. First, change the "green" back to "brown".

Now let's take another example from the beginning of this guide. Type:

$<sbrown$blue$><CR>
BV<CR>

First we went into Command Mode. Do you recognize the Search and Replace command inside the < >'s? You may have noticed two things:

1) It's alright to use small letters for commands while in Command Mode ("s" instead of "S").

2) This command does exactly the same thing as the command:

12Sbrown$blue$<CR>

In general form, the Loop command is:

n<command string>

In all earlier commands, if the number before the command (n) was omitted, n was assumed to be equal to 1. However, in the Loop command, if n is omitted, n is assumed to be equal to 65,535. For all intents and purposes you may figure that if the n is omitted, the commands inside the < >'s will be carried out until the end of the file is reached.

Let's try a different example using Loops. Type:

B<Sdog$frog$V><CR>

The main difference in this command is that the "V" is inside the Loop. This will cause **WORDMASTER** to go into the Video Mode with the cursor positioned immediately after the word "frog" every time it replaces "dog" with "frog". Each time you are ready for it to continue, press ESCAPE and it will go into the Command Mode and carry out the next instruction. When there are no more "dog"s, the statement, "## B<Sdog$frog$V>" will appear. That's just to let you know that it cannot carry out your command.

Here's another example, type:

B<Njump$V><CR>

This time the command finds each occurence of the string "jump" and goes into Video Mode with the cursor positioned immediately after the "p". You may check to see if this was the "jump" you were interested in finding, and then edit the section as desired. When you've finished editing, just press ESCAPE and the next "jump" will be located. If there are no more "jump"s, the statement, "## B<Njump$V>" will appear.

Here is an IMPORTANT thing to remember: If you ever want to get out of a command which **WORDMASTER** is executing, just type ^C.

F. FILE INPUT and OUTPUT. Let's say you are writing a document and you want to use sections of text that you had previously saved on the disk. The next two commands will allow you to do this.

Position the cursor somewhere in the middle of the file. Then go into Command Mode and type:

YEXAMPLE.RSC<CR>

The disk drive will turn on for a few seconds. Now type:

BV<CR>

In the middle of the text (where the cursor was before), you will see the paragraph we were working on a few pages back. The command we just used is the "Y" command, it Yanks a file off the disk and inserts it into the text.

Now go back into Command Mode and type:

4WTEST<CR>

The disk turns on again, but this time we used the "W" command and Wrote the four lines after the cursor into a new file. Names of files are usually in the format "NAME.TYP" where the NAME can be up to 8 characters long and the TYP (type) is 3 characters long. NAME and TYP are separated by a period (.). However, when using the Yank or Write commands, if you leave off the period and TYP as we did above (TEST), **WORDMASTER** automatically adds ".LIB" (for LIBRARY) to the end of the file name.

Now go back into Video Mode and notice that the text is exactly the same, but the cursor is now located after the section of text that we just wrote onto the disk.

Let's check TEST.LIB. Go back to Command Mode and type:

4<YTEST.LIB>BV<CR>

You should now find the file TEST.LIB inserted into the text four times.

Let's summarize. The (d:) below means that you can optionally specify which disk the file is on:

Y(d:)name.typ$ is Yank file from disk and insert into text
nW(d:)name.typ$ is Write the next n lines onto the disk

G. EXITING **WORDMASTER** and UPDATING FILES. There are two ways to exit **WORDMASTER**. One way we have already used, the "Q" command (for Quit). The other way is the "E" command for End edit. Using it will write the current text into the current disk file (EXAMPLE2.RSC) and exit to CP/M. Whenever you use "E", the previous version of the file (if there is one) will be renamed NAME.BAK for BAcKup file as with the "H" command. Let's try it. From the Command Mode, type:

E<CR>

You are now back in CP/M, the current version of the file has been saved on the disk, and it is called EXAMPLE2.RSC.

To summarize the four disk commands:

E to End edit, update files and exit to CP/M
Q to Quit (terminate) **WORDMASTER** without writing to the disk
H to update file and continue editing
O to return to last version of file and continue editing

This completes the material to be covered in the Operators Guide. But there's a lot more to **WORD-MASTER** such as sophisticated matching on search strings, etc. For more information on the capabilities of **WORDMASTER**, please refer the Users Guide. Remember that learning how to use **WORDMASTER** is like learning how to drive a car....you need to practice in order to feel comfortable behind the wheel and after a while, you won't have to think about where the brake pedal is when you want to use it!

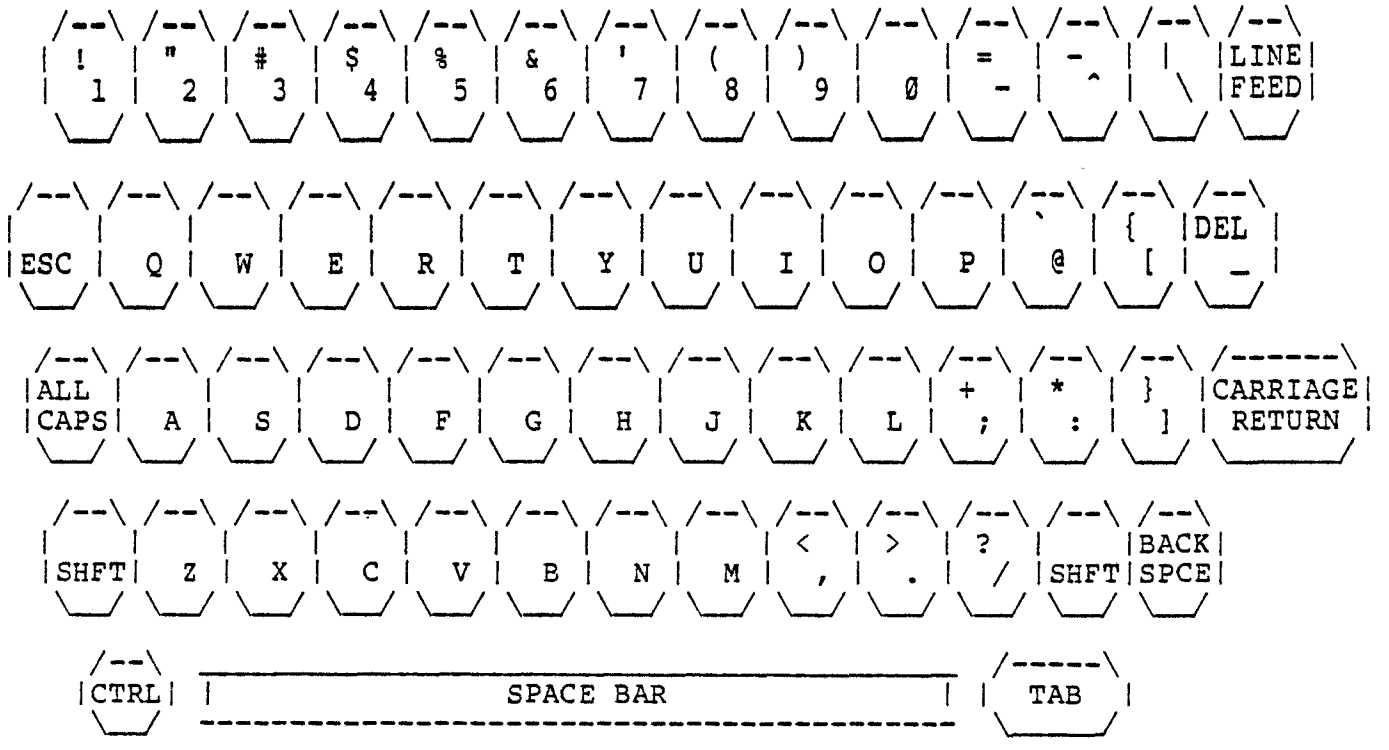
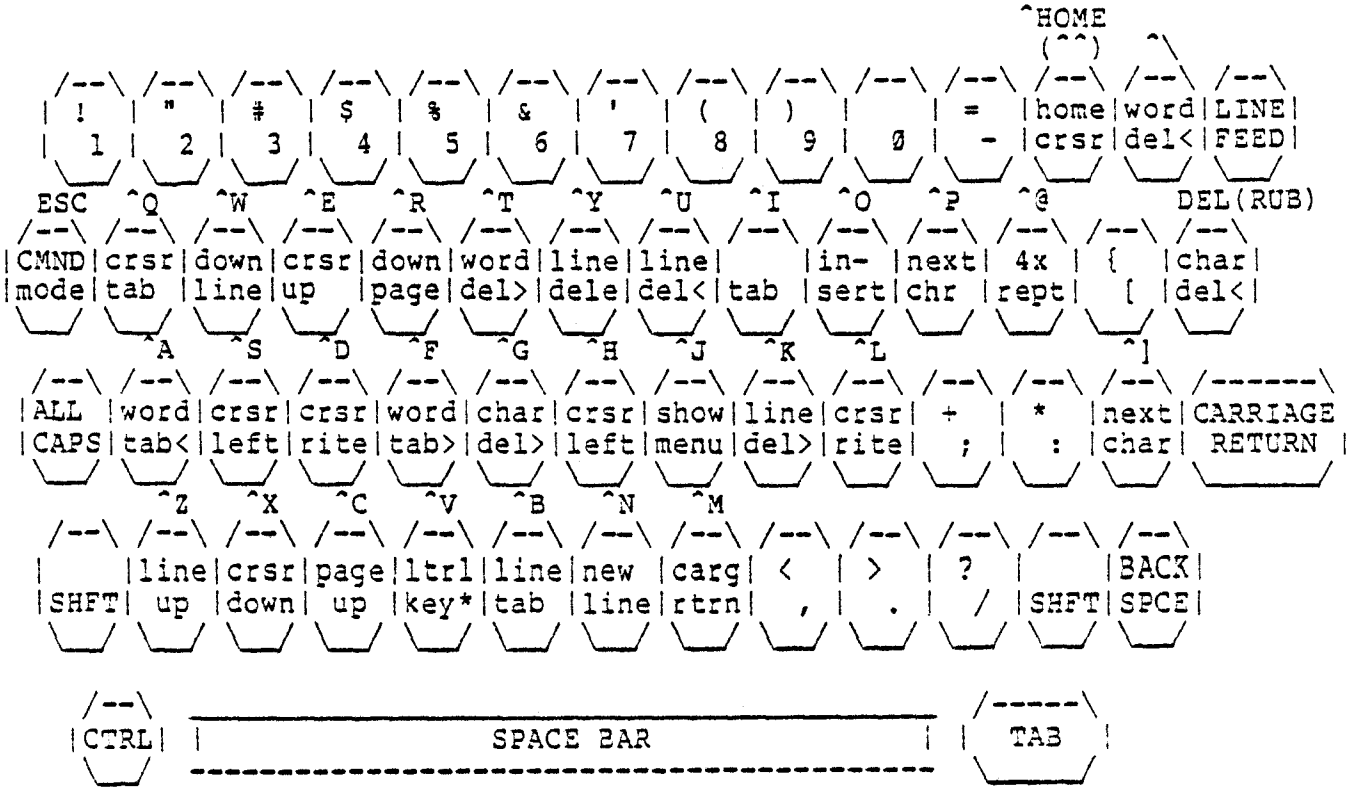


Figure 1. Typical Keyboard Layout



*See reference manual for "ltrl key" function.

Figure 2. WORD-MASTER Keyboard

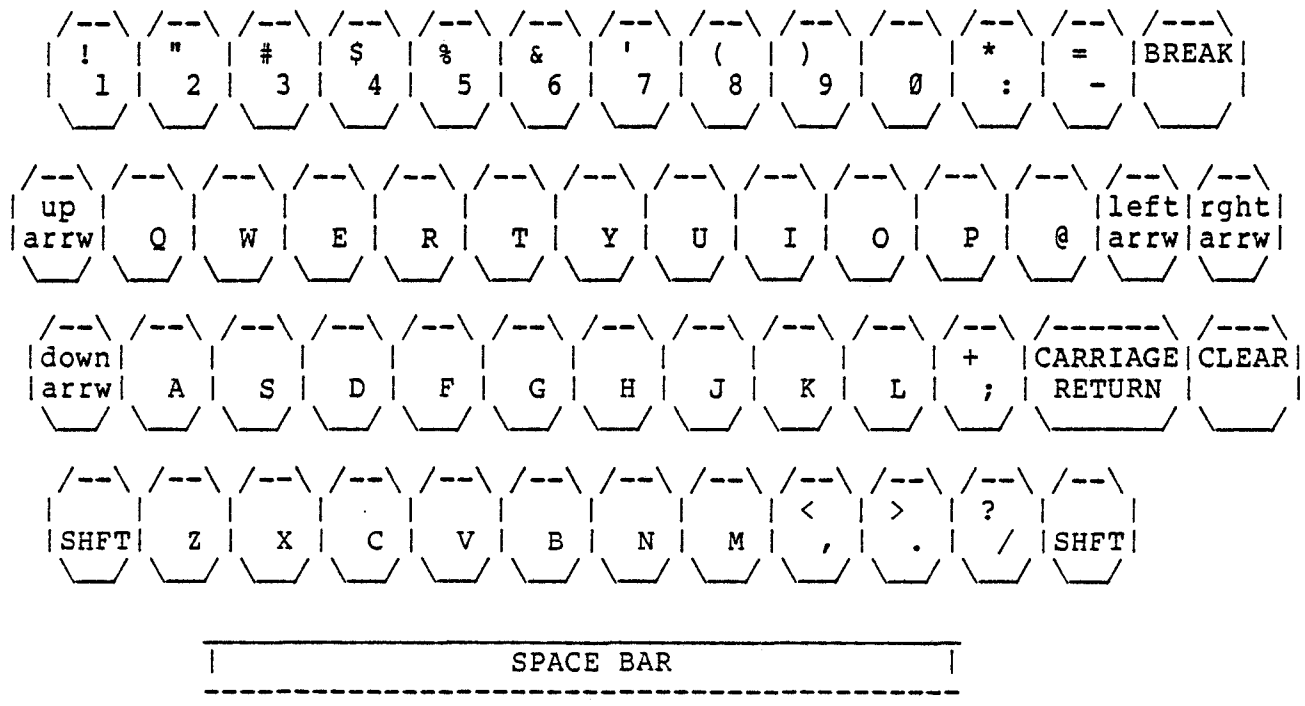
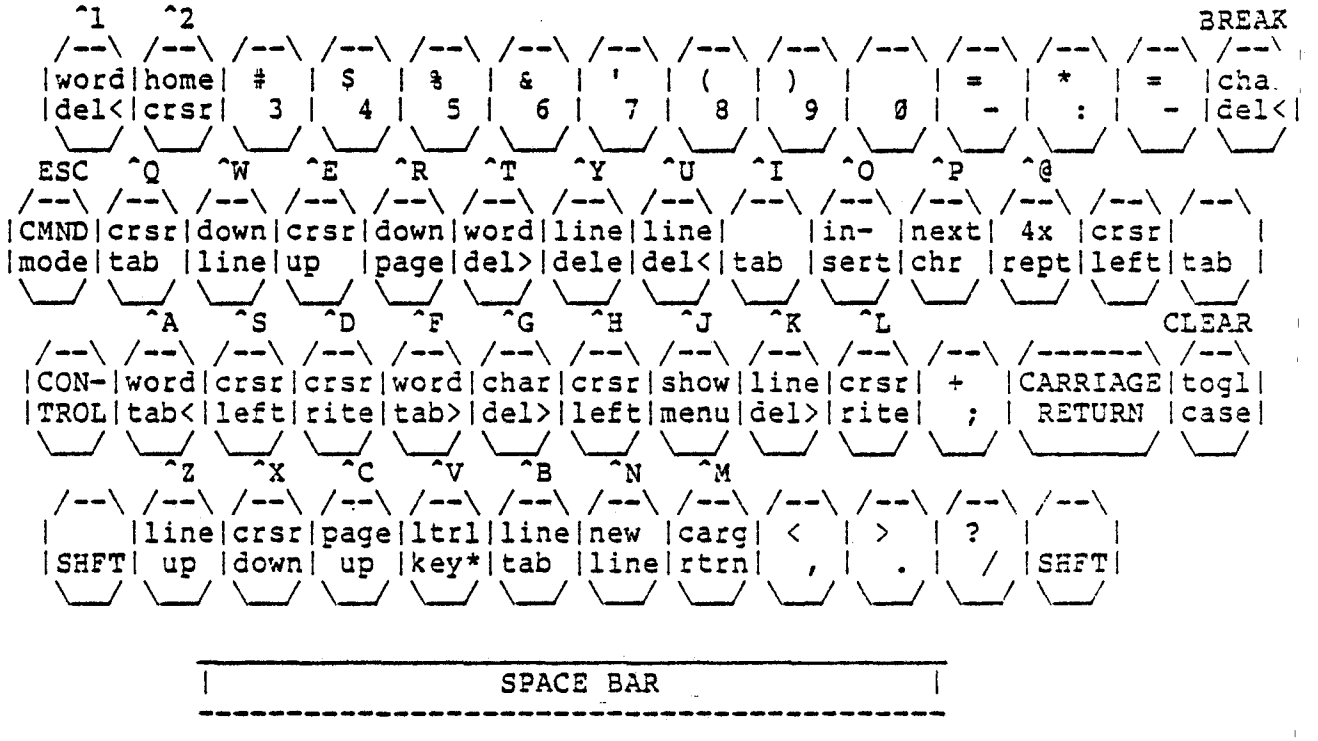


Figure 3. TRS80 Keyboard Layout



*See reference manual for "ltrl key" function.

Figure 4. **WORD-MASTER** TRS80 keyboard

GETTING STARTED

We will begin by doing some "cookbook" exercises to show off the system, dazzle your mind and quicken your curiosity for reading the rest of this guide.

Turn on your computer; slip in a diskette with **WORDMASTER** on it; and bring up the CP/M system. If you don't know how to do this, better refer to the turn-on procedure for your particular computer before proceeding further. When the A> appears, type:

A>WM EXAMPLE.RSC (and then press the RETURN button)

After a few seconds, you'll see the MicroPro copyright notice, a message that says "NEW FILE", after which the screen blanks and a white rectangular symbol, called the cursor, appears in the upper left hand corner of the screen. **WORDMASTER** is now ready for use. Use the keyboard illustrations in the back of this manual or, preferably, have your terminal ready for use.

Look at your keyboard. Look at the key to the left of the keyboard labeled CTRL. This key is the "key" to the power of **WORDMASTER.** Whenever you see an up-arrow ^ in this manual, it means that you press the CTRL key. Thus to enter ^R, you press CTRL and, without releasing the CTRL key, you press R.

Now, look at the right side of the keyboard and locate the RETURN key. Wherever you see a <CR> in this manual, it means that you press the RETURN key.

Now look at the left side of the keyboard and locate the ESCAPE key (sometimes labeled ESC). Whenever you see a dollar sign $ in this manual, it means that you press the ESCAPE key.

Let's try something to make sure we understand each other. Type the following regardless of what happens while you are typing:

$QLThe quick brown fox jumped over the lazy dog.^N<CR>
12QGBV<CR>

If you typed EXACTLY what is shown, regardless of what went on while you were typing, you should now see twelve lines of the famous nimble fox. If you don't, go back and read this section from the beginning and start over. I'm sure you'll succeed this time.

WORD-MASTER MODES

There are three different operating "modes" in **WORDMASTER.** This means that certain keys on the keyboard will cause different actions depending on the current mode you are using. **WORDMASTER**'s three modes are the Video Mode, the Command Mode and the Insert Mode.

The differences between modes may be summarized as follows:

- In Video Mode, you have the opportunity to see the appearance of the text as you make alterations. (If you are familiar with CP/M's editor, it may help you to know that the cursor position on the screen is a true representation of where the "pointer" is.) You can position the cursor in all four directions, page through text back and forth, insert and delete lines or part of lines and input or delete characters and words.

- In Command Mode, you can do everything you can in Video Mode, but any changes in the text are only displayed on your command. In addition, other facilities are possible including searching, substituting, looping, matching and moving around blocks of text. You can also combine and/or separate text from/to different files on your diskettes.

- Use Insert Mode for all original entry and for large insertions. In this mode, a backspace also erases the last character typed. You cannot backspace to a previous line.

We leave and enter the different **WORDMASTER** modes as follows:

- **WORDMASTER** always starts in Video Mode.
- To enter Command Mode from Video Mode or Insert Mode, press the ESC key (indicated by a "$")
- To enter Video Mode from Command Mode type: V<CR>
- To enter Insert Mode, first enter Command Mode and type: I<CR>

Let's perform some exercises to try out our new capabilities. Begin by typing:

```
A>WM EXAMPLE.RSC<CR>
```

What we just did was to call up the **WORDMASTER** program (WM) and tell it we want to do some work on a file called EXAMPLE.RSC. Since this file did not exist on the disk, **WORDMASTER** tells us that it is a new file. The screen will now be clear (we are in the Video Mode) and you may now enter text. Go ahead and type in the following paragraph. For the sake of this exercise, press RETURN at the end of each line of text as indicated by the "<CR>".

```
**************************
*                        *
*   E X E R C I S E S    *
*                        *
**************************
```

It is very IMPORTANT to WATCH THE SCREEN as you do these exercises so you observe what happens with each of your keystrokes!

It is also important to follow the "recipes" exactly as indicated. You will probably want to try one of the commands a few extra times. Restrain yourself if you can. But, if you can't resist and happen to make a mistake which you don't know how to correct, you can always bring the file back up from the disk by typing:

$O<CR>Y<CR> to cancel current **WORDMASTER** operations and bring it back up again with your saved file.

E. LINE INSERT and DELETE. First type:
^HOME<CR><CR> to place the cursor over the "C" of "CRT"
then: ^N^N to insert two blank lines
and: one<CR>two^E

What should have happened is that you inserted two new lines which now occupy lines three and four on the screen with the words "one" and "two", respectively. The cursor position should be just to the right of the word "one" on line three. So:

^N is line insert

Now type:

^Y^Y

The inserted lines have disappeared and the succeeding lines have moved up to fill the gap.

^Y is line delete regardless of cursor position on the line.

The cursor should now be over the "C" in "CRT". Now type:

^F^F^F^N

All the words on the line to the right of the word "or" have moved down one line and over to the left margin, pushing all succeeding lines down one line. Thus:

^N is also line insert in mid-line.

F. CHARACTER INSERT and DELETE. You can turn the character insertion toggle on and off with ^O. You know when the insertion toggle is on (i.e. active), because the the cursor is shown overlaying a < sign regardless of cursor position; the character occupying the cursor position is not visible unless you move the cursor.

Next type:

^E^E

The cursor will now be two spaces to the right of the word "or".

Now type:

^D^S^D^S

Well, you just moved the cursor over the invisible RETURN character which we inserted with a ^N earlier. Picture the RETURN as a dam which breaks up a series of words into lines. Watch what happens when we delete this dam.

Type:

^G

As you already know, ^G deletes the character it is over, in this case, a RETURN. The dam has been broken and the next line down merges with the first line.

Now try typing:

^F^X^D

The cursor is now over the space between the words "phrase" and "is". Look at your keyboard and find the key with DEL on it (RUB OUT on some keyboards). Character deletion may also occur to the left. Now press the DEL key three times and notice that the "ase" has been deleted.

To summarize:

^G is delete character under the cursor
DEL (or RUB) is delete character left

G. FULL or PARTIAL WORD and PARTIAL LINE DELETION.

First type:

^A^T^T

The cursor was first placed on the "p" of what was the word "phrase". Then "phr" was deleted when you typed the first ^T and " is" was deleted with the second ^T.

Now type:

^@^@^F

And the cursor moved ahead 16 words. It did not move it ahead 8 words. That's because the first ^@ caused the second ^@ to do its thing 4 times, or 4 times 4 equals 16.

Just to make sure, type:

^@^@^@^F

The cursor zoomed ahead 64 times (4 times 4 times 4 equals 64). However, we have now reached our limit: ^@^@^@^@ will not give us 256 times but only 4 times.

I. SCREEN SCROLLING. First, check to see how many lines of text your screen can display. It will probably be either 16 or 24 lines.

Depending on that number, type one of the following:

16 lines ^HOME^HOME^@^@<CR>This is page 2<CR>^@^@<CR>This is page 3<CR>

24 lines ^HOME^HOME^@^@<CR>^@<CR>^@<CR>This is page 2<CR>
^@^@<CR>^@<CR>^@<CR>This is page 3<CR>

First we placed the cursor at the bottom of the text. Then we used enough ^@'s with RETURNs <CR> to make the text scroll off the top of the screen:

4 X 4 = 16 for 16-line screens
4 X 4 + 4 + 4 = 24 for 24-line screens

Next we typed a phrase indicating the page number, added another 16 or 24 RETURNs and another page number.

We now have three "screenfuls" of text. Now, how do we get to see them? We use the SCROLLING commands! Let's try them - type:

^R^R

Each time you type ^R, the file of text moves DOWN one screenful, that is, we move towards the beginning of the file.

Now type:

^C^C

Each time you type ^C, the file moves UP one screenful, towards the end of the file. If you reach the end or the beginning of the file, nothing will happen (where could it go?!).

II. THE COMMAND MODE

We will now go back to those original examples and take them apart so they make sense.

A. JUMP to BEGINNING/END of TEXT.

First type:

$ (remember, that is the ESCAPE key)

When in Video Mode or Insert Mode, pressing the ESCAPE key will always put us into the Command Mode.

Next type:

B<CR>

Beginning of the file. It's there even though you can't see it. In Command Mode, there are some powerful commands, but to actually have them displayed, we must be in the Video Mode. So type:

V<CR>

Now we are in the Video Mode with the cursor over the first character of the file.

Now type:

$-BV<CR>

We are now back in the Video Mode with the cursor over the last screenful of text. Typing a "-B" will move you to the end of the file. If "B" means the Beginning of the file, then of course "-B" would mean the opposite of the Beginning, or the End.

Before we go on to the next command, lets get rid of all the garbage we now have in the file and start over again. Since we previously saved the paragraph you typed, let's get it back. Type:

$O<CR>Y<CR>

The "O" command gets us back the original version of the file (or the version of the file as it was after the last "H" command). Since this could be dangerous if typed by accident (you could lose any new corrections or updates to the text), the question "CONFIRM(Y/N)?" appeared to give you a second chance. Typing "Y"<CR> carried out the command. Typing anything else would have returned you to the Command Mode. You should now have the paragraph displayed on the screen as it was when you finished typing it.

Type:

$QT<CR>

The three lines that are still in the Scratchpad are Typed on the screen. "QT" means Type (display) the contents of the Scratchpad.

Now type:

B2QP<CR>

As before, the cursor went to the beginning of the file (even though you didn't see it) and the next 2 lines were Put into the Scratchpad.

To check it, type:

QT<CR>

Notice that the 3 lines that were in the Scratchpad before have been replaced by the two new lines.

Next type:

3/QP<CR>

The command "/QP" means the same as "QP" but this time, the new lines are added on to the end of whatever was in the Scratchpad without emptying it (appended).

Check it by typing:

QT<CR>

There are now five lines in the Scratchpad.

Type:

QKQT<CR>

Nothing happens! That's because "QK" means Kill the Scratchpad (that is, empty it out). So when you typed the "QT" command, it Typed out the now empty Scratchpad.

Do you remember those exercises we did in the beginning of this guide? Do you remember when I said that they would all be totally clear to you? We are now to the point where you can understand what happened then. First, let's start with a clean slate. Type:

Q<CR>Y<CR>

Let's summarize the Scratchpad commands. The "n" below represents any number:

nQP is Put n lines (counting from the cursor down) into the Scratchpad
n/QP is the same as QP but append to current contents of Scratchpad
nQG is Get contents of the Scratchpad and insert into text n times
QT is Type (display) contents of the Scratchpad
QK is Kill (empty) contents of the Scratchpad
QLstring$ is load string into Scratchpad, replacing current contents
n/QLstring$ is append string to current Scratchpad contents n times

C. TEXT SEARCHING. Let's say you wanted to locate a specific portion of text in a large file. One way would be to start at the beginning of the file and page through it. But **WORDMASTER** has a special SEARCH feature. In fact, it has two different Search features. First, place the cursor at the beginning of the file and make sure you are in the Command Mode. Then type:

Fbrown$<CR>

The "F" command says Find the following string ("brown" in this case). Notice that there is a "$" (ESCAPE key) at the end of the word to be searched for. This is a "terminator" for the string. That is, the ESCAPE key is pressed to tell the computer that this is the end of the string to be searched for. Now go into the Video Mode. Notice that the cursor has been placed just after the first word "brown".

Now return to the Command Mode and type:

3Fquick$V<CR>

The cursor should now be immediately after the third "quick" following the previous cursor position. A number (n) preceding the "F" command means find the nth occurence of the string. Next type:

-2Ffox$V<CR>

The cursor is now just after the second "fox" counting backwards from the last cursor position. A negative number (-n) means search backwards from the cursor and find the nth occurence of the string.

If the "nth" match cannot be located, the cursor's position will remain unchanged, and you will be informed of this failure by the message "## Fstring". In fact, whenever you are in Command Mode and you type in a command which cannot be carried out (for whatever reason), you will see the message "## command" where command is what you just typed in.

I mentioned that there are two different types of Search commands. The kind we just worked with, "F", is called a "short search". A short search will look for a matching string within 2000 characters of the cursor or all of the text stored in the computers internal memory (RAM), whichever is greater. A short search (F) is usually used to find nearby text.

To summarize:

(-)nSold string$new string$ is short Search and replace n times
(-)nRold string$new string$ is long search and Replace n times

The (-) means that you can use the minus sign for a backward Search and Replace.

E. LOOPS. You already know that more than one command can be typed on a line before hitting RETURN. **WORDMASTER** also allows the repetition of a series of commands (also called "Command string") a specific number of times. First, change the "green" back to "brown".

Now let's take another example from the beginning of this guide. Type:

$<sbrown$blue$><CR>
BV<CR>

First we went into Command Mode. Do you recognize the Search and Replace command inside the < >'s? You may have noticed two things:

1) It's alright to use small letters for commands while in Command Mode ("s" instead of "S").

2) This command does exactly the same thing as the command:

12Sbrown$blue$<CR>

In general form, the Loop command is:

n<command string>

In all earlier commands, if the number before the command (n) was omitted, n was assumed to be equal to 1. However, in the Loop command, if n is omitted, n is assumed to be equal to 65,535. For all intents and purposes you may figure that if the n is omitted, the commands inside the < >'s will be carried out until the end of the file is reached.

Let's try a different example using Loops. Type:

B<Sdog$frog$V><CR>

The main difference in this command is that the "V" is inside the Loop. This will cause **WORDMASTER** to go into the Video Mode with the cursor positioned immediately after the word "frog" every time it replaces "dog" with "frog". Each time you are ready for it to continue, press ESCAPE and it will go into the Command Mode and carry out the next instruction. When there are no more "dog"s, the statement, "## B<Sdog$frog$V>" will appear. That's just to let you know that it cannot carry out your command.

Let's check TEST.LIB. Go back to Command Mode and type:

4<YTEST.LIB>BV<CR>

You should now find the file TEST.LIB inserted into the text four times.

Let's summarize. The (d:) below means that you can optionally specify which disk the file is on:

Y(d:)name.typ$ is Yank file from disk and insert into text
nW(d:)name.typ$ is Write the next n lines onto the disk

G. EXITING **WORDMASTER** and UPDATING FILES. There are two ways to exit **WORDMASTER**. One way we have already used, the "Q" command (for Quit). The other way is the "E" command for End edit. Using it will write the current text into the current disk file (EXAMPLE2.RSC) and exit to CP/M. Whenever you use "E", the previous version of the file (if there is one) will be renamed NAME.BAK for BAcKup file as with the "H" command. Let's try it. From the Command Mode, type:

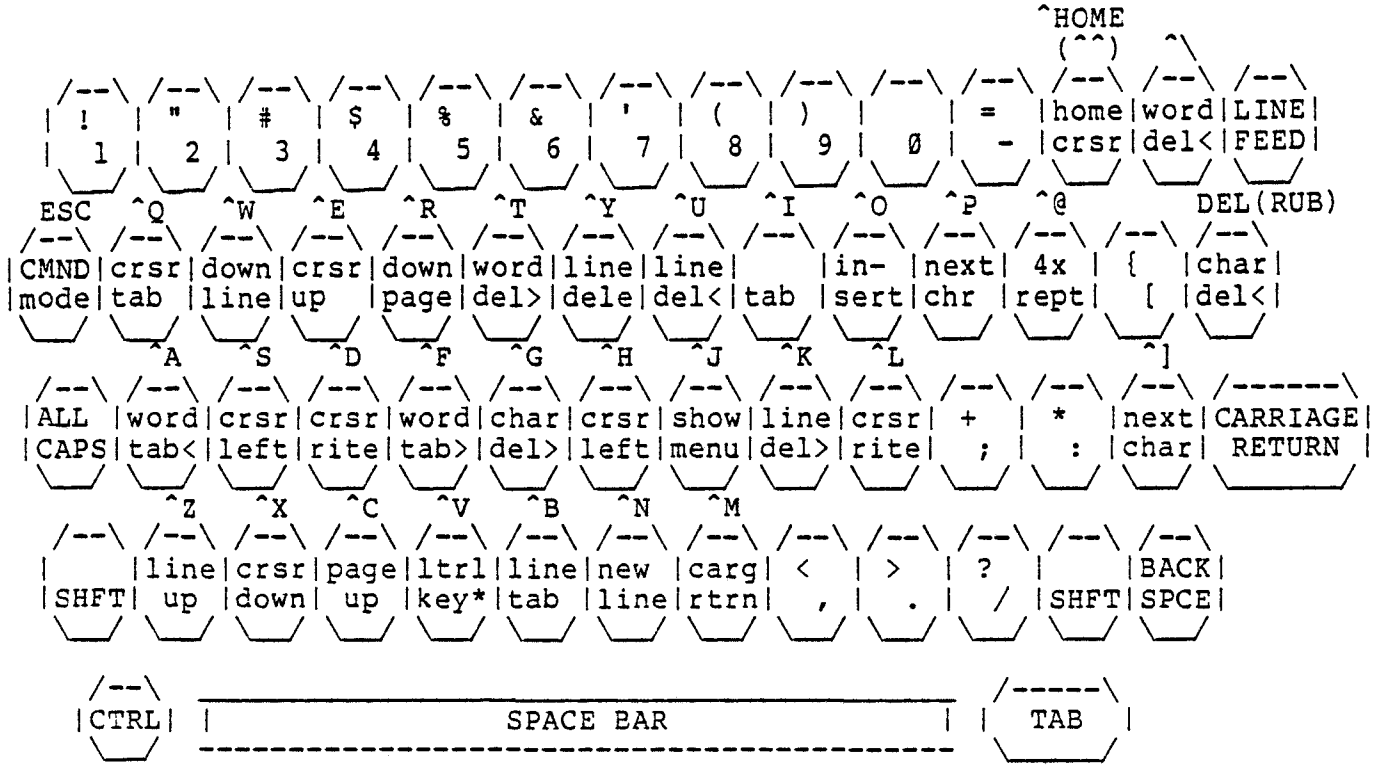
E<CR>

You are now back in CP/M, the current version of the file has been saved on the disk, and it is called EXAMPLE2.RSC.

To summarize the four disk commands:

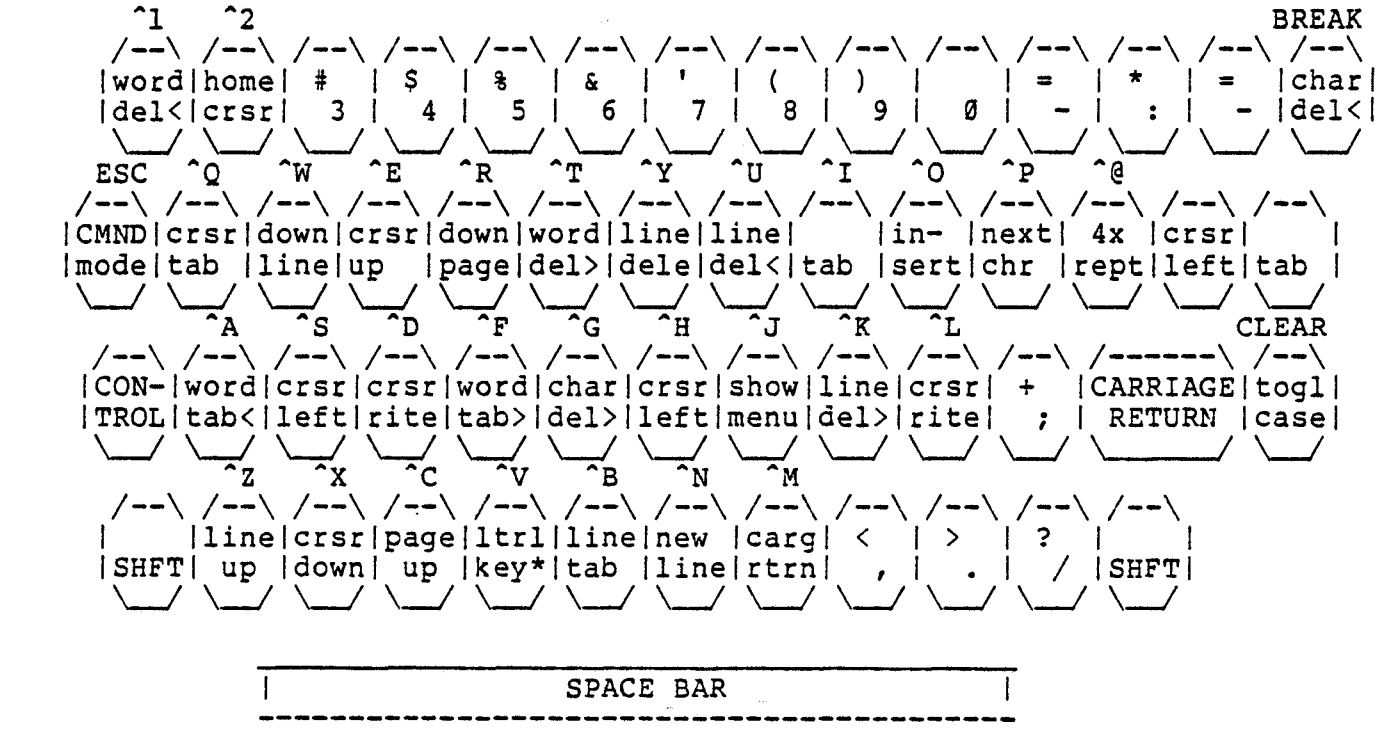
E to End edit, update files and exit to CP/M
Q to Quit (terminate) **WORDMASTER** without writing to the disk
H to update file and continue editing
O to return to last version of file and continue editing

This completes the material to be covered in the Operators Guide. But there's a lot more to **WORD-MASTER** such as sophisticated matching on search strings, etc. For more information on the capabilities of **WORDMASTER**, please refer the Users Guide. Remember that learning how to use **WORDMASTER** is like learning how to drive a car....you need to practice in order to feel comfortable behind the wheel and after a while, you won't have to think about where the brake pedal is when you want to use it!



*See reference manual for "ltrl key" function.

Figure 2. **WORD-MASTER** Keyboard



*See reference manual for "ltrl key" function.

Figure 4. **WORD-MASTER** TRS80 keyboard

WORD-MASTER asks:

CONFIRM(Y/N)?

If you respond with "Y" and a carriage return, **WORD-MASTER** execution will be terminated; any other response causes a return to command mode.

O return to original input file (or state of file as of last "H" command) and continue **WORD-MASTER.** Asks "CONFIRM" as "Q" does.

VII. STRING COMMANDS

The commands described in this section can be combined on the same command line; the entire string of commands up to the terminating carriage return is executed left to right before another command line is input.

A. Command Syntax

1. Components of commands

string represents a string of any characters except carriage return, escape, ^Z, and the command mode specials, with the following additional specials:

^N causes a carriage-return-line-feed to be in the string.

^Y causes an escape to be included in the string.

$ "string" terminator: escape (echoed as $) or ^Z.

May be omitted at end of command line.

n an integer 0-65,535. If omitted, 1 is assumed, except with "<", 65,535 is assumed. "#" means 65,535. Where permitted, specifies number of times command is executed, or number of lines or characters to operate on.

+- + or -. + assumed if neither given. Generally, "+" means toward the end of the file, and "-" means towards the beginning (backwards).

@ represents carriage return (or line feed) where it has special significance.

2. Command syntax

Generally, a command consists of a letter (or two letters) that specify the command function, followed by 0, 1, or 2 "string" arguments, depending on the command. For most commands, the command letter may be preceded by a count, n, and sometimes a sign, +-. For some commands, a / specifies a modified form of the command. The / goes between the count and the command letter. For example, the /F command is a variation of the F command.

A command line consists of one or more properly formed commands, terminated by carriage return.

B. Character Commands

Each of the following is preceded by a signed count specifying a number of characters to operate on or move past. "+" means characters after the character pointer (toward the end of file); "-" means before. The excess count is ignored if the end or the beginning of the file is encountered.

Note that a carriage-return line-feed is treated as two characters.

Commands:

+-nC move pointer n Characters.

+-nD Delete n characters.

C. Line commands

Each of the following is preceded by a signed count signifying the number of lines to operate on. A "line" is considered to be a bunch of characters terminated by a carriage return (normally) and a line feed; the cr-lf is considered part of the line it follows. A count of "1" means up to and including first line feed after the character pointer; "2", to the second, etc. "Ø" means backwards from the character pointer to, but not including, the first preceding line feed (meaning to Ø characters if the pointer is at the beginning of the line); 2 means backwards to second line feed (part of current line before pointer, plus preceding line); etc. If the count specifies more lines than are present, the command stops at the end or beginning of the file.

Commands:

+-nL move pointer n Lines

+-nK kill (delete) n lines

+-nT type (display) n lines

Several points about the T (type) command require further description:

1. Special characters in the file:
   Tab characters (entered as ^I) in the file are displayed as multiple spaces, with stops every 8 columns.
   A rubout (hex 7F) in the file displays as a "~".
   An escape character (hex 1B) in the file displays as "$".
   Other control characters display as "^" and a letter.

2. Continuation Lines: When a file line longer than the screen width less 1 is displayed, ">>" followed by the continuation of the line will be displayed on the next screen line. The ">>" signifies that there is no carriage-return line feed in the file at this point.

3. Limitation on number of characters T can display:
   A request to type more lines than **WORD-MASTER** can fit in RAM results in "{{" being printed to signify truncation at the beginning, or "}}" to signify truncation at the end. To display a large number of lines, use of the P command (Section VII.J.) or Video mode (section IX) is suggested instead of T.

D. Inserting text

Commands:

nIstring$ insert "string" in file n times at current pointer position, moving pointer to after each insertion.

nIstring@ if no escape or ^Z is given at end of command string, the cr-lf is included in the string to be inserted. (In all other commands, the final escape can be omitted, if at the end of the command line, without altering the command).

I@ enter Insert mode: text typed in is entered into file at pointer until an escape or ^Z is input. No additional prompt is printed until the terminating escape or ^Z is entered. The pointer is left after inserted text. Carriage returns typed in insert mode are stored and echoed as carriage-return line-feed. See "Special Characters in Insert Mode" (section VIII-D) below.

See also the "A" command below (section VII-J).

E. Text search and substitution

Definitions:

key	search key; same as other "string" arguments (recall ^N, ^Y), with the following additional special characters:
^A	matches any character
^S	matches a Separator character: any character but a letter or digit.
^Ox	any character Other than the following character x (but matches the sequence ^Ox if present in your file).
short search	search for a "key" within 2000 characters of the character pointer, or the number of characters that happen to be in RAM, if larger. If key not found, character pointer is unchanged.
long search	search forward or backward to end or beginning of file. If key not found, pointer will be left at the end/beginning of file.

A "short search" is recommended to find nearby text. It gives a rapid failure is the text is not found nearby, and does not move the pointer if the text is not found. A "long search" will search to the limit of the file, leaves the pointer in a different place on failure, and can take many seconds in a long file.

The amount of text searched by a "short search" increases with the amount of RAM in use, and sometimes increases as successive "short searches" are performed from the same pointer position, but it is never less than 2000 characters and the pointer is never moved if the search fails.

Commands:

+-nFkey$	does "short search" for "key" n times. Pointer is moved after found text on a forward search (+ sign), or before found text on a backward (- sign) search. If not found, an error message occurs and rest of command line is not executed.
+-nNkey$	same, but does "long search".
+-nSkey$newtext$	do "short search" for "key", delete it, and replace it with the string "newtext". Repeat the Substitution n times, each time leaving pointer after newtext (forward search) or before newtext (reverse search).

+-nRkey$newtext$
same as "S", except "long search" is used (Replace).

The above commands issue an error message and terminate processing of the command line if the key is not found. The following forms allow command processing to continue if the text is not found; they differ only in what happens if the key is not found:

+-n/Fkey$ short search and branch on failure.

+-n/Nkey$ long search and branch on failure.

+-/Skey$newtext$
short search, substitute, branch on failure.

+-/Rkey$newtext$
long search, substitution, branch on failure.

"Branch on failure" means that the innermost loop (section VII, H.) or the innermost QX command (section VII, I.) is terminated. No message is typed, the remaining commands in the current loop or in the Scratchpad are skipped, and processing continues after the loop or after the QX. If neither a loop nor a QX is in process, then execution of the command line is terminated without an error message.

Use of /F, /N, /S, and /R will be clarified and exemplified in section VII-H.

All of the above substitution commands - S, R, /S, /R - will serve to find and delete text if a null "newtext" argument is used. This includes the following forms:

Skey$$
Skey$@
Skey@

This is a difference from ED.

Note that a **WORD-MASTER** command being executed can be aborted by typing ^C. While this applies to all commands, it is particularly useful to stop a mistyped long search command from going all the way to the end of the file looking for text which is not present.

F. Miscellaneous commands

V enter Video mode

A screenful of text in the vicinity of the character pointer is displayed, with the cursor at the pointer position. If you have suitable hardware as defined in section II and Appendix A, video editing as described in section IX may now be done. The V command is a convenient way to display the contents of the file near the pointer even if video editing is not intended.

Typing an escape will cause a return to command mode. If the V was the last (or only) command in the command line, **WORD-MASTER** will print an * near the bottom of the screen and input a new command line. If additional commands followed the V, execution of the current command line will be completed before a new line is input.

Use of V's imbedded in command strings, particularly in conjunction with loops (section VII-H), is very powerful. Each time a V is executed, the operator can observe the effect of the preceding command, correct it with video edit commands if desired, then hit escape to cause command string execution to continue.

nZ "sleep" roughly n seconds.

; rest of command-line is a comment. (In Scratchpad Memory (QX command, section VII-I), command interpretation continues after next carriage return).

n! insert character whose ASCII code is n (decimal) into file at pointer. Allows insertion of arbitrary special 7-bit characters into file for unusual applications. Not needed for normal file editing; use with caution. Insertion of an end-of-file indicator (decimal 26) causes a warning message; normally, you should immediately delete a 26 with one of the following forms:

```
Skey$$
Skey$@
 Skey@
```

This is a difference from ED.

G. File I/O

Input of the file being edited is automatic, as is writing the updated file when "E" or "H" is given. The file I/O commands described here are used when it is desired to access specific additional files, to merge files, split up a file, or move or duplicate more text than can be conveniently manipulated with the Scratchpad commands.

Commands:

Y[d:]name.typ$	the entire specified file is read and inserted into the edit buffer at the current pointer position. The pointer is left after the last character of the file read. (Yank).
nW[d:]name.typ$	Write n lines forward from the character pointer onto the specified file. Leave pointer after last character written out. Previous contents of specified file are lost; there is no provision for appending to a file.
	In both of the above commands, the file type defaults to ".LIB" if no type and no "." is included in the command. A blank type many be forced by including the period.

H. Loops

Command form:

n<command string>

repeats the command string inside the < >'s times, or until an error or interruption from the console. If n is omitted, 65,535 is assumed. This is commonly used when a search failure will terminate the repetition. < >'s may be nested to a level of about 10. Matching ">"'s are assumed as necessary at the end of the command line.

Examples:

10<RABCDE0TT>@

changes the next 10 ABC's in the file to DE, and types the changed line after each replacement.

5<10<5IA$6IB$I^N$>20<RC$DEFC>>@

inserts 10 lines consisting of 5 A's, followed by 6 B's, changes the odd-numbered ones of the next 40 C's to DE's, then repeats all of the above 4 more times. Recall that I^N$ inserts a carriage-return line feed.

Nifty example:

B<Ngrumble$V>

This command finds each occurence of the string "grumble" in the file, and goes into video mode with the cursor positioned immediately after the found string. The user may inspect the context in which "grumble" occured, edit it as desired, then hit escape to leave video mode. **WORD-MASTER** will then search forward to the next occurence of "grumble" and again enter video mode. (Read about video mode in section IX - you'll find it rewarding.)

Example of use of loops together with "branch on failure" search commands (See /F, /N, /R, /S in section VII. E.):

B</RJones$Smith$>B</REdwardTom>@

move to the beginning of the file, change "Jones"'s to "Smith"'s; then, when no more "Jones"'s can be found, move again to the beginning of the file and change "Edward"'s to "Tom"'s until no more "Edward's can be found.

This could also be accomplished by giving the command line

B#RJones$Smith@,

waiting for it to complete (it will give an error message when another "Jones" cannot be found); then, give the command line

B#REdward$Tom@

The purpose of the more complicated form is to permit specifying a number of operations at once, then attend to other tasks while **WORD-MASTER** is performing them.

Note that:

B#RJones$Smith$B#REdward$Tom@

would not accomplish the desired goal, because the first R command gives an error and terminates command execution when another "Jones" cannot be found.

Use of /R instead of R means "branch out ofloop on search failure"; the < > 's delimit where to branch to.

For a long file, the example could be accomplished more quickly by moving backwards through the file for the second substitution command:

B</RJones$Smith$>-B<-/REdwardTom@

I. Scratchpad Commands

Commands:

nQP — put n lines (forward from the character pointer) of text from the file into the Scratchpad. The text is deleted from the file, and replaces any previous contents of the Scratchpad.

n/QP — same, but text is appended to current contents of Scratchpad Memory.

nQG — get Scratchpad: insert contents of Scratchpad into file at character pointer, leaving pointer after inserted text. Contents of Scratchpad not altered. Repeated n times.

QT — type contents of Scratchpad.

QK — empty (Kill) Scratchpad.

QLstring$ — load string into Scratchpad, replacing current contents.

n/QLstring$ — append string to current Scratchpad contents n times.

nQX — execute Scratchpad: execute contents of Scratchpad as a **WORD-MASTER** command string n times, then return to executing previous command string. May be nested about 6 levels.

The command string to be QX'd may be introduced into the Scratchpad with the QL command, or it may be prepared in the edit buffer (making it possible to use all WORD-MASTER facilities to enter, display, and correct the string), then moved to the Scratchpad with QP. The command string may contain carriage return line feeds and may be as long as desired up to the capacity of the buffer. Comments beginning with ; and continuing to the next carriage return may be included in the string.

When the Scratchpad is being executed, carriage returns and line feeds between commands are ignored and carriage returns and line feeds in string arguments are considered to be part of the argument. Thus, escapes to terminate arguments may not be omitted if the command string continues on another line.

Escapes may be entered by typing a ^Y to the QL or I command or in Insert mode; in video mode, type ^]$ (control- right- bracket, escape).

NOTE: the command string in the Scratchpad may append to the buffer to extend itself (/QP or /QL), but it should not delete or alter the existing contents. Doing a QP or QL or QK within the string being QX'd is a guaranteed disaster against which **WORD-MASTER** has no defenses.

Examples:

5QPQGQT@

put next 5 lines of edit buffer into Scratchpad (5QL), restore them to edit buffer (QG), type them (QT). The lines remain in the Scrgtchpad and may later be duplicated elsewhere in the file with another QG command, or executed as **WORD-MASTER** commands with QX.

5QP2ØLQG@

move 5 lines of file forward 2Ø lines.

NSUBR:$ØL1ØQP-3N^SRET^S$LQG@

find the next occurrence of "SUBR:" in the file, put 1Ø lines, including the entire line containing "SUBR:" into the Scratchpad (deleting from the file), move backwards through the file past 3 occurrences of "RET", surrounded by separator characters, move forward to the beginning of the next line, and insert the 1Ø lines there.

QLRJONES^YSMITH^YØTT@

put the string "RJONES$SMITH$ØTT" into the Scratchpad. Subsequent QX commands will cause **WORD-MASTER** to change the next "JONES" to "SMITH" and type the line in which the change was made.

J. Abbreviated and Composite Commands

The following commands are provided for convenience:

+-nP move and display screen fulls (pages). Equivalent to n<+-23L23T>, except ØP means just 23T.

+-n@ a number at the end of the command string means +-nLT, i.e. move n lines and type 1 line. Interesting cases include "+", "-", and "Ø" only.

line feed
a null line terminated by a line feed means "1L1T", that is, move the character pointer to the next line and type it. A null line terminated by carriage return does not do this; if the command line is not null (ie, contains commands), line feed will not do this.

nAstring append string - insert below current line - equivalent to n<1I^Jtext.

Example: 2ØA.br@ inserts a line consisting of .br after the current line and after each of the following 19 lines.

A@ A at end of command line is equivalent to 1LI@; i.e. move pointer to next line and enter insert mode.

M is equivalent to "<". It thus can be used as in ED.

VIII. SPECIAL CHARACTERS IN VARIOUS CONTEXTS

In the following (and throughout this manual), ^x, where x is any letter, means the character control-x, typed by holding down the CTRL key and striking the appropriate letter, and @ represents carriage return (or line feed).

A. Special Characters in Command Mode

When a command is being typed in, the following characters have the special effects described:

@	carriage return (or line feed) terminates line and and starts execution.
^U or ^X	delete entire line
^H	delete previous character and erase it on screen.
rubout	delete and echo previous character. Works with any terminal, however some versions of CP/M trap the rubout character to allow its use as a backspace. UNDER THESE CIRCUMSTANCES, USE OF THE RUBOUT KEY IN **WORD-MASTER** IS NOT RECOMMENDED.
^R	retypes line. Use to get clean copy after rubouts.
^\	(control-backslash) delete preceding word and erase it on screen. Moving leftward, erases any spaces and tabs present, then any one character, then as many alphanumeric characters as are present. Thus, "words" are deliminated by spaces or tabs, and terminated at the rigth by punctuation characters in the absence of a space or tab.
^E	echos carriage return, line feed on console without including them in command line or terminating line.
^I	tab. enters "tab" character into line; echoes as enough spaces to move cursor to next tab stop (there are tab stops every 8 columns).
^V	initiates video display control sequence. see VIII-G.
^C	deletes line and asks "ABORT?". A response of Y@ aborts the edit and returns to the CP/M or IMDOS console command processor; a response not beginning with Y causes a new command line to be input.

Note: ^P does NOT have its usual CP/M / IMDOS function of turning on and off echo of console output to the LST: device.

When a command is being executed (other than the V command; see VIII-E), there is one character which has special significance:

^C aborts the command string in progress, types a message, and returns to command input.

Other characters typed during command execution are either lost or are saved as part of the next command. In the latter case, they appear on the screen after the *. If you type very slowly, it is possible to type the next command while the current command is being executed.

B. Additional Special Characters in String Arguements

In addition to the characters described above for command input, the following apply when typed within the "string" argument(s), to the F, N, R, S, Istring$, Astring$, and QL commands and their / variants:

^N is converted to carriage-return line-feed.

^Y is converted to an escape which is included in (does not terminate) the string argument.

Recall that string arguments are terminated by an escape or control-Z (^Z), which is shown as $ in the command descriptions.

C. Yet More Special Characters in Search Key String Arguments

Within the arguments to F or N, or the first argument to R or S, the following also apply:

^A matches Any character.

^S matches any Separator (not letter or digit)

^Ox matches any character other than the following character "x" (but will match ^Ox if present in your file).

D. Special Characters in Insert Mode

When text is being inserted (after an I@ or A@ command, until the escape or ^Z that terminates the insertion), the following special character definitions apply:

@ typing carriage return enters carriage return and line feed into the file

^U or ^X erases characters back to preceding line feed (like OK), even if this includes characters that were in file before Insert Mode was entered.

^R retypes current line from preceding line feed to pointer position.

rubout deletes and echoes preceding character. If entire current insertion has been deleted, deletes character before it. To erase a carriage return line feed, use two rubouts, one for the cr, one for the lf. Works with any terminal. ^R may then be used to retype remaining portion of line.

^H deletes preceding character and erases on video terminal screen. Will not erase carriage returns or line feeds (use rubout); will not erase characters that have not been displayed on screen. That is, if pointer was not at the beginning of line when insert mode was entered, ^H will erase characters that were already in the file if and only if they have been displayed with ^R or with Øt before I@.

^\ (control-backslash) deletes preceding "word" and erases in on screen. Moving leftward, deletes as many spaces and tabs as are present, then deletes any one character except carriage return or line feed, then as many letters and digits as are present.

^I Control-I enters a tab into the file and is echoed as multiple spaces with tab stops every 8 columns.

^E echo cr-lf without putting it in file.

^Y enters an escape into the buffer, to permit building WORD-MASTER command strings to be moved into the Scratchpad and executed with QX.

^V initiate video display control sequence. see VIII-G.

Escape, ^Z terminate Insert Mode. Escape echoes as $. WORD-MASTER prints an * and accepts a new command line.

^C terminate insert mode and type "INTERRUPTED" message; characters previously typed remain in file.

other control characters
are entered into buffer, e.g. ^L inserts a form feed; ^N inserts a ^N.

E. Special Characters in Video Edit Mode

In Video Mode, control characters perform editing functions as described in section IX. Escape causes a return to command mode. Non-control characters are entered into the file. A summary of the Video Edit control characters is given at the end of this manual.

F. Upper and Lower Case

Upper and lower case may be used freely, as desired, in all modes. Command letters have the same meaning in either case. There are no provisions to facilitate editing lower case text when using an upper-case-only terminal.

G. VIO Control Sequences

WORD-MASTER has a provision to support video display hardware which is controlled by special characters output to it: if ^V is typed (in any **WORD-MASTER** mode), the next character typed is echoed directly to the console output without being processed by **WORD-MASTER**; if this character is an escape, then the next character is also echoed directly to the output.

For instance, for the IMSAI VIO video display, the following are among the useful control sequences (where $ indicates escape):

^V$L	toggle between 12 and 24 lines on the screen
^V$C	toggle between 4Ø and 8Ø columns

When an IMSAI VIO display is in use and the screen format is changed with one of the above commands, **WORD-MASTER** immediately clears the screen, readjusts its internal screen height and width parameters, and redisplays the line that was being typed (in Video edit mode, a screenful of text is again put up).

IX. VIDEO EDIT MODE

A. Introduction

Video Edit mode might better be described as "video display and edit mode", because a screenful of text from the file is on display and is immediately updated to show the effect of each key struck. The position of the display cursor shows the position of the character pointer in the file - it is always on the character position after the pointer (recall that the pointer resides between, not on, characters).

Any non-control character typed goes into the file at the cursor position, either replacing the character under the cursor or as an insertion before it. Control characters perform functions including cursor motion by character, word, or line, file motion by line or screenful, text deletion by character, word, or line, and insertion on/off.

Following sections describe the available control character functions in detail, and a summary is given at the end of the manual.

After each character is typed, **WORD-MASTER** proceeds to update the display. This may involve outputting only a single character, or it may require redisplaying the entire screen. If another key is struck before the screen update is complete, the screen update will be deferred and the key will be responded to. Thus, **WORD-MASTER** can keep up with fast typists even on a low baud rate terminal, with the file display being automatically brought current at each pause in typing.

Video mode is entered automatically when **WORD-MASTER** is invoked; it can be exited by striking the escape key, and reentered with the V command. When editing is being done in Video mode, occasional exits to command mode are useful in order to perform functions such as searches, multiple substitutions, and text moves. Also, to terminate editing and record the updated file on the disk, it is necessary to escape to command mode and issue an E command.

For correct functioning of the video edit mode, a CRT terminal or video display with certain hardware functions is required. These requirements are given in section II of this manual.

If you are using a Lear-Siegler ADM-3A terminal, be sure the internal "Clear screen" and "Cursor control" switches are on. You may also find that you wish to disconnect the beeper speaker (white 3-pin connector with 2 yellow wires at left side of board).

B. File Representation

In video editing, a carriage-return line-feed is treated as a single character for insertion, deletion, and cursor motion purposes, despite the fact that it is actually stored as two characters.

A tab character is treated as a single character, despite the fact that it displays as multiple (blank) columns. A control character, if present in the file, is displayed as two characters, ^ and a letter, but edited as one. In the unlikely event that there is a rubout in the file, it is displayed as a ~. An escape character in the file will display as $.

File lines that require more columns for display than the screen width are displayed on multiple lines, with ">>" signifying the beginning of each continuation line. The rightmost column of the screen is not used. An error message will occur if a file line is too long to display on all the screen lines available.

The display is always maintained with the beginning of a file line (as opposed to a continuation line) at the top of the screen, and the entire file line containing the cursor on the screen.

C. Replacement and Insertion of Text

1. Replacing, Inserting, and Appending Characters

Control characters introduced in this section:

^O insertion on/off

^I tab - a file character, not a control character

^P put next character in file even if control

Any non-control character typed when insertion (next paragraph) is off replaces the character at the cursor, except at the end of a line, where the character is inserted without deleting the carriage return. The cursor moves to the right as characters are typed.

Typing ^O turns on insertion (and another ^O turns it off). When insertion is on, non-control characters typed are inserted at the cursor, and the rest of the line is moved to the right and redisplayed.

When insertion is on, a "<" is displayed in the cursor. This reminder can hide a character of the file. Turning insertion off when not needed is recommended.

If the cursor is at the end of the file, characters typed are appended to the file irregardless of the state of the insertion toggle.

The tab character, typed as control-I, is the only control character which is normally taken as a file character. Tabs are inserted, replaced, or appended as other non-control characters are.

To insert a control character other than tab into the file, such as a form feed (control-L), type a ^P (control right bracket) before the character.

2. Carriage Return

When insertion is on, typing a carriage return inserts a carriage return, and leaves the cursor after it. This can be used to divide a line into two lines. The part of the screen after the cursor moves down a line.

When insertion is off, typing a carriage return moves the cursor to the beginning of the next file line without altering the file. If you wish to erase the rest of the current line before moving the cursor to the next line, type ^K.

If the cursor is at the end of the file, a carriage return is appended to the file.

3. Replacing and Inserting Lines

Control characters:

^K erase from cursor to end of line

^N insert carriage return, leave cursor before it

To replace a line, position the cursor at the beginning of it (cursor positioning commands are described in the next section). Type the new line of text. Then type a ^K to erase the rest of the old line, if it was longer than the new line.

There are two ways to insert lines. Both assume you have the cursor at the beginning of the line you wish to insert above. First, you can turn on insertion (^O) and type the new line followed by a carriage return. This shoves the following line to the right until the carriage return is typed. Second, you may type a ^N, followed by the new line. The ^N inserts a carriage return and leaves the cursor before it, thus setting up a blank line to type into. The ^N method produces a less confusing display, and works the same whether or not insertion is on.

D. Moving the Cursor

1. Where the cursor won't go

The cursor will not move right beyond the position after the last character in a line in the file. An attempt to move it farther right moves it to the left end of the next line. To add text to the right of a line, position the cursor at the end of the line, space or tab over to the desired column, then type the text.

The cursor will not go into the middle of the space occupied by a tab, or onto the second column position of a control character displayed as ^-letter. It pops across those single characters.

When moved up or down, the cursor will also move left, if necessary, to avoid landing beyond the end of a line or in the middle of a tab.

Moving the cursor "up" at the top of the screen, or "down" at the bottom, causes the screen to be redisplayed with the file moved down or up a line.

The cursor will not move down beyond the end of the file. Before any text has been inserted in a new file, it won't move anywhere at all.

2. Moving cursor single characters

Control Characters:

^S cursor left one character. if at left edge of screen, goes to right end of preceding line.

^D cursor right one character. if at right end of line, cursor goes to left end of next line.

3. Moving cursor by word

For the purposes of cursor motion and text deletion in Video mode (and also for text deletion with ^W in Command and Insert modes), a "word" is defined as:

Any number of adjacent letters and digits, followed by

zero or one character that is not a letter, digit, space, tab, carriage return, or line feed, followed by

any number of spaces, tabs, carriage returns, and line feeds.

Thus, "words" are delimited by blank characters (where "blank" is broadly defined as space, tab, carriage return, or line feed), and are terminated by punctuation characters if no blank is present. This definition is convenient when editing programs, where items of interest are often separated by special characters without surrounding spaces.

The control characters to move the cursor by word are:

^F cursor forward word: cursor moves right over the "word" it is now in, and over any "blanks" present to beginning of next "word". If necessary, the cursor will move down to the next line.

^A cursor back word: cursor moves left to the beginning of a word, going to preceding line if necessary.

4. Moving cursor by tab stop

Control character:

^Q cursor right tab stop. Will not move cursor beyond end of current line.
There are tab stops every 8 columns.

5. Moving cursor by line

Control characters:

^E cursor up line.

^X cursor down line.

carriage return
if insertion is off (^O), cursor moves to beginning of next line. (if insertion is on, a carriage return is inserted in the file.)

If necessary, the above characters move the file up or down one line on the screen. With ^E and ^X, the cursor stays at the same column position, or nearest possible column to right.

6. Moving cursor to screen limits

Control characters:

^Home (^^) cursor top/bottom of screen. One moves cursor to top left of screen; two, with no intervening characters, moves it to bottom left (or to last file line).

^B cursor left/right of line. One ^B moves cursor to left of screen on same line, two move it to right end of same line.

E. Moving the File on the Screen

1. Moving the file by line

Control characters:

^W display additonal file line at top of screen. Screen is redisplayed with one line eliminated at bottom.

^Z display additional file line at bottom of screen. A line disappears off top of screen.

The cursor remains on same character unless that character goes off the screen, in which case the cursor (and file pointer) are moved down or up a line.

2. Moving the file by screenful

Control characters:

^R move backward through the file one screenful: the first line in the new display will be the line following the last line in the old display.

^C move forward in file on screenful.

The number of file lines in a "screenful" depends on the number of lines on the screen of the terminal in use and is appropriately reduced for continuation lines.

The file motion commands can take a noticeable amount of time to process, but another character may be typed before they complete. For speed, **WORD-MASTER** will suppress its screen updating activity until it has caught up with your typing.

F. Deleting Text

1. Deleting text by character

Control characters:

rubout or ^-
: delete character before cursor. On ADM-3A, depress either "SHIFT" or "CTRL" and "RUB". Successive rubouts delete successive characters to left.

^G
: delete character at cursor. Successive ^G's delete successive characters to right.

Either rubout or ^G will delete a carriage return, thus combining two lines into one, if the cursor is positioned at the beginning or end of a line, respectively.

2. Deleting text by word

The following control characters use the definitions of "word" and "blank" given in section IX-D-3 above:

^\
: delete word left. Moving left, deletes as many "blanks" as are present, (if any), then deletes one "word". Mainly to delete word you were just typing. Note that if cursor is on first character of a "word", the PRECEDING word will be deleted.

^T
: delete word right. Deletes "blanks", then "word" at, and to the right of, cursor. If there were no blanks before the word, spaces and tabs after the word will be deleted only if there was a "blank" before the cursor - this turns out to usually do the "right thing".

The above control characters will delete carriage returns, if there is a carriage return between the cursor and the word. If the cursor is in the middle of a "word", only the word to the left or right will be deleted.

3. Deleting text by line

Control characters:

^Y deletes entire line containing cursor, including carriage return at end. Lower lines on screen move up, cursor goes to beginning of line that replaces deleted line.

^U deletes portion of line to left of cursor, if any. That is, the line you were just typing in. Does not delete carriage return.

^K deletes portion of line to right of cursor, excluding carriage return.

G. Repeating Next Character

Control character:

^@ do next character 4 times.

Two ^@'s causes the next character to be done 16 times; three, 64 times; but four ^@'s do NOT cause 256 times.

Examples:

^@^X	cursor down 4 lines.
^@^@^A^@^A^A	cursor back 21 words.
^@^@^@^C	move forward in file 64 screenfuls.
^@^@-	put 16 -'s into file
^@^N	make 4 blank lines below cursor

With some following characters, the repeated execution is visible on the screen; with others, the screen is unchanged or goes blank until the repetition is completed.

H. Miscellaneous

Escape returns to command mode. An * is printed at the bottom of the screen, and a command-line, as described in preceding sections of this manual, is accepted. The command line, or a subsequent one, may contain a "V" to return to video mode. To terminate an edit session, it is necessary to escape to command mode and give the E command.

^I tab. Types as control-I, but treated as a file character, as described under "Replacement and Insertion of Characters" above.

^N inserts carriage return into file and leaves cursor before, not after, it.

^P causes next character to be inserted in file, even if it is a control character. Replacement versus insertion, and the special effects of carriage return, work as usual.

^V initiates video display control sequence as described in section VIII-G. If an IMSAI VIO video display is in use, screen format may be changed whenever desired with ^V sequences.

X. ERROR MESSAGES

The following errors are non-fatal; **WORD-MASTER** returns to command line input:

command-line text being searched for not found by F, N, R, or S command. The search command that failed, and the rest of the current command line, is printed after the ##.

??? command line without a preceding message, indicates an unrecognized command. The command that was being processed and the following portion of the command line, is printed after the ?'s. Also printed after most specific messages shown below.

QX? command line unrecognized command during execution of Scratchpad; command line printed is remainder of Scratchpad.

DISK FULL destination diskette is full.

Recovery from DISK FULL errors: sometimes moving pointer backwards yields a DISK FULL error, yet moving it forward, or giving on E command, will work. Otherwise, it may be possible to save some of the file by deleting lines until an E command ceases to give a DISK FULL error. Also, if you know the name of a file you are willing to delete on the destination disk, you can free up most of the space the file occupies by doing a "1W" command onto it (see Section VII-G).

Prevention of DISK FULL errors: before starting an edit, make sure the destination disk contains enough space for the new file (figured after the .BAK file is erased), plus almost again this much. The "almost again as much" isn't actually necessary if you avoid backing up in the file, or if the entire file can be held in RAM.

DIRECTORY FULL destination diskette file directory is full.

Q-BUF FULL attempt to put too much text in Scratchpad. QL, QP, or QA command has been prematurely terminated. The capacity of the Scratchpad can be increased by using more RAM.

MEMORY SHORTAGE, TRY CLEARING QBUF
too little memory for certain internal manipulations. Should never occur, but if it does occur, should only occur when using the minimum amount of RAM and the Scratchpad is full. Give QX command and proceed; for future edits, relocate operating system for more RAM if available.

FILE LINE LONGER THAN ENTIRE SCREEN
attempt to video-edit a section of text that contains no line feeds in the entire capacity of the screen. Insert carriage returns with I command before video-editing.

PUTCUR ERR internal error. If reproducible, report bug.

The following are fatal errors; they occur when **WORD-MASTER** is started and are followed by an immediate exit to the CCP:

TOO LITTLE MEMORY attempt to operate **WORD-MASTER** under CP/M relocated for too little RAM (less than 2ØK). Buy more RAM if necessary, relocate CP/M with the CPM command or RELOCate IMDOS with the RELOC command, and try again.

FILE EXISTS destination disk different than source, and a file of given name already exists on destination disk. Delete or rename file and give **WORD-MASTER** command again.

NO FILE NAME The CCP command **WORD-MASTER@** was typed; no file name was given after the word **"WORD-MASTER"**. It is necessary to include the name of the file to be created or edited in the console command used to start **WORD-MASTER.**

Informational Messages:

NEW FILE typed for your information when **WORD-MASTER** is invoked, if the file named in the **WORD-MASTER** command does not already exist.

INTERRUPTED typed when a command is aborted with ^C.

TURKEY an end-of-file indicator (control-Z) was just inserted with ! (Command mode) or ^P^Z (Video mode). Unless you have some very special use in mind, you would be well advised to delete it.

XI. USING THE **WORD-MASTER** MENU

The menu file, WM.HLP, is paged onto the screen with ^J. In video mode, each ^J keystroke brings the next page of the menu onto the screen. In command mode, the first ^Q is followed by hitting RETURN. Successive application of ^J thereafter will page the rest of the menu onto the screen. You may cancel the rest of the menu display by striking RETURN. WM.HLP has been set up to display on a 64x16 screen. Feel free to edit WM.HLP to suit your needs. The page delimiters are the beginning of the file, embedded ^Q, and end of file. Thus you may modify or create your own menu as long as only one such file resides on the logged in drive and is called WM.HLP.

Appendix A:

Modification for Use of Video Edit Mode with Other CRT Terminals

As **WORD-MASTER** is supplied, Video Edit Mode works with a CRT terminal with the following characteristics and capabilities:

screen height 24 lines;

screen width 8Ø columns or greater;

position cursor on the character sequence:
escape (1B hex),
= (3D hex),
line number (Ø=top) plus 2Ø hex,
column number (Ø=left) plus 2Ø hex;

clear screen on either:
the character control-Z (1A hex),or
the character sequence
escape (1B hex,),
* (2A hex).

OR with an IMSAI VIO video display.

CRT terminals which meet the above requirements include:

Lear-Siegler ADM-3A CRT with internal "clear screen" and "cursor control" switches enabled, and

SOROC CRT terminal.

WORD-MASTER may be "patched" to make Video Edit mode work with CRT terminals or Video output boards that can be accessed as CP/M / IMDOS console devices and which have backspacing, cursor- positioning, and screen- clearing capabilities, but which differ in screen dimensions or control characters from the above specifications.
(CP/M is a trademark of Digital Research.)

The relevant variables and subroutines, and a mechanism providing for patches of indefinite length, are grouped in one area of **WORD-MASTER.** The assembly listing for this section is on the next two pages and contains comments intended to be sufficient to permit a programmer to determine and make the required patches.

WORD-MASTER also has a provision for patching in an optional "erase to end of line" control character (EREOL, at the end of the listing). If your hardware has this capability, enabling its use by putting the non-Ø character value in location EREOL will speed up Video Edit mode's screen updating.

Release 1.Ø7 has additional provisions for terminals requiring timing delays for clear-screen and cursor addressing sequences. You may need to revise the nominal settings shown in the listings. As supplied, WORD-MASTER release 1.Ø7 will support Hazeltine 15ØØ, Lear Seigler ADM-3A and 31, and the SOROC IQ-12Ø at baud rates up to 19.2 kb.

```
                ;*************************************************
                ;*                                               *
                ;*  USER-MODIFYABLE ROUTINES AND CONSTANTS FOR   *
                ;*  HARDWARE-DEPENDENT TERMINAL CHARACTERISTICS  *
                ;*  AND FUNCTIONS USED BY WORD-MASTER.           *
                ;*                        RELEASE 1.061  3/10/79 *
                ;*************************************************
                ;

                ; *** SUBROUTINE TO CLEAR SCREEN AND HOME CURSOR ***
                ;
                ; THIS VERSION WORKS FOR TWO TYPES OF CRT TERMINALS,
                ; ADM3-A AND SOROC-IQ/120, AND ONE MEMORY-MAPPED VIDEO
                ; BOARD, THE IMSAI VIO.
                ;
0180                    ORG     180H
01EF =          OUTCHR  EQU     01EFH   ;OUTPUT CHARACTER FOR RELEASE 1.04
29B8 =          MEMORY  EQU     29B8H   ;USER PATCH AREA FOR RELEASE 1.04
0180 000000     CLRSCRN: NOP! NOP! NOP  ;SPACE FOR LONGER PATCH /
0183 000000              NOP! NOP! NOP  ; MAKE ADDRS MATCH OLD VERSION
                    ;FIRST, SEND ESCAPE, *. THIS WORKS FOR SOROC
                    ;AND WILL BE ERASED BY SUBSEQUENT ^Z FOR OTHERS.
0186 3E1B               MVI A,1BH       ;(1) GET ESCAPE CHARACTER
0188 CDEF01             CALL OUTCHR     ;SEND IT TO TERMINAL
018B 3E2A               MVI A,'*'       ;(2) GET ASTERISK
018D CDEF01             CALL OUTCHR     ;SEND IT TO TERMINAL
                    ;NOW SEND CTRL-Z. THIS WORKS FOR ADM-3A, IMSAI VIO.
0190 3E1A               MVI A,1AH       ;(3) GET CONTROL-Z CHARACTER
0192 CDEF01             CALL OUTCHR     ;SEND IT AND RETURN TO CALLER
0195 C9                 RET
                   ;TO CHANGE ABOVE ROUTINE TO SEND DIFFERENT
                   ;CHARACTERS, PATCH IN DESIRED CHARACTER IN
                   ;INSTRUCTION AT (1), (2), OR (3) ABOVE.
                   ;TO SEND LESS THAN 3 CHARACTERS, PATCH IN A
                   ;"RET" AFTER LAST DESIRED "CALL OUTCHR".

                ;
                ; *** SUBROUTINE TO POSITION CURSOR AT      ***
                ; ***     LINE L (0=TOP), COLUMN H (0=LEFT) ***
               0f~ ****

                ; *** SUBROUTINE TO CLEAR SCREEN AND HOME CURSOR ***
                ;
                ; THIS VERSION WORKS FOR THE HAZELTINE 1500
                ;
0180                    ORG     180H
01EF =          OUTCHR  EQU     01EFH   ;OUTPUT CHARACTER FOR RELEASE 1.061
29B8 =          MEMORY  EQU     29B8H   ;USER PATCH AREA FOR RELEASE 1.061
0180 000000     CLRSCRN: NOP! NOP! NOP  ;SPACE FOR LONGER PATCH /
0183 000000              NOP! NOP! NOP  ; MAKE ADDRS MATCH OLD VERSION
                    ;SEND TILDE, AND 1CH. THIS WORKS FOR HAZELTINE 1500
0186 3E7E               MVI A,7EH       ;(1) GET TILDE CHARACTER
0188 CDEF01             CALL OUTCHR     ;SEND IT TO TERMINAL
018B 3E1C               MVI A,1CH       ;(2) GET CLEAR-SCREEN CODE
```

```
018D C3EF01           JMP OUTCHR      ;SEND IT TO TERMINAL AND EXIT
                    ;NOP'S FOR COMPATIBILITY.
0190 0000             NOP! NOP!
0192 000000           NOP! NOP! NOP!
0195 00               NOP

                    ;
                    ; *** SUBROUTINE TO POSITION CURSOR AT      ***
                    ; ***     LINE L (Ø=TOP), COLUMN H (Ø=LEFT) ***
                    ;
                    ;HAZELTINE 15ØØ VERSION:
                    ; SENDS TILDE, 11H, X(H), Y(L)
                    ;
0196 3E7ECDEF01 TCURSOR: MVI A,7EH! CALL OUTCHR        ;SEND TILDE
019B 3E11CDEF01         MVI A,11H! CALL OUTCHR  ;SEND CURSOR ADDRESSING COMMAND
01AØ 7CCDEF01           MOV A,H! CALL OUTCHR    ;SEND COLUMN NUMBER.
01A4 7DC3EF01           MOV A,L! JMP OUTCHR     ;SEND LINE NUMBER AND EXIT.
01A8 000000             NOP! NOP! NOP           ;FOR COMPATIBILITY.
01AB 0000               NOP! NOP                ;...

01AD 0000000000         DB Ø,Ø,Ø,Ø,Ø    ;ADDITIONAL PATCH SPACE
01B2 0000000000         DB Ø,Ø,Ø,Ø,Ø    ;...


                        ;NOTE: BACKSPACE ROUTINE THAT WAS IN
                        ;PRIOR RELEASES IS NO LONGER NEEDED.
```

```
                ; **** MODIFYABLE CONSTANTS *****

                 ;PBEGMEM POINTS TO BEGINNING OF MEMORY TO USE
                 ;FOR EDIT BUFFER AND SCRATCHPAD. IF SPACE IS NEEDED
                 ;FOR PATCHES, PUT THEM WHERE THIS POINTS AND
                 ;INCREASE THIS POINTER. REMEMBER TO USE A LARGE
                 ;ENOUGH "SAVE" COMMAND!

01B7 B829       PBEGMEM: DW MEMORY

                 ;SCREEN SIZE: TAKEN FROM THE FOLLOWING,
                 ;EXCEPT SET AUTOMATICALLY TO MATCH HARDWARE
                 ;VALUES WHEN IMSAI VIO VIDEO DISPLAY IS IN USE
                 ;(DETECTED BY PRESENCE OF THE VIO ROM AT PROPER
                 ;ADDRESS, AND CON: IOBYTE FIELD = 2 OR 3).

01B9 18         HITE:   DB 24   ;MUST BE EXACT SCREEN HEIGHT IN LINES
01BA 50         WID:    DB 80   ;MUST BE <= EXACT SCREEN WIDTH

                 ;EREOL CONTAINS THE CHARACTER(S) TO ERASE SCREEN
                 ;TO END-OF-LINE WITHOUT MOVING CURSOR, IF SUCH A
                 ;CHARACTER IS AVAILABLE IN THE TERMINAL HARDWARE.
                 ;IF 0, WILL BE SIMULATED BY SOFTWARE.
                 ;AUTOMATICALLY SET TO CTL-U WHEN VIO IS IN USE.

01BB 7E         EREOL:  DB 7EH  ;(FIRST) CHARACTER, OR 0 IF NONE ** PATCHED FOR
01BC 0F                 DB 0FH  ;SECOND CHARACTER IF TERMINAL ** PATCHED FOR HA
                                ;...REQUIRES 2-CHARACTER
                                ;...SEQUENCE, ELSE A 0.

                 ;NOVIO: IF NON-0, WILL NOT LOOK FOR IMSAI VIO.
                 ;PATCH NON-0 IF VIO PROVIOSIONS INTERFERE WITH
                 ;YOUR TERMINAL.

01BD FF         NOVIO:  DB 0FFH         ;PATCHED FOR HAZELTINE 1500
01BE 000000             DB 0,0,0        ;RESERVED FOR EXPANSION


                ;DELAYS EXECUTED AFTER VARIOUS TERMINAL FUNCTIONS,
                ;BEFORE NEXT CHARACTER IS SENT TO TERMINAL. THESE
                ;ALLOW TIME FOR TERMINAL TO RESPOND, AS REQUIRED
                ;BY SOME TERMINALS WHEN USED AT HIGH BAUD RATES.
                ;INCREASE IF YOU EXPERIENCE, FOR EXAMPLE, LOSS OF
                ;CHARACTERS AFTER CLEAR SCREEN.  EACH DELAY IS
                ;APPROX NUMBER OF MILLISECONDS ON 4MHZ PROCESSOR;
                ;DELAY IS TWICE AS LONG AS SHOWN FOR 2MHZ 8080.

01C1 19         DELCLR: DB 25   ;DELAY AFTER CLEAR SCREEN: 25+ MSEC.
01C2 0A         DELCUS: DB 10   ;DELAY AFTER POSITION CURSOR: 10+MSEC.
01C3 05         DELERE: DB 5    ;DELAY AFTER ERASE TO EOL: 5+ MSEC.

01C4 0000000000         DB 0,0,0,0,0            ;MORE
01C9 0000000000         DB 0,0,0,0,0            ;EXTRA
01CE 0000000000         DB 0,0,0,0,0            ;PATCHING
01D3 0000000000         DB 0,0,0,0,0            ;SPACE

01D8                    END
```

```
                ;*************************************************
                ;*                                               *
                ;*  USER-MODIFYABLE ROUTINES AND CONSTANTS FOR   *
                ;*  HARDWARE-DEPENDENT TERMINAL CHARACTERISTICS  *
                ;*  AND FUNCTIONS USED BY WORD-MASTER.           *
                ;*                        RELEASE 1.061  3/10/79 *
                ;*************************************************
                ;
                ;
                ; *** SUBROUTINE TO CLEAR SCREEN AND HOME CURSOR ***
                ;
                ; THIS VERSION WORKS FOR THE BEEHIVE 150 (CROMEMCO 3100)
                ;
                ;SEND ESCAPE, AND 'E'. THIS WORKS FOR BEEHIVE 150
0180                    ORG     180H
01EF =          OUTCHR  EQU     01EFH
29B8 =          MEMORY  EQU     29B8H
0180 3E1B       CLRSCRN: MVI A,1BH      ;(1) GET ESCAPE CHARACTER
0182 CDEF01              CALL OUTCHR    ;SEND IT TO TERMINAL
0185 3E45                MVI A,'E'      ;(2) GET CLEAR-SCREEN CODE 'E'
0187 CDEF01              CALL OUTCHR    ;SEND IT TO TERMINAL
018A AF                  XRA A          ;CLEAR A-REG
018B CDEF01              CALL OUTCHR    ;SEND IT TO TERMINAL
018E AF                  XRA A          ;CLEAR A-REG
018F CDEF01              CALL OUTCHR    ;SEND IT TO TERMINAL
0192 AF                  XRA A          ;CLEAR A-REG
0193 C3EF01              JMP OUTCHR     ;SEND IT TO TERMINAL AND EXIT

                ;
                ; *** SUBROUTINE TO POSITION CURSOR AT      ***
                ; ***     LINE L (0=TOP), COLUMN H (0=LEFT) ***
                ;
                ;BEEHIVE 150 VERSION:
                ; SENDS ESCAPE, 'F', Y(L)+20H, X(H)+20H
                ;
0196 3E1BCDEF01TCURSOR: MVI A,1BH! CALL OUTCHR ;SEND ESCAPE
019B 3E46CDEF01         MVI A,'F'! CALL OUTCHR  ;SEND CURSOR ADDRESSING COMAND
01A0 3E20               MVI A,20H               ;GET BIAS
01A2 85CDEF01           ADD L! CALL OUTCHR      ;SEND ROW NUMBER
01A6 3E20               MVI A,20H               ;GET BIAS
01A8 84CDEF01           ADD H! CALL OUTCHR      ;SEND COL NUMBER
01AC C9                 RET                     ;EXIT
01AD 0000000000         DB 0,0,0,0,0            ;ADDITIONAL PATCH SPACE(UNUSED F
01B2 0000000000         DB 0,0,0,0,0            ;...


                        ;NOTE: BACKSPACE ROUTINE THAT WAS IN
                        ;PRIOR RELEASES IS NO LONGER NEEDED.
```

```
                ; **** MODIFYABLE CONSTANTS *****

                 ;PBEGMEM POINTS TO BEGINNING OF MEMORY TO USE
                 ;FOR EDIT BUFFER AND SCRATCHPAD. IF SPACE IS NEEDED
                 ;FOR PATCHES, PUT THEM WHERE THIS POINTS AND
                 ;INCREASE THIS POINTER. REMEMBER TO USE A LARGE
                 ;ENOUGH "SAVE" COMMAND!

01B7 B829       PBEGMEM: DW MEMORY

                 ;SCREEN SIZE: TAKEN FROM THE FOLLOWING,
                 ;EXCEPT SET AUTOMATICALLY TO MATCH HARDWARE
                 ;VALUES WHEN IMSAI VIO VIDEO DISPLAY IS IN USE
                 ;(DETECTED BY PRESENCE OF THE VIO ROM AT PROPER
                 ;ADDRESS, AND CON: IOBYTE FIELD = 2 OR 3).

01B9 18         HITE:   DB 24   ;MUST BE EXACT SCREEN HEIGHT IN LINES
01BA 50         WID:    DB 80   ;MUST BE <= EXACT SCREEN WIDTH

                 ;EREOL CONTAINS THE CHARACTER(S) TO ERASE SCREEN
                 ;TO END-OF-LINE WITHOUT MOVING CURSOR, IF SUCH A
                 ;CHARACTER IS AVAILABLE IN THE TERMINAL HARDWARE.
                 ;IF 0, WILL BE SIMULATED BY SOFTWARE.
                 ;AUTOMATICALLY SET TO CTL-U WHEN VIO IS IN USE.

01BB 00         EREOL:  DB 0    ;(FIRST) CHARACTER, OR 0 IF NONE
01BC 00                 DB 0    ;SECOND CHARACTER IF TERMINAL
                                ;...REQUIRES 2-CHARACTER
                                ;...SEQUENCE, ELSE A 0.

                 ;NOVIO: IF NON-0, WILL NOT LOOK FOR IMSAI VIO.
                 ;PATCH NON-0 IF VIO PROVIOSIONS INTERFERE WITH
                 ;YOUR TERMINAL.

01BD FF         NOVIO:  DB 0FFH         ;PATCHED FOR BEEHIVE 150
01BE 000000             DB 0,0,0        ;RESERVED FOR EXPANSION


                ;DELAYS EXECUTED AFTER VARIOUS TERMINAL FUNCTIONS,
                ;BEFORE NEXT CHARACTER IS SENT TO TERMINAL. THESE
                ;ALLOW TIME FOR TERMINAL TO RESPOND, AS REQUIRED
                ;BY SOME TERMINALS WHEN USED AT HIGH BAUD RATES.
                ;INCREASE IF YOU EXPERIENCE, FOR EXAMPLE, LOSS OF
                ;CHARACTERS AFTER CLEAR SCREEN.  EACH DELAY IS
                ;APPROX NUMBER OF MILLISECONDS ON 4MHZ PROCESSOR;
                ;DELAY IS TWICE AS LONG AS SHOWN FOR 2MHZ 8080.

01C1 19         DELCLR: DB 25   ;DELAY AFTER CLEAR SCREEN: 25+ MSEC.
01C2 0A         DELCUS: DB 10   ;DELAY AFTER POSITION CURSOR: 10+MSEC.
01C3 05         DELERE: DB 5    ;DELAY AFTER ERASE TO EOL: 5+ MSEC.

01C4 0000000000         DB 0,0,0,0,0            ;MORE
01C9 0000000000         DB 0,0,0,0,0            ;EXTRA
01CE 0000000000         DB 0,0,0,0,0            ;PATCHING
01D3 0000000000         DB 0,0,0,0,0            ;SPACE
01D8                    END
```

```
                ;
                ;*****************************************************
                ;*                                                   *
                ;*  USER-MODIFYABLE ROUTINES AND CONSTANTS FOR       *
                ;*  HARDWARE-DEPENDENT TERMINAL CHARACTERISTICS      *
                ;*  AND FUNCTIONS USED BY WORD-MASTER.               *
                ;*                          RELEASE 1.061  3/10/79   *
                ;*****************************************************
                ;
                ; *** SUBROUTINE TO CLEAR SCREEN AND HOME CURSOR ***
                ;
                ; THIS VERSION WORKS FOR THE PTCO-SOL COMPUTER
                ;
0180                    ORG     180H
01EF =          OUTCHR  EQU     01EFH
29B8 =          MEMORY  EQU     29B8H
0180 000000     CLRSCRN: NOP! NOP! NOP  ;SPACE FOR LONGER PATCH /
0183 000000              NOP! NOP! NOP  ; MAKE ADDRS MATCH OLD VERSION
0186 3E0B                MVI A,0BH      ;(1) GET CLEAR-HOME COMMAND CODE
0188 C3EF01              JMP OUTCHR     ;SEND IT TO TERMINAL AND EXIT.
018B 0000                NOP! NOP       ;NOP'S FOR COMPATIBILITY
018D 000000              NOP! NOP! NOP  ;...
                                        ;NOP'S FOR COMPATIBILITY.
0190 0000                NOP! NOP!
0192 000000              NOP! NOP! NOP!
0195 00                  NOP

                ;
                ; *** SUBROUTINE TO POSITION CURSOR AT      ***
                ; ***     LINE L (0=TOP), COLUMN H (0=LEFT) ***
                ;
                ;PTCO SOL VERSION:
                ; SENDS ESC, 02H, Y(L), ESC, 01H, X(H)
                ;
0196 3E1BCDEF01TCURSOR: MVI A,1BH! CALL OUTCHR ;SEND ESCAPE
019B 3E02CDEF01         MVI A,02H! CALL OUTCHR  ;SEND ROW ADDRESSING COMAND
01A0 7DCDEF01           MOV A,L! CALL OUTCHR    ;SEND ROW NUMBER.
01A4 3E1BCDEF01         MVI A,1BH! CALL OUTCHR  ;SEND ESCAPE
01A9 3E01CDEF01         MVI A,01H! CALL OUTCHR  ;SEND COLUMN ADDRESSING COMND.
01AE 7CC3EF01           MOV A,H! JMP OUTCHR     ;SEND COLUMN NUMBER AND EX
01B2 0000000000         DB 0,0,0,0,0            ;ADDITIONAL PATCH SPACE...


                ;NOTE: BACKSPACE ROUTINE THAT WAS IN
                ;PRIOR RELEASES IS NO LONGER NEEDED.
                ; **** MODIFYABLE CONSTANTS *****
                ;PBEGMEM POINTS TO BEGINNING OF MEMORY TO USE
                ;FOR EDIT BUFFER AND SCRATCHPAD. IF SPACE IS NEEDED
                ;FOR PATCHES, PUT THEM WHERE THIS POINTS AND
                ;INCREASE THIS POINTER. REMEMBER TO USE A LARGE
                ;ENOUGH "SAVE" COMMAND!
```

```
01B7 B829       PBEGMEM: DW MEMORY

                 ;SCREEN SIZE: TAKEN FROM THE FOLLOWING,
                 ;EXCEPT SET AUTOMATICALLY TO MATCH HARDWARE
                 ;VALUES WHEN IMSAI VIO VIDEO DISPLAY IS IN USE
                 ;(DETECTED BY PRESENCE OF THE VIO ROM AT PROPER
                 ;ADDRESS, AND CON: IOBYTE FIELD = 2 OR 3).

01B9 10         HITE:   DB 16   ;MUST BE EXACT SCREEN HEIGHT IN LINES
01BA 40         WID:    DB 64   ;MUST BE <= EXACT SCREEN WIDTH

                 ;EREOL CONTAINS THE CHARACTER(S) TO ERASE SCREEN
                 ;TO END-OF-LINE WITHOUT MOVING CURSOR, IF SUCH A
                 ;CHARACTER IS AVAILABLE IN THE TERMINAL HARDWARE.
                 ;IF 0, WILL BE SIMULATED BY SOFTWARE.
                 ;AUTOMATICALLY SET TO CTL-U WHEN VIO IS IN USE.

01BB 00         EREOL:  DB 0    ;(FIRST) CHARACTER, OR 0 IF NONE
01BC 00                 DB 0    ;SECOND CHARACTER IF TERMINAL
                                ;...REQUIRES 2-CHARACTER
                                ;...SEQUENCE, ELSE A 0.

                 ;NOVIO: IF NON-0, WILL NOT LOOK FOR IMSAI VIO.
                 ;PATCH NON-0 IF VIO PROVIOSIONS INTERFERE WITH
                 ;YOUR TERMINAL.

01BD FF         NOVIO:  DB 0FFH         ;PATCHED FOR PTCO-SOL
01BE 000000             DB 0,0,0        ;RESERVED FOR EXPANSION


                ;DELAYS EXECUTED AFTER VARIOUS TERMINAL FUNCTIONS,
                ;BEFORE NEXT CHARACTER IS SENT TO TERMINAL. THESE
                ;ALLOW TIME FOR TERMINAL TO RESPOND, AS REQUIRED
                ;BY SOME TERMINALS WHEN USED AT HIGH BAUD RATES.
                ;INCREASE IF YOU EXPERIENCE, FOR EXAMPLE, LOSS OF
                ;CHARACTERS AFTER CLEAR SCREEN.  EACH DELAY IS
                ;APPROX NUMBER OF MILLISECONDS ON 4MHZ PROCESSOR;
                ;DELAY IS TWICE AS LONG AS SHOWN FOR 2MHZ 8080.

01C1 19         DELCLR: DB 25   ;DELAY AFTER CLEAR SCREEN: 25+ MSEC.
01C2 0A         DELCUS: DB 10   ;DELAY AFTER POSITION CURSOR: 10+MSEC.
01C3 05         DELERE: DB 5    ;DELAY AFTER ERASE TO EOL: 5+ MSEC.

01C4 0000000000         DB 0,0,0,0,0            ;MORE
01C9 0000000000         DB 0,0,0,0,0            ;EXTRA
01CE 0000000000         DB 0,0,0,0,0            ;PATCHING
01D3 0000000000         DB 0,0,0,0,0            ;SPACE

01D8                    END
```

Due to TRS8Ø Model-1 keyboard and RAM limitations, we made some minor modifications to this version of **WORD-MASTER.** The changes are outlined below:

The function of the LEFT ARROW key, normally used to backspace or delete the last character entered, has been moved to the BREAK key.

The DOWN ARROW key located directly to the left of the "A" key has been renamed CTRL key in this manual. This key is used for entering special CTRL characters in **WORD-MASTER** & is represented by the "^" character in this manual. When ever you see the "^" character, you should press and hold the DOWN-ARROW key and momentarily press the letter key. For example, to enter ^Y (which deletes the present line of text you are editing) you press CTRL(DOWN ARROW) and, without releasing the CTRL key, you momentarily press the Y key. Refer to figures 3 and 4 in the operators guide, pages 27 and 28 for additional reference information.

There are no provisions for single key entry of the ESCAPE code from the TRS8Ø keyboard. Whenever ESCAPE is expected, type ^UP ARROW. As explained in the previous paragraph, one would first press the CTRL(DOWN ARROW) and, without releasing the CTRL key, you press the UP ARROW key located directly above. Refer to figure 4, page 27 of the **WORD-MASTER** operator's guide for additional information

WORD-MASTER gives you the ability to enter both lower and UPPER case characters in the text you are editing. When you start **WORD-MASTER,** all text will be entered in lower case. The SHIFT key will cause data to be entered in upper case as long as it is pressed. To cause data to be entered in UPPER case only, you can press the CLEAR key. Thus the CLEAR key on the TRS8Ø Model-1 keyboard is interpreted as SHIFT LOCK. This key toggles the case of the data entered. If you press it again all data will be entered in lower case, etc... There is a modification to the keyboard that will allow data to be shown in the case entered, which will be explained later in this document.

There are two control keys that are not the same with the TRS80 Model-1 version of **WORD-MASTER**. They are as follows:

^1 is delete word left of cursor. This is the same as ^\ in the manual. Any time you are asked to enter ^\ in the manual you should enter ^1 on the TRS80 keyboard.

^2 is home cursor. This is the same as ^^ in the manual. Any time you are asked to enter ^^ you should enter ^2.

If you have not installed the hardware modification for lower case display, you can still determine the case of any given letter on the screen by positioning the cursor over the letter. The cursor changes shape depending on what case letter is beneath it. The cursor will be larger for upper case letters than for lower case letters.

IMPORTANT - If for any reason you have to patch and save **WORD-MASTER**, use the following CPM intrinsic command:

"SAVE 43 WMXX.COM"

Where XX represents the version, if any, you are saving. This is different than the instuctions on page 5 & 6 of the manual.

The HEATH89 version of **WORD-MASTER** has features that are not available on any other version of **WORD-MASTER**; namely, the ability to program the function keys located on the top row of keys on the keyboard. What is a programmable function key? A programmable function key allows you to assign as many as 30 characters to be printed on the display with a single depression of the function key. For example, you could program the f_1 key, so that any time you pressed it, the text "MicroPro International" would be entered as if you were to enter this text one key stroke at a time. You could also program the f_2 key to say "1299 Fourth Street, Suite 400". Then by pressing the f_1 key then pressing the f_2 key you would print the company name and address in just two keystrokes.

There are 8 programmable function keys. Only 5 of them are labeled as such (f_1-f_5) the other three are the three keys directly to the right of the f_5 key (erase, blue square & red square). The key labeled with the white square is used to program the other 8 function keys.

When **WORD-MASTER** is first executed, the function keys will have the following default values:

f_1 - DELETE: this key when pressed will have the same effect as pressing the DELETE key.

f_2 - BEGINING OF FILE: this key when pressed in video mode will put you into command mode, position the cursor at the begining of the file, then return back to video mode. This can all be done in one keystroke. (Same as $BV<CR>).

f_3 - END OF FILE: this key when pressed in video mode has the same effect as the f_2 key but positions the cursor at the end of the file. (Same as $-BV<CR>).

f_4 - SAVE FILE: this key when pressed in video mode will put you into command mode, save the file, then return back to the video mode. (Same as $H<CR>).

The other 4 function keys have no effect unless they are programmed as explained on the next page.

Programming the function keys:

The user may program the function keys for any 1 to 30 character sequence of letters, commands, control codes, etc. , at any time while running **WORD-MASTER** by doing the following steps:

1. Press the key labeled with the white square on the top row of the keyboard. This enters you into the program function key mode. The computer will respond with the following message on the bottom row of the screen.

"ENTER FUNCTION KEY # (1-8) YOU WISH TO DEFINE OR RETURN: #"

2. At this point you enter the number corresponding to the function key you wish to program. If you do not wish to program any function keys then press the RETURN key to bring you back to where you were. The computer will respond with the following message. The # sign represents the key number you just entered.

"ENTER DATA KEY #:"

3. You are now ready to enter the data to be represented by the selected function key. Up to 30 characters may be entered. Control characters will be represeneted by the symbol up-arrow "^" preceding the control character entered ("^A" will be displayed when you press the CTRL key, and while still holding it down, press the "A" key). The DELETE key will delete the last character entered. To terminate entry, press the RETURN key. The function key will now be programmed to create the data just entered whenever depressed.

   If you wish to include a RETURN character as part of the data to be programmed, you can do this by fisrt typing CTRL/P, then pressing the RETURN key. The CTRL/P will not be defined as a character but signals the program that the next character entered will be defined literally as entered. If you wish CTRL/P to be included as a character you would type CTRL/P twice in a row. The first time you enter CTRL/P, it signals to include the next character literally, which could be the CTRL/P that you entered the second time.

IMPORTANT: The installation procedure on pages 4 & 5 of the manual do not have to be followed to run WORD-MASTER on the HEATH89. **WORD-MASTER** can be run just as is. If there is any reason to have to use the CPM SAVE command, make certain that you use this format:
"SAVE 44 XX.COM"
Where XX is the file name you are saving WORD-MASTER as. Note we use 44 while the manual says to use 38.

PROGRAMMER NOTES:

1. When referencing any listings contained in this manual, such as the video patches, add 4200H to any address shown. This is because the HEATH89's base address is at 4200H as opposed to 0H, which is the standard address.

2. To set your own default values for the 8 function keys, perform the following procedure:

   Load **WORD-MASTER** using DDT with the following command: "DDT WM.COM"<CR>

   Locate the starting buffer address of the function key you wish to define from the table at the bottom of this page.

   The first byte at the address contains the number of bytes of data the function key is to define. This is a binary number and can not be of value zero. The next 30 bytes is the buffer to contain the data to be defined. You may enter the length and data by using the DDT substitute command.

   When done entering the new default function key values leave DDT by typing CTRL/C.

   Enter the following CPM command. Where XX represents the version you wish to rename the new **WORD-MASTER.**

   "SAVE 44 WMXX.COM"<CR>

TABLE OF ADDRESSES FOR THE FUNCTION KEY BUFFERS:

FUNCTION KEY #	LENGTH ADDRESS	DATA ADDRESS
1	6DE3H	6DE4H
2	6E02H	6E03H
3	6E21H	6E22H
4	6E40H	6E41H
5	6E5FH	6E60H
6	6E7EH	6E7FH
7	6E9DH	6E9EH
8	6EACH	6EADH

This Page Intentionally Left Blank.

WORD-MASTER COMMAND MODE SUMMARY

+-	means	+ or - allowed here, defaults to + if omitted
@	means	carriage return (or line feed) necessary here
$	means	escape, or ^Z, or carriage return
n	means	a number; 1 assumed if omitted; # means 65535.

+-nC	move n Characters
+-nD	Delete n characters
+-nL	move n Lines
+-nK	Kill (delete) n lines
b-nT	Type n lines
+-nP	move and type pages (23 lines each)
+-n@	move n lines, type 1 line
nItext$	Insert text n times
I@	enter Insert mode (esc to return)
A@, nAtext$	(Append) do 1L then Insert
+-nFkey$	(Find) short search for key
+-nNkey$	(Next) long search for key
+-nSkey$newtext$	short search and Substitute
+-nRkey$nextext$	(Replace) long search and substitute
/F, /N, /S, /R	Same, except if not found, exit current loop or QX and continue
n<...>	loop: repeat ... n times (default 64K)
Y[d:]name.typ$	(Yank) read file
nW[d:]name.typ$	Write n lines onto file
nQP	Put n lines into Scratchpad, delete from file
n/QP	append n lines to Scratchpad
nQG	(Get) copy Scratchpad into file n times
QT	Type Scratchpad
QK	(Kill) clear Scratchpad
QX	eXecute Scratchpad
QLtext$	(Load) put text into Scratchpad
n/QLtext$	append text$ to Scratchpad
nZ	(Sleep) wait n seconds
V	enter Video edit mode (ESC to return)
n!	put character code n in file
;	rest of line is comment
E	Save changes and End edit
H	Save changes, end edit and start over
Q	(Quit) abandon edit
O	return to Original input file and cancel current changes

"text" special characters: ^N cr-lf, ^Y esc.

"key" special characters: ^S separator, ^A any, ^Ox not x, plus those for "text".

WORD-MASTER VIDEO EDIT MODE SUMMARY

Non-control characters typed are entered in file.
Control characters perform the following functions:

CONTROL CODE	HEX CODE	FUNCTION
^O	0FH	insert toggle on/off
^S	13H	cursor left character
^D	04H	cursor right character
^A	01H	cursor left word
^F	06H	cursor right word
^Q	11H	cursor right tab stop
^X	18H	cursor down line
^E	05H	cursor up line
^^	1EH	cursor top/bottom of screen (twice for bottom)
^B	02H	cursor left/right of line (twice for right)
^W	17H	file down 1 line on screen
^Z	1AH	file up 1 line on screen
^R	12H	file down 1 screenful (toward beginning of file)
^C	03H	file up 1 screenful
^G	07H	delete character right
rubout	7FH	delete character left
^\	1CH	delete word left
^T	14H	delete word right
^U	15H	delete line left
^K	0BH	delete line right
^Y	19H	delete entire line including carraige return
^I	09H	tab - a file character
^N	14H	insert carriage return, leave cursor before it
<CR>	0DH	carriage return insertion off: cursor to beginning next line insertion on: insert in file at end of file: append to file
^@	00H	do next character 4 times
^P	10H	put next character in file even if control
^V	16H	VIO control - see VIII-G
escape	1BH	leave Video Edit mode, go into command mode

In command mode type E, carriage return to terminate edit; use V command to return to video editing; see preceding command summary for other commands.

At any given moment **WORD-MASTER** is in one of 3 modes:

Video mode — A portion of the file is displayed on the CRT screen and the display is continuously updated as changes are made in the file. All printing characters typed modify the file; control keys accomplish command functions. **WORD-MASTER** starts in this mode.

Command mode — A context editing command is being input or excuted. This mode permits searches, global replacements, text moves, etcetera, and makes available the familiar ED command functions.

Insert mode — Text typed at the console is being inserted in the file. Video mode is usually preferable for this purpose; Insert Mode is provided for use with hardcopy terminals and for compatibility with ED.

WORD-MASTER's Video edit mode provides a new level of ease and convenience in inspecting and updating text files when using a CRT terminal or video display. Control keys permit moving to any point in the file without leaving video mode. Functions are provided to position the cursor by line, word, and character. The continuously updated display speeds editing and reduces mistakes.

WORD-MASTER's context-editing commands are very close to being a superset of ED's commands. Most familiar ED commands will have a similiar effect in **WORD-MASTER**, except that A and W commands should not be used as in ED because text is input and output automatically.

II. HARDWARE AND SOFTWARE REQUIREMENTS AND INSTALLATION INSTRUCTIONS

WORD-MASTER will run on any 8Ø8Ø/8Ø85 microcomputer system with floppy disk(s), CP/M or any derivative operating system, and at least 2ØK of RAM (24K to 32K is desirable)

The operating system must be relocated for at least 2ØK of RAM and best performance will be obtained if it is relocated for all available RAM. "Operating system relocation" is accomplished with various versions of the MOVCPM command. See your system documentation for further information.

WORD-MASTER's Command and Insert modes may be used with any console device. Video Edit mode requires a CRT terminal or video display capable of random cursor addressing, cursor backspacing, and clear screen functions. The devices which are supported as **WORD-MASTER** is supplied include:

For the ONE-DISK PROCEDURE:

Insert your CP/M diskette in the drive, boot up the system and enter:

```
A>DDT@
```

Remove the CP/M diskette, insert the installation diskette and enter:

```
-IWM.COM@
-R@
-Ixxx.HEX@      where xxx=three character filename corresponding
                 to your terminal
-R@
```

Remove the installation diskette, insert the CP/M diskette and enter:

```
-GØ@
A>SAVE 38 WM.COM@
```

This completes the installation of the executable file. If you wish to use the 1.Ø7 menu facility, you must copy the file WM.HLP to the CP/M diskette as well. You can use DDT only if you sysgen a CP/M nucleus onto the installation diskette, rename the WM.HLP file to something with a file type of .COM (DON'T USE WM.COM!), bring the file into memory with DDT as shown above, swap disks again, warm start boot and SAVE the file. Before using your installed copy of **WORD-MASTER**, rename the menu file back to WM.HLP. To determine the number of 256-byte blocks to enter in the SAVE command, either note the number of records in the file shown by a STAT command, divide by 2 and round up, or, note the hi-order address printed by DDT, and convert the leftmost 2 hex digits to decimal.

III. THE FILE, THE CHARACTER POINTER, AND THE SCRATCHPAD

For the purposes of this manual, the text being edited is said to be "the FILE". The current place in the file is identified by an imaginary CHARACTER POINTER, which is always between two characters, before the first character of the file, or after the last. In video mode, the location of the character pointer is shown by a cursor on the character after it; it is invisible in text printed by context edit commands.

Another place text can be is the "Scratchpad Memory". Text is moved to and from the Scratchpad Memory by explicit commands (Section VIII) for purposes such as moving text around in the file or composing macro commands. You needn't understand the Scratchpad concept to begin to use **WORD-MASTER.**

After initializing itself and deleting any existing BAK file, **WORD-MASTER** will enter Video Edit mode. The first 2Ø or so lines of the file will be displayed on the screen, with the cursor in the upper left corner. (If the file is new, the screen will be blank; if the screen has less than 24 display lines, fewer lines will be presented.)

To learn how to use video mode, read section IX of this manual.

To enter command mode, hit the "escape" (ESC) key on your keyboard. WORD-MASTER will then prompt with an * at the bottom of the screen, signifying that it is ready to accept a command line. To learn about **WORD-MASTER** context editing commands, continue reading here.

If you are using a hardcopy terminal or a non-compatible CRT, immediately hit escape and do your editing with Command and Insert modes. NOTE: BACKSPACE will NOT operate until patching has been performed for your terminal. Refer to the installation instructions for further information.

V. **WORD-MASTER** COMMAND LINES

A **WORD-MASTER** context-editing command line consists of one or more of the commands described below, terminated by a carriage return or line feed character.

WORD-MASTER signifies that it is in Command mode and ready to accept a command line by printing an "*". To get into Command mode from video mode, type "escape". (To return to video mode, the V command, section VII-F, is used.) To interrupt a command in progress, and return to command input, type ^C (control-C; hold down the CTRL key while typing a C).

The special characters that may be used to correct typing errors, etc. while a command line is being input are described in section VIII-A. The available functions are a superset of those available in CP/M and IMDOS.

VI. NON-STRING COMMANDS

The following must be the only command in a command line:

E end edit - update files and exit to the CCP.

H end edit and start over - update and rename files, then edit the result again. Moves character pointer to beginning of file, inter-changes source and destination disks if different, and enters Video mode. Use occasionally during a long edit so that disk file is current, to minimize loss of editing already done if **WORD-MASTER** is accidentally terminated.

Q terminate **WORD-MASTER** without recording the edited text on the disk. Input file remains unchanged (or reflects only changes made before the last H command). When the Q command is given,

A. Command Syntax

1. Components of commands

string represents a string of any characters except carriage return, escape, ^Z, and the command mode specials, with the following additional specials:

^N causes a carriage-return-line-feed to be in the string.

^Y causes an escape to be included in the string.

$ "string" terminator: escape (echoed as $) or ^Z.

May be omitted at end of command line.

n an integer Ø-65,535. If omitted, 1 is assumed, except with "<", 65,535 is assumed. "#" means 65,535. Where permitted, specifies number of times command is executed, or number of lines or characters to operate on.

+- + or -. + assumed if neither given. Generally, "+" means toward the end of the file, and "-" means towards the beginning (backwards).

@ represents carriage return (or line feed) where it has special significance.

2. Command syntax

Generally, a command consists of a letter (or two letters) that specify the command function, followed by Ø, 1, or 2 "string" arguments, depending on the command. For most commands, the command letter may be preceded by a count, n, and sometimes a sign, +-. For some commands, a / specifies a modified form of the command. The / goes between the count and the command letter. For example, the /F command is a variation of the F command.

A command line consists of one or more properly formed commands, terminated by carriage return.

+-nT type (display) n lines

Several points about the T (type) command require further description:

1. Special characters in the file:
   Tab characters (entered as ^I) in the file are displayed as multiple spaces, with stops every 8 columns.
   A rubout (hex 7F) in the file displays as a "~".
   An escape character (hex 1B) in the file displays as "$".
   Other control characters display as "^" and a letter.

2. Continuation Lines: When a file line longer than the screen width less 1 is displayed, ">>" followed by the continuation of the line will be displayed on the next screen line. The ">>" signifies that there is no carriage-return line feed in the file at this point.

3. Limitation on number of characters T can display:
   A request to type more lines than **WORD-MASTER** can fit in RAM results in "{{" being printed to signify truncation at the beginning, or "}}" to signify truncation at the end. To display a large number of lines, use of the P command (Section VII.J.) or Video mode (section IX) is suggested instead of T.

D. Inserting text

Commands:

nIstring$ insert "string" in file n times at current pointer position, moving pointer to after each insertion.

nIstring@ if no escape or ^Z is given at end of command string, the cr-lf is included in the string to be inserted. (In all other commands, the final escape can be omitted, if at the end of the command line, without altering the command).

I@ enter Insert mode: text typed in is entered into file at pointer until an escape or ^Z is input. No additional prompt is printed until the terminating escape or ^Z is entered. The pointer is left after inserted text. Carriage returns typed in insert mode are stored and echoed as carriage-return line-feed. See "Special Characters in Insert Mode" (section VIII-D) below.

See also the "A" command below (section VII-J).

+-nRkey$newtext$
same as "S", except "long search" is used (Replace).

The above commands issue an error message and terminate processing of the command line if the key is not found. The following forms allow command processing to continue if the text is not found; they differ only in what happens if the key is not found:

+-n/Fkey$ short search and branch on failure.

+-n/Nkey$ long search and branch on failure.

+-/Skey$newtext$
short search, substitute, branch on failure.

+-/Rkey$newtext$
long search, substitution, branch on failure.

"Branch on failure" means that the innermost loop (section VII, H.) or the innermost QX command (section VII, I.) is terminated. No message is typed, the remaining commands in the current loop or in the Scratchpad are skipped, and processing continues after the loop or after the QX. If neither a loop nor a QX is in process, then execution of the command line is terminated without an error message.

Use of /F, /N, /S, and /R will be clarified and exemplified in section VII-H.

All of the above substitution commands - S, R, /S, /R - will serve to find and delete text if a null "newtext" argument is used. This includes the following forms:

Skey$$
Skey$@
Skey@

This is a difference from ED.

Note that a **WORD-MASTER** command being executed can be aborted by typing ^C. While this applies to all commands, it is particularly useful to stop a mistyped long search command from going all the way to the end of the file looking for text which is not present.

G. File I/O

Input of the file being edited is automatic, as is writing the updated file when "E" or "H" is given. The file I/O commands described here are used when it is desired to access specific additional files, to merge files, split up a file, or move or duplicate more text than can be conveniently manipulated with the Scratchpad commands.

Commands:

Y[d:]name.typ$ — the entire specified file is read and inserted into the edit buffer at the current pointer position. The pointer is left after the last character of the file read. (Yank).

nW[d:]name.typ$ — Write n lines forward from the character pointer onto the specified file. Leave pointer after last character written out. Previous contents of specified file are lost; there is no provision for appending to a file.

In both of the above commands, the file type defaults to ".LIB" if no type and no "." is included in the command. A blank type many be forced by including the period.

Example of use of loops together with "branch on failure" search commands (See /F, /N, /R, /S in section VII. E.):

```
B</RJones$Smith$>B</REdward$Tom$>@
```

move to the beginning of the file, change "Jones"'s to "Smith"'s; then, when no more "Jones"'s can be found, move again to the beginning of the file and change "Edward"'s to "Tom"'s until no more "Edward's can be found.

This could also be accomplished by giving the command line

```
B#RJones$Smith@,
```

waiting for it to complete (it will give an error message when another "Jones" cannot be found); then, give the command line

```
B#REdward$Tom@
```

The purpose of the more complicated form is to permit specifying a number of operations at once, then attend to other tasks while **WORD-MASTER** is performing them.

Note that:

```
B#RJones$Smith$B#REdward$Tom@
```

would not accomplish the desired goal, because the first R command gives an error and terminates command execution when another "Jones" cannot be found.

Use of /R instead of R means "branch out ofloop on search failure"; the < > 's delimit where to branch to.

For a long file, the example could be accomplished more quickly by moving backwards through the file for the second substitution command:

```
B</RJones$Smith$>-B<-/REdward$Tom$@
```

NOTE: the command string in the Scratchpad may append to the buffer to extend itself (/QP or /QL), but it should not delete or alter the existing contents. Doing a QP or QL or QK within the string being QX'd is a guaranteed disaster against which **WORD-MASTER** has no defenses.

Examples:

5QPQGQT@

put next 5 lines of edit buffer into Scratchpad (5QL), restore them to edit buffer (QG), type them (QT). The lines remain in the Scrgtchpad and may later be duplicated elsewhere in the file with another QG command, or executed as **WORD-MASTER** commands with QX.

5QP2ØLQG@

move 5 lines of file forward 2Ø lines.

NSUBR:$ØL1ØQP-3N^SRET^S$LQG@

find the next occurrence of "SUBR:" in the file, put 1Ø lines, including the entire line containing "SUBR:" into the Scratchpad (deleting from the file), move backwards through the file past 3 occurrences of "RET", surrounded by separator characters, move forward to the beginning of the next line, and insert the 1Ø lines there.

QLRJONES^YSMITH^YØTT@

put the string "RJONES$SMITH$ØTT" into the Scratchpad. Subsequent QX commands will cause **WORD-MASTER** to change the next "JONES" to "SMITH" and type the line in which the change was made.

VIII. SPECIAL CHARACTERS IN VARIOUS CONTEXTS

In the following (and throughout this manual), ^x, where x is any letter, means the character control-x, typed by holding down the CTRL key and striking the appropriate letter, and @ represents carriage return (or line feed).

A. Special Characters in Command Mode

When a command is being typed in, the following characters have the special effects described:

@	carriage return (or line feed) terminates line and and starts execution.
^U or ^X	delete entire line
^H	delete previous character and erase it on screen.
rubout	delete and echo previous character. Works with any terminal, however some versions of CP/M trap the rubout character to allow its use as a backspace. UNDER THESE CIRCUMSTANCES, USE OF THE RUBOUT KEY IN **WORD-MASTER** IS NOT RECOMMENDED.
^R	retypes line. Use to get clean copy after rubouts.
^\	(control-backslash) delete preceding word and erase it on screen. Moving leftward, erases any spaces and tabs present, then any one character, then as many alphanumeric characters as are present. Thus, "words" are deliminated by spaces or tabs, and terminated at the rigth by punctuation characters in the absence of a space or tab.
^E	echos carriage return, line feed on console without including them in command line or terminating line.
^I	tab. enters "tab" character into line; echoes as enough spaces to move cursor to next tab stop (there are tab stops every 8 columns).
^V	initiates video display control sequence. see VIII-G.
^C	deletes line and asks "ABORT?". A response of Y@ aborts the edit and returns to the CP/M or IMDOS console command processor; a response not beginning with Y causes a new command line to be input.

Note: ^P does NOT have its usual CP/M / IMDOS function of turning on and off echo of console output to the LST: device.

D. Special Characters in Insert Mode

When text is being inserted (after an I@ or A@ command, until the escape or ^Z that terminates the insertion), the following special character definitions apply:

@ typing carriage return enters carriage return and line feed into the file

^U or ^X erases characters back to preceding line feed (like OK), even if this includes characters that were in file before Insert Mode was entered.

^R retypes current line from preceding line feed to pointer position.

rubout deletes and echoes preceding character. If entire current insertion has been deleted, deletes character before it. To erase a carriage return line feed, use two rubouts, one for the cr, one for the lf. Works with any terminal. ^R may then be used to retype remaining portion of line.

^H deletes preceding character and erases on video terminal screen. Will not erase carriage returns or line feeds (use rubout); will not erase characters that have not been displayed on screen. That is, if pointer was not at the beginning of line when insert mode was entered, ^H will erase characters that were already in the file if and only if they have been displayed with ^R or with Øt before I@.

^\ (control-backslash) deletes preceding "word" and erases in on screen. Moving leftward, deletes as many spaces and tabs as are present, then deletes any one character except carriage return or line feed, then as many letters and digits as are present.

^I Control-I enters a tab into the file and is echoed as multiple spaces with tab stops every 8 columns.

^E echo cr-lf without putting it in file.

^Y enters an escape into the buffer, to permit building WORD-MASTER command strings to be moved into the Scratchpad and executed with QX.

^V initiate video display control sequence. see VIII-G.

Escape, ^Z terminate Insert Mode. Escape echoes as $. **WORD-MASTER** prints an * and accepts a new command line.

^C terminate insert mode and type "INTERRUPTED" message; characters previously typed remain in file.

other control characters
are entered into buffer, e.g. ^L inserts a form feed; ^N inserts a ^N.

IX. VIDEO EDIT MODE

A. Introduction

Video Edit mode might better be described as "video display and edit mode", because a screenful of text from the file is on display and is immediately updated to show the effect of each key struck. The position of the display cursor shows the position of the character pointer in the file - it is always on the character position after the pointer (recall that the pointer resides between, not on, characters).

Any non-control character typed goes into the file at the cursor position, either replacing the character under the cursor or as an insertion before it. Control characters perform functions including cursor motion by character, word, or line, file motion by line or screenful, text deletion by character, word, or line, and insertion on/off.

Following sections describe the available control character functions in detail, and a summary is given at the end of the manual.

After each character is typed, **WORD-MASTER** proceeds to update the display. This may involve outputting only a single character, or it may require redisplaying the entire screen. If another key is struck before the screen update is complete, the screen update will be deferred and the key will be responded to. Thus, **WORD-MASTER** can keep up with fast typists even on a low baud rate terminal, with the file display being automatically brought current at each pause in typing.

Video mode is entered automatically when **WORD-MASTER** is invoked; it can be exited by striking the escape key, and reentered with the V command. When editing is being done in Video mode, occasional exits to command mode are useful in order to perform functions such as searches, multiple substitutions, and text moves. Also, to terminate editing and record the updated file on the disk, it is necessary to escape to command mode and issue an E command.

For correct functioning of the video edit mode, a CRT terminal or video display with certain hardware functions is required. These requirements are given in section II of this manual.

If you are using a Lear-Siegler ADM-3A terminal, be sure the internal "Clear screen" and "Cursor control" switches are on. You may also find that you wish to disconnect the beeper speaker (white 3-pin connector with 2 yellow wires at left side of board).

C. Replacement and Insertion of Text

1. Replacing, Inserting, and Appending Characters

Control characters introduced in this section:

^O insertion on/off

^I tab - a file character, not a control character

^P put next character in file even if control

Any non-control character typed when insertion (next paragraph) is off replaces the character at the cursor, except at the end of a line, where the character is inserted without deleting the carriage return. The cursor moves to the right as characters are typed.

Typing ^O turns on insertion (and another ^O turns it off). When insertion is on, non-control characters typed are inserted at the cursor, and the rest of the line is moved to the right and redisplayed.

When insertion is on, a "<" is displayed in the cursor. This reminder can hide a character of the file. Turning insertion off when not needed is recommended.

If the cursor is at the end of the file, characters typed are appended to the file irregardless of the state of the insertion toggle.

The tab character, typed as control-I, is the only control character which is normally taken as a file character. Tabs are inserted, replaced, or appended as other non-control characters are.

To insert a control character other than tab into the file, such as a form feed (control-L), type a ^P (control right bracket) before the character.

2. Carriage Return

When insertion is on, typing a carriage return inserts a carriage return, and leaves the cursor after it. This can be used to divide a line into two lines. The part of the screen after the cursor moves down a line.

When insertion is off, typing a carriage return moves the cursor to the beginning of the next file line without altering the file. If you wish to erase the rest of the current line before moving the cursor to the next line, type ^K.

If the cursor is at the end of the file, a carriage return is appended to the file.

2. Moving cursor single characters

Control Characters:

^S cursor left one character. if at left edge of screen, goes to right end of preceding line.

^D cursor right one character. if at right end of line, cursor goes to left end of next line.

3. Moving cursor by word

For the purposes of cursor motion and text deletion in Video mode (and also for text deletion with ^W in Command and Insert modes), a "word" is defined as:

Any number of adjacent letters and digits, followed by

zero or one character that is not a letter, digit, space, tab, carriage return, or line feed, followed by

any number of spaces, tabs, carriage returns, and line feeds.

Thus, "words" are delimited by blank characters (where "blank" is broadly defined as space, tab, carriage return, or line feed), and are terminated by punctuation characters if no blank is present. This definition is convenient when editing programs, where items of interest are often separated by special characters without surrounding spaces.

The control characters to move the cursor by word are:

^F cursor forward word: cursor moves right over the "word" it is now in, and over any "blanks" present to beginning of next "word". If necessary, the cursor will move down to the next line.

^A cursor back word: cursor moves left to the beginning of a word, going to preceding line if necessary.

4. Moving cursor by tab stop

Control character:

^Q cursor right tab stop. Will not move cursor beyond end of current line.
There are tab stops every 8 columns.

E. Moving the File on the Screen

1. Moving the file by line

Control characters:

^W display additonal file line at top of screen. Screen is redisplayed with one line eliminated at bottom.

^Z display additional file line at bottom of screen. A line disappears off top of screen.

The cursor remains on same character unless that character goes off the screen, in which case the cursor (and file pointer) are moved down or up a line.

2. Moving the file by screenful

Control characters:

^R move backward through the file one screenful: the first line in the new display will be the line following the last line in the old display.

^C move forward in file on screenful.

The number of file lines in a "screenful" depends on the number of lines on the screen of the terminal in use and is appropriately reduced for continuation lines.

The file motion commands can take a noticeable amount of time to process, but another character may be typed before they complete. For speed, **WORD-MASTER** will suppress its screen updating activity until it has caught up with your typing.

3. Deleting text by line

Control characters:

^Y deletes entire line containing cursor, including carriage return at end. Lower lines on screen move up, cursor goes to beginning of line that replaces deleted line.

^U deletes portion of line to left of cursor, if any. That is, the line you were just typing in. Does not delete carriage return.

^K deletes portion of line to right of cursor, excluding carriage return.

G. Repeating Next Character

Control character:

^@ do next character 4 times.

Two ^@'s causes the next character to be done 16 times; three, 64 times; but four ^@'s do NOT cause 256 times.

Examples:

^@^X	cursor down 4 lines.
^@^@^A^@^A^A	cursor back 21 words.
^@^@^@^C	move forward in file 64 screenfuls.
^@^@-	put 16 -'s into file
^@^N	make 4 blank lines below cursor

With some following characters, the repeated execution is visible on the screen; with others, the screen is unchanged or goes blank until the repetition is completed.

X. ERROR MESSAGES

The following errors are non-fatal; **WORD-MASTER** returns to command line input:

command-line — text being searched for not found by F, N, R, or S command. The search command that failed, and the rest of the current command line, is printed after the ##.

??? command line — without a preceding message, indicates an unrecognized command. The command that was being processed and the following portion of the command line, is printed after the ?'s. Also printed after most specific messages shown below.

QX? command line — unrecognized command during execution of Scratchpad; command line printed is remainder of Scratchpad.

DISK FULL — destination diskette is full.

Recovery from DISK FULL errors: sometimes moving pointer backwards yields a DISK FULL error, yet moving it forward, or giving on E command, will work. Otherwise, it may be possible to save some of the file by deleting lines until an E command ceases to give a DISK FULL error. Also, if you know the name of a file you are willing to delete on the destination disk, you can free up most of the space the file occupies by doing a "1W" command onto it (see Section VII-G).

Prevention of DISK FULL errors: before starting an edit, make sure the destination disk contains enough space for the new file (figured after the .BAK file is erased), plus almost again this much. The "almost again as much" isn't actually necessary if you avoid backing up in the file, or if the entire file can be held in RAM.

DIRECTORY FULL — destination diskette file directory is full.

Q-BUF FULL — attempt to put too much text in Scratchpad. QL, QP, or QA command has been prematurely terminated. The capacity of the Scratchpad can be increased by using more RAM.

MEMORY SHORTAGE, TRY CLEARING QBUF — too little memory for certain internal manipulations. Should never occur, but if it does occur, should only occur when using the minimum amount of RAM and the Scratchpad is full. Give QK command and proceed; for future edits, relocate operating system for more RAM if available.

Appendix A:

Modification for Use of Video Edit Mode with Other CRT Terminals

As **WORD-MASTER** is supplied, Video Edit Mode works with a CRT terminal with the following characteristics and capabilities:

screen height 24 lines;

screen width 8Ø columns or greater;

position cursor on the character sequence:
escape (1B hex),
= (3D hex),
line number (Ø=top) plus 2Ø hex,
column number (Ø=left) plus 2Ø hex;

clear screen on either:
the character control-Z (1A hex),or
the character sequence
escape (1B hex,),
* (2A hex).

OR with an IMSAI VIO video display.

CRT terminals which meet the above requirements include:

Lear-Siegler ADM-3A CRT with internal "clear screen" and "cursor control" switches enabled, and

SOROC CRT terminal.

WORD-MASTER may be "patched" to make Video Edit mode work with CRT terminals or Video output boards that can be accessed as CP/M / IMDOS console devices and which have backspacing, cursor- positioning, and screen- clearing capabilities, but which differ in screen dimensions or control characters from the above specifications.
(CP/M is a trademark of Digital Research.)

The relevant variables and subroutines, and a mechanism providing for patches of indefinite length, are grouped in one area of **WORD-MASTER.** The assembly listing for this section is on the next two pages and contains comments intended to be sufficient to permit a programmer to determine and make the required patches.

WORD-MASTER also has a provision for patching in an optional "erase to end of line" control character (EREOL, at the end of the listing). If your hardware has this capability, enabling its use by putting the non-Ø character value in location EREOL will speed up Video Edit mode's screen updating.

Release 1.Ø7 has additional provisions for terminals requiring timing delays for clear-screen and cursor addressing sequences. You may need to revise the nominal settings shown in the listings. As supplied, WORD-MASTER release 1.Ø7 will support Hazeltine 15ØØ, Lear Seigler ADM-3A and 31, and the SOROC IQ-12Ø at baud rates up to 19.2 kb.

```
018D C3EF01            JMP OUTCHR      ;SEND IT TO TERMINAL AND EXIT
                     ;NOP'S FOR COMPATIBILITY.
0190 0000              NOP! NOP!
0192 000000            NOP! NOP! NOP!
0195 00                NOP

                     ;
                     ; *** SUBROUTINE TO POSITION CURSOR AT      ***
                     ; ***     LINE L (0=TOP), COLUMN H (0=LEFT) ***
                     ;
                     ;HAZELTINE 1500 VERSION:
                     ; SENDS TILDE, 11H, X(H), Y(L)
                     ;
0196 3E7ECDEF01 TCURSOR: MVI A,7EH! CALL OUTCHR        ;SEND TILDE
019B 3E11CDEF01        MVI A,11H! CALL OUTCHR  ;SEND CURSOR ADDRESSING COMAND
01A0 7CCDEF01          MOV A,H! CALL OUTCHR    ;SEND COLUMN NUMBER.
01A4 7DC3EF01          MOV A,L! JMP OUTCHR     ;SEND LINE NUMBER AND EXIT.
01A8 000000            NOP! NOP! NOP           ;FOR COMPATIBILITY.
01AB 0000              NOP! NOP                ;...

01AD 0000000000        DB 0,0,0,0,0    ;ADDITIONAL PATCH SPACE
01B2 0000000000        DB 0,0,0,0,0    ;...


                       ;NOTE: BACKSPACE ROUTINE THAT WAS IN
                       ;PRIOR RELEASES IS NO LONGER NEEDED.
```

```
                ;
                ;**************************************************
                ;*                                                *
                ;*  USER-MODIFYABLE ROUTINES AND CONSTANTS FOR    *
                ;*  HARDWARE-DEPENDENT TERMINAL CHARACTERISTICS   *
                ;*  AND FUNCTIONS USED BY WORD-MASTER.            *
                ;*                        RELEASE 1.061  3/10/79  *
                ;**************************************************
                ;
                ;
                ; *** SUBROUTINE TO CLEAR SCREEN AND HOME CURSOR ***
                ;
                ; THIS VERSION WORKS FOR THE BEEHIVE 150 (CROMEMCO 3100)
                ;
                ;SEND ESCAPE, AND 'E'. THIS WORKS FOR BEEHIVE 150
0180                    ORG     180H
01EF =          OUTCHR  EQU     01EFH
29B8 =          MEMORY  EQU     29B8H
0180 3E1B       CLRSCRN: MVI A,1BH      ;(1) GET ESCAPE CHARACTER
0182 CDEF01              CALL OUTCHR    ;SEND IT TO TERMINAL
0185 3E45                MVI A,'E'      ;(2) GET CLEAR-SCREEN CODE 'E'
0187 CDEF01              CALL OUTCHR    ;SEND IT TO TERMINAL
018A AF                  XRA A          ;CLEAR A-REG
018B CDEF01              CALL OUTCHR    ;SEND IT TO TERMINAL
018E AF                  XRA A          ;CLEAR A-REG
018F CDEF01              CALL OUTCHR    ;SEND IT TO TERMINAL
0192 AF                  XRA A          ;CLEAR A-REG
0193 C3EF01              JMP OUTCHR     ;SEND IT TO TERMINAL AND EXIT

                ;
                ; *** SUBROUTINE TO POSITION CURSOR AT      ***
                ; ***     LINE L (0=TOP), COLUMN H (0=LEFT) ***
                ;
                ;BEEHIVE 150 VERSION:
                ; SENDS ESCAPE, 'F', Y(L)+20H, X(H)+20H
                ;
0196 3E1BCDEF01TCURSOR: MVI A,1BH! CALL OUTCHR ;SEND ESCAPE
019B 3E46CDEF01         MVI A,'F'! CALL OUTCHR  ;SEND CURSOR ADDRESSING COMAND
01A0 3E20               MVI A,20H               ;GET BIAS
01A2 85CDEF01           ADD L! CALL OUTCHR      ;SEND ROW NUMBER
01A6 3E20               MVI A,20H               ;GET BIAS
01A8 84CDEF01           ADD H! CALL OUTCHR      ;SEND COL NUMBER
01AC C9                 RET                     ;EXIT
01AD 0000000000         DB 0,0,0,0,0            ;ADDITIONAL PATCH SPACE(UNUSED F
01B2 0000000000         DB 0,0,0,0,0            ;...


                        ;NOTE: BACKSPACE ROUTINE THAT WAS IN
                        ;PRIOR RELEASES IS NO LONGER NEEDED.
```

```
                ;
                ;*************************************************
                ;*                                               *
                ;*  USER-MODIFYABLE ROUTINES AND CONSTANTS FOR   *
                ;*  HARDWARE-DEPENDENT TERMINAL CHARACTERISTICS  *
                ;*  AND FUNCTIONS USED BY WORD-MASTER.           *
                ;*                        RELEASE 1.061  3/10/79 *
                ;*************************************************
                ;
                ; *** SUBROUTINE TO CLEAR SCREEN AND HOME CURSOR ***
                ;
                ; THIS VERSION WORKS FOR THE PTCO-SOL COMPUTER
                ;
0180                    ORG     180H
01EF =          OUTCHR  EQU     01EFH
29B8 =          MEMORY  EQU     29B8H
0180 000000     CLRSCRN: NOP! NOP! NOP  ;SPACE FOR LONGER PATCH /
0183 000000              NOP! NOP! NOP  ; MAKE ADDRS MATCH OLD VERSION
0186 3E0B                MVI A,0BH      ;(1) GET CLEAR-HOME COMMAND CODE
0188 C3EF01              JMP OUTCHR     ;SEND IT TO TERMINAL AND EXIT.
018B 0000                NOP! NOP       ;NOP'S FOR COMPATIBILITY
018D 000000              NOP! NOP! NOP  ;...
                                      ;NOP'S FOR COMPATIBILITY.
0190 0000                NOP! NOP!
0192 000000              NOP! NOP! NOP!
0195 00                  NOP

                ;
                ; *** SUBROUTINE TO POSITION CURSOR AT      ***
                ; ***     LINE L (0=TOP), COLUMN H (0=LEFT) ***
                ;
                ;PTCO SOL VERSION:
                ; SENDS ESC, 02H, Y(L), ESC, 01H, X(H)
                ;
0196 3E1BCDEF01TCURSOR: MVI A,1BH! CALL OUTCHR ;SEND ESCAPE
019B 3E02CDEF01         MVI A,02H! CALL OUTCHR  ;SEND ROW ADDRESSING COMAND
01A0 7DCDEF01           MOV A,L! CALL OUTCHR    ;SEND ROW NUMBER.
01A4 3E1BCDEF01         MVI A,1BH! CALL OUTCHR  ;SEND ESCAPE
01A9 3E01CDEF01         MVI A,01H! CALL OUTCHR  ;SEND COLUMN ADDRESSING COMND.
01AE 7CC3EF01           MOV A,H! JMP OUTCHR     ;SEND COLUMN NUMBER AND EXIT.
01B2 0000000000         DB 0,0,0,0,0            ;ADDITIONAL PATCH SPACE...


                ;NOTE: BACKSPACE ROUTINE THAT WAS IN
                ;PRIOR RELEASES IS NO LONGER NEEDED.
                ; **** MODIFYABLE CONSTANTS *****
                ;PBEGMEM POINTS TO BEGINNING OF MEMORY TO USE
                ;FOR EDIT BUFFER AND SCRATCHPAD. IF SPACE IS NEEDED
                ;FOR PATCHES, PUT THEM WHERE THIS POINTS AND
                ;INCREASE THIS POINTER. REMEMBER TO USE A LARGE
                ;ENOUGH "SAVE" COMMAND!
```

There are two control keys that are not the same with the TRS8Ø Model-1 version of **WORD-MASTER.** They are as follows:

^1 is delete word left of cursor. This is the same as ^\ in the manual. Any time you are asked to enter ^\ in the manual you should enter ^1 on the TRS8Ø keyboard.

^2 is home cursor. This is the same as ^^ in the manual. Any time you are asked to enter ^^ you should enter ^2.

If you have not installed the hardware modification for lower case display, you can still determine the case of any given letter on the screen by positioning the cursor over the letter. The cursor changes shape depending on what case letter is beneath it. The cursor will be larger for upper case letters than for lower case letters.

IMPORTANT - If for any reason you have to patch and save **WORD-MASTER,** use the following CPM intrinsic command:

"SAVE 43 WMXX.COM"

Where XX represents the version, if any, you are saving. This is different than the instuctions on page 5 & 6 of the manual.

Programming the function keys:

The user may program the function keys for any 1 to 3Ø character sequence of letters, commands, control codes, etc. , at any time while running **WORD-MASTER** by doing the following steps:

1. Press the key labeled with the white square on the top row of the keyboard. This enters you into the program function key mode. The computer will respond with the following message on the bottom row of the screen.

"ENTER FUNCTION KEY # (1-8) YOU WISH TO DEFINE OR RETURN: #"

2. At this point you enter the number corresponding to the function key you wish to program. If you do not wish to program any function keys then press the RETURN key to bring you back to where you were. The computer will respond with the following message. The # sign represents the key number you just entered.

"ENTER DATA KEY #:"

3. You are now ready to enter the data to be represented by the selected function key. Up to 3Ø characters may be entered. Control characters will be represeneted by the symbol up-arrow "^" preceding the control character entered ("^A" will be displayed when you press the CTRL key, and while still holding it down, press the "A" key). The DELETE key will delete the last character entered. To terminate entry, press the RETURN key. The function key will now be programmed to create the data just entered whenever depressed.

   If you wish to include a RETURN character as part of the data to be programmed, you can do this by fisrt typing CTRL/P, then pressing the RETURN key. The CTRL/P will not be defined as a character but signals the program that the next character entered will be defined literally as entered. If you wish CTRL/P to be included as a character you would type CTRL/P twice in a row. The first time you enter CTRL/P, it signals to include the next character literally, which could be the CTRL/P that you entered the second time.

IMPORTANT: The installation procedure on pages 4 & 5 of the manual do not have to be followed to run **WORD-MASTER** on the HEATH89. **WORD-MASTER** can be run just as is. If there is any reason to have to use the CPM SAVE command, make certain that you use this format: "SAVE 44 XX.COM" Where XX is the file name you are saving **WORD-MASTER** as. Note we use 44 while the manual says to use 38.

This Page Intentionally Left Blank.

TABLE OF CONTENTS