Telewriter-64™

the Color Computer Word Processor

PRODUCED UNDER LICENCE BY

COMPUTER HUT SOFTWARE

21 WILLIAMS ST., BOWEN Q. 4805 - Ph. (077) 862220

Telewriter-64 Addenda, Errata, Etc. PLEASE READ THIS FIRST

1) The POKE addresses given in the manual to change the screen color, are off by 3 (see page 33 in the Reference Manual section of the User's guide, or page 2 in the Upgrade Manual). They should be:

Cassette	16/32K	POKE12525,240
Cassette	64K	POKE61125,240
Disk	16/32K	POKE12732,240
Disk	64K	POKE61262,240

2) On some machines, Telewriter-64 will not load properly, or it will load, but strange things may happen during use. The machines on which such problems have been seen are:

a) some 32K Color computers purchased after October 1982 (the so-called "F" board machines)

b) 16K "F" board machines which have been upgraded to 32K via the "piggy-back" method (there is an easier, cleaner way to do this upgrade).

c) Machines upgraded to 64K in which the mod has not been done properly, or one or more of the RAM chips are not 100%.

In all cases, what is happening is that Telewriter-64 is being tricked into seeing a 64K machine, when, in fact, it is not. The solution is this: Restart the machine, then:

A) For cassette based systems:

1) Do a CLOAD as usual (TELE64 loads from doing this).

2) When the screen says OK (i.e TELE64 is loaded), type:

```
19 GOTO 100
```

(followed by hitting ENTER at the end).

3) Then type RUN

B) For disk based systems:

1) LOAD "U (instead of RUNning it).

2) then, when the screen says OK, type:

```
19 GOTO 100
```

(followed by hitting ENTER at the end).

3) Then type RUN

This fix will cause Telewriter to omit the test for 64K and assume the machine is 16K or 32K.

3) If you write for technical support, please include the serial number on your Telewriter-64 cassette or disk, in your letter. If you call, please state your serial number before asking your question. We will not provide support to people without serial numbers or with invalid serial numbers.

4) To upgrade from Telewriter-64 cassette to Telewriter-64 Disk, simply send us your cassette serial # and $15. Keep your manual and keep your cassette. The manual contains all you need to know about the disk version. But once you receive the disk, you must return the cassette to us, in order to be eligible for future support.

If, however, you wish to keep the cassette, for your own use or as a collector's item, include an additional $5 (a total of $20).

5) To read BASIC programs into Telewriter-64 using the ASCII I/O option, the BASIC program must be saved using the "A" option. That is, if the filename is "PROG", save it with:

CSAVE "PROG",A (cassette) or SAVE "PROG",A (disk)

6) If you have problems loading the Telewriter tape, and #2 above does not help, there is another copy of Telewriter on side 2 of the tape, so try that. If neither side loads and you do not have a standard Radio Shack recorder (either CCR-81 or CTR-80A), that may be the problem. If possible, borrow one of these recorders (or see if you can try one at a Radio Shack store) and try loading the tape using that. If all this fails, return the tape to us and we will replace it. It is very unlikely that both sides of the tape will be bad, but, occasionally this does happen (especially when the damage occurs in transit).

TELEWRITER-64

User's Manual

TABLE OF CONTENTS

TELEWRITER-64 TUTORIAL

0 INTRODUCTION

This tutorial is intended to allow the first-time user of Telewriter-64 to begin writing with it immediately. No knowledge of programming or word processing is assumed, but some familiarity with the TRS-80 Color Computer is helpful -- Chapters 1 and 8 of "Getting Started With Color Basic" may be enough.

As with any tool, there are ways to use Telewriter-64 so as to optimize its effectiveness, and these often depend on having some understanding of how the system works. The tutorial will get into some of this, but for a more precise, ordered overview, a reference manual is provided. The tutorial will guide you quickly through doing the most important functions. The Reference Manual will answer "what-if", and "why" questions that come up when using the system. Generally, it won't mean much until you've used Telewriter-64.

If you have some knowledge of word processing or programming already, use the tutorial to jump into the system, and read appropriate sections of the reference manual at the same time. Both are ordered pretty much the same so that easy cross-reference is possible and so that the Reference Manual may be used as a tutorial as well.

In what follows, then, simply do exactly what the tutorial says. This will allow you to start using the system immediately, and quickly learn its basics.

1 RUNNING TELEWRITER-64: CASSETTE

Place the Telewriter-64 cassette in your recorder and rewind it. Reset the counter to 000. Press the Play button and type CLOAD. The program will be found (at about 006 on the counter) and loaded into memory. The top of the screen should say F TELE64 while the tape is loading. When this is done and the screen says "OK", type RUN and hit ENTER. Leave the recorder's Play button on. The copyright notice will appear, and the recorder will start again, reading in the rest of Telewriter-64.

When the recorder stops, you will be looking at the Main menu. The words in reverse video (white on black) are the available commands.

2 RUNNING TELEWRITER-64: DISK

The Telewriter-64 Disk is <u>NOT</u> copy protected. This is for your convenience in making backup copies <u>for your own use</u>. The first thing you should do is make a backup copy of the Telewriter-64 disk and put the original away for safe keeping.

To run Telewriter-64, place the Telewriter disk in drive 0 and type RUN "U. After you hit ENTER, the copyright notice will appear and the disk will thunk a few times. When it's done, the Main menu will appear. The words in reverse video (white on black) are the available commands.

3 CREATE A TEXT FILE

In the menu, a command is invoked by its first letter. So, to start, hit the N key. This will allow you to create a NEW FILE (document). The screen will become white. You are now in the Editor. The small black square at the upper left marks the End of Text. It's at the start of the page because you haven't put any text in memory yet.

Now hit ENTER 5 times, then hit Up Arrow 5 times. What you've just done is push the End of Text marker out of the way. It isn't normally necessary to do this. It's done now simply for the purposes of this demonstration.

(ENTER works like Carriage Return on a typewriter. It inserts an invisible "CR" character (Carriage Return) to mark the end of the present line, and moves you down to the next line. Since it is treated as text, it pushes the End of Text marker down to make room for the new text.)

The small flashing line you now see at the top left of the screen is the cursor. The cursor is what marks your current place in the text. It flashes so it can be spotted quickly in the midst of a screenful of letters.

Hold the Shift key down and hit the zero key (i.e. Shift-0), and then release both. Notice that the cursor is flashing much slower now. This slower rate indicates that you're in **lower case mode**. The faster rate indicates **upper case only mode**. In upper case mode, all letters typed will appear (and be stored) as upper case. Lower case mode, makes the keyboard work like a typewriter -- i.e. all the letters appear as lower case, and pressing the shift key along with a letter will get upper case.

4 INSERTING TEXT -- TYPE!

Now, just type. Type :

Now is the time fo

This illustrates the most fundamental functioning of the editor: Any key hit (except for the Arrow keys, SHIFT, CLEAR, and BREAK) will be inserted into the text directly at the cursor position on the screen. The cursor moves ahead one character position so it points to the same character it started at (in this case the non-printing CR character). The inserted character is now directly "behind" the cursor.

Now finish the sentence by typing:

r all good men to come to the aid of the party.

The first thing you'll notice is that when you type the "d" in "aid", the word is instantly shifted to the next line -- this is called word wrap. It means, when you're about to exceed the end of the line, the program simply takes the whole word

and shifts it to the next line for you. As a result, you rarely need to use the ENTER key (carriage return) to terminate a line, as you do on a typewriter.

Now, hit ENTER, and then type a few more lines to get the feel of it. Don't worry about mistakes. Everything you type not only appears on the screen, it is also simultaneously stored in memory. The place in memory where the text is stored is referred to as the text buffer.

5 OPTIMIZING THE DISPLAY

The upper and lower case Telewriter-64 text display uses the Color Computer's high resolution graphics capability. Generally, on any Black and White TV (even old clunkers), the characters will be clear on the screen and quite readable. If you wish, adjust the fine tuning, brightness, and contrast controls to get the most pleasing display.

On Color TVs there is more variance as (by their nature) they are frequently fuzzier than B&W sets. The buff background color on the standard Telewriter-64 screen is optimized for B&W, but on a color TV, the buff ground leads to color sparkles around the letters. If you find this bothersome, there are 2 solutions:

a) turn the color control on the TV to zero.

b) you can change the editor screen's background color to green. This is described in appendix A of the reference manual.

6 MOVING THE CURSOR

When you've finished typing a few additional sentences, press the Left Arrow key a number of times. With each press the cursor moves 1 character to the left. Now, hold the Left Arrow key down, and press the shift key. As long as you hold both keys down the cursor moves continuously until it gets to the beginning of the text. (It will stop there because the cursor only goes where there's text and now there's no place left for it to go.)

Hit the Right Arrow key a few times. Now hold it down and press Shift. You can release either key to stop it. Press both keys down again and it goes again (until it gets to the end of text). Do this with the Up Arrow, and then with the Down Arrow.

The cursor will move in the appropriate direction but stay at the left margin.

7 THE CONTROL KEY

The CLEAR key on the Color Computer keyboard functions as the Control Key. It's used to turn other keys into commands in the following manner.

Hold the CLEAR key down and press the Up Arrow key. The cursor will jump to the top of the text (if it wasn't already there). Hold the Clear key down and press the Down Arrow key. The cursor will jump to the bottom of the text. Keep holding the Clear key down and press Up Arrow and Down Arrow alternately a few times. Leave the cursor at the top of the text.

Hold the Clear key down and press Right Arrow. Clear Right Arrow moves you to the end of the current line. Hold the Clear key down and hit Left Arrow. Clear Left Arrow moves you to the beginning of the line. (From here on, instead of saying "Hold the Clear key down and press the N key", we'll simply say "Hit Clear-N".)

Changes will be made anywhere in the text, simply by moving the cursor there and typing in what's to be added, or hitting the delete key to delete unwanted or erroneous text (see 8 below). To summarize the cursor commands:

```
Right Arrow = move cursor 1 char right
Left Arrow  = move cursor 1 char left
Up Arrow    = move cursor up 1 line
Down Arrow  = move cursor down 1 line

Clear Right Arrow = cursor to end of line
Clear Left Arrow  = cursor to start of line
Clear Up Arrow    = cursor to top of text
Clear Down Arrow  = cursor to bottom of text
Shift Any Arrow   = auto repeat cursor in that direction
```

8 DELETING TEXT

Now move the cursor to the first "o" in good (in the top line).

Hit the red BREAK key. The "o" is deleted and the rest of the line moves to the left to fill the gap. The cursor stays put and now points at the character originally to its right.

Hit the BREAK key again. The next "o" is deleted. Hit it again, and the "d" is gone. Hold the Clear key down and press BREAK. The "g" behind the cursor is deleted. So, the BREAK key deletes the character <u>at</u> the cursor, Clear-BREAK deletes the character directly before the cursor.

Now type the word "good" back in. Hit Clear-Up Arrow to move the cursor to the top of text again. Hold Clear down and hit the "K" key. "K" stands for Kill line. It deletes everything from the cursor to the end of the line. The lines below the deleted line move up to fill the vacuum.

9 CONTINUATION LINES

Move the cursor back to the top of the text (Clear Up Arrow). Start re-typing what's been deleted:

```
Now is the time for all good men to co
```

The letters will be inserted, as expected, at the cursor. The text will be pushed one space right for each character inserted, as will the cursor so it still points to the "a" in "aid" that it started at. The screen should look as follows:

```
Now is the time for all good men to coaid of the
pa
rty.
These are the lines that you've typed in. These
are the lines that you've typed in. These are the
lines that you've typed in.
```

Notice that when you hit the "t" of "to", the text below the current line moved down in a chunk (called "scrolling" down), and, as you continued to type, "o co", the "rty." of "party" appeared on the next line. This is called an overflow or continuation line. Simply, the text has exceeded the 51 character width of the screen and overflowed onto the next screen line.

Nothing is hurt when this happens. The rest of the text on the page makes room for the overflow by scrolling down, and you are able to continue inserting anywhere in the line without any problem.

Now, hit Down Arrow once. Notice that the cursor skips past the continuation part of the line. Hit Up Arrow now and notice that it does likewise. Such a "continuation line", though it occupies more than one line on the screen, is treated as a single line by many of the editor commands.

For another example, hit Clear Right Arrow. The cursor will jump to the end of the continuation line rather than the end of the current screen line. Hit Clear-Left Arrow and it jumps to the beginning of the continuation line even though it is 2 "screen lines" away from where it started.

Again, this is no problem. You can still get to any part of the continuation line with the left or right cursor commands (Left Arrow or Right Arrow). To show this, hit Down Arrow once to get to the next line (after the continuation line). Now hit Left Arrow once, and watch the cursor move to the end of the overflow line, where up cursor and down cursor wouldn't go.

Notice that the cursor is sitting after the period in "rty." where no character appears. This is the non-printing (i.e. invisible) CR character which marks the end of every line (including blank lines -- in which it may be the only character).

Hit the BREAK key now. The CR is deleted, and the line below is pulled up and combined with the current line. When the CR is deleted, the line no longer ends at this point and becomes combined with the following line. (This adds to the current overflow line as well.) Now hit ENTER. This inserts a CR at the cursor position and terminates the line, thus pushing what follows down to start a new line. (See section 2.6 of the Reference Manual for more information on CR)

Hit Left Arrow 3 times now to show that you can go anywhere in the continuation part of the line. Even though there's no problem here, however, overflow lines can cause confusion, and slow down certain editor features. (For example, inserting into a long continuation line can slow down your typing speed.)

Move the cursor to the "g" in "good" and hit the BREAK (delete) key 5 times. Notice as you do this that the overflow pulls back around the screen and, once it no longer overflows, the lines below are pulled back up as they were before. Now type "good " back in so the line overflows once again (be sure to include the space after "good_").

10 ALIGNMENT

One way to eliminate overflow lines is to insert a CR just before the first word that overflows. Move the cursor to the "p" of "party" and hit ENTER. Hit Up Arrow once then Down Arrow 4 times to prove that the continuation line is gone.

This is one way to do it, but notice that it still leaves fragmented lines. There is an easier way, which resolves this problem as well. Move the cursor back to the "p" of "party", and hit Back Arrow once. Hit the delete (BREAK) key to eliminate the CR and recreate the continuation line.

Now hold Clear down and hit the A key. This is the Alignment command. The screen is re-written and the broken lines have been filled out. Hit Up Arrow once to get back to the top line and the screen should look like this:

> Now is the time for all good men to coaid of the party. These are the lines that you've typed in. These are the lines that you've typed in. These are the lines that you've typed in.

Now hit Down Arrow 4 times and notice that it goes to each line. The continuation lines have been eliminated.

Also notice that the broken lines have been filled out, giving the text a more unified, coherent appearance. This is the other major function of the Clear-A command. Editing, inserting into and deleting from lines, invariably leaves your text chopped up -- some lines with only a few words in them, others that overflow the screen edge. The program could take care of cleaning all these up automatically, but at that point it would start interfering with your typing. Instead, it provides a user controlled command which you can invoke whenever it's desired to realign the text on the screen.

In general, when you're just typing in text, word wrap takes care of keeping the lines aligned. But when you start going into the middle of lines and inserting and deleting, the Clear-A command comes in handy to clean things up, and its liberal use is highly recommended. It can quickly become natural to hit CLEAR-A after any change that overflows or fragments lines.

Now go back and finish fixing up the line. Move the cursor to the "a" in "coaid", and type

me to the aid

Notice that when the insert gets to the right margin, wordwrap will take over and make a new line (pushing the "aid" you've just typed down to the next line) instead of overflowing. Notice also that the screen rewrites rather than scrolling down when the wordwrap happens. (This occurs when you're inserting into an existing line -- see section 2.5 paragraph 5 in the Reference manual for the way to speed optimize these situations.) While the screen is rewriting, letters typed in will not be accepted.

(If you continued typing from here, the line would again overflow when it got to the end of the screen.) Now hit BREAK 3 times to delete the extra word "aid", and hit Clear-A to fill out the lines.

In summary, then: To test for an overflow line, just run the up or down cursor by it. To get rid of overflow lines, hit Clear-A. Essentially, you don't have to care or be concerned about them, but if slightly confusing things start to happen while you're editing, or the text starts looking too chopped for your taste, just hit Clear-A.

11 ALIGNMENT - PARAGRAPHS

In the example at the beginning of section 9 above, the text was transformed from:

```
Now is the time for all good men to coaid of the
pa
rty.
These are the lines that you've typed in.  These
are the lines that you've typed in.  These are the
lines that you've typed in.
```

to:

```
Now is the time for all good men to coaid  of  the
party.  These are the lines that you've typed in.
These are the lines that you've typed  in.  These
are the lines that you've typed in.
```

What starts out looking like a separate paragraph "These are the lines that you've...", is grabbed up and combined with the preceding line, so it's just one more line in the preceding paragraph. But this is exactly what the Align routine needs to do in order to fill out broken lines.

This is something, however, you may not want it to do at certain times. Paragraphs and lists are the best examples. What if you had wanted the part you'd typed in to be a new paragraph?

Move the cursor to the period after "party" and then move it to the first letter of the next sentence. Now hit ENTER. Essentially you are inserting a Carrige Return/Line Feed character (CR) at the cursor position, so everything that follows will be on a new line. The text now looks similar to the way it looked before you hit Clear-A.

Now hit ENTER again. This inserts a line space in the text, so it now looks like:

> Now is the time for all good men to come to the aid of the party.
>
> These are the lines that you've typed in. These are the lines that you've typed in. These are the lines that you've typed in.

This is one of the standard ways of indicating paragraph breaks in text. Now hit Clear-A. The text looks like:

> Now is the time for all good men to come to the aid of the party.
>
> These are the lines that you've typed in. These are the lines that you've typed in. These are the lines that you've typed in.

Notice that even though the second section may be rearranged within itself to fill out lines, it is still left as a separate paragraph and not combined with the line above it when Clear-A does its thing.

Thus, a line space (generated by typing two ENTERs in a row at the end of the line preceding the new paragraph) indicates the paragraph break and keeps the Align routine from moving it into another line. The line space also serves this same function when the text is formatted and re-aligned for printing.

Now hit Up Arrow once, moving the cursor upto the line space below "aid of the party". Hit the BREAK key. This deletes the CR character there which marked the line space, and the text below is pulled up. (Note that the screen re-writes and the text below moves up whenever a "CR" is deleted.)

Hit the space bar three times. This inserts 3 spaces at the beginning of the line and is the same as indenting a paragraph. Now, hit Clear-A and notice that, once again, the "paragraph" is left alone.

The rule, then, for the way the alignment routine works and paragraphs are formed is this: the alignment routine feels free to move any line that is flush against the left margin of the screen. But indenting a line, one or more spaces, or preceding it with a line space (a blank line = ENTER = CR) on the screen, will preserve its position as a new line, here and during printout. The Alignment command may realign it internally but it won't deprive it of its status as a separate paragraph.

12 ALIGNMENT EVASION

In the case of lists, if you want the following:

1 Carrots
2 Apples
3 Oranges
4 Artichoke
5 Cauliflower

you would need to indent each separate line at least 1 space. If you don't, the alignment routine will turn it into:

1 Carrots 2 Apples 3 Oranges 4 Artichoke 5 Cauliflower

(You might want to try this yourself. Jump to the bottom of the text and type in the list, first, without indenting each line, then with. Leave a line space between the two lists. Then hit Clear-A and observe the results.)

Other situations in which you'd like to selectively avoid alignment include letter headings, tables, columnar material, and programs. Since there will be times that you don't want the individual lines to be indented even by one space, a special "embedded" command is provided that effectively "disables" alignment for specially marked blocks of text. This command will be discussed in section 75 below.

13 PLAY

At this point, it might be worthwhile to spend some time simply playing with the editor. Write something with it, or clean up or modify what's already in memory. See what happens with lengthy continuation lines.

14 THE MAIN MENU

For what follows, it'd be good to have a large chunk of text in memory. Rather than typing it all in, there's a faster way that'll illustrate another important feature of the system.

Hold the Clear key down and hit M (Menu). The text will disappear and you'll be facing the same menu you were facing when all this started. The Main menu gives you access to features not associated with the actual editing of the text (mainly storing and retrieving cassette text files). Use of this menu will be covered fully in sections 28-37 below and in sections 1, 3 and 4 of the reference manual.

Now hit the E key (Edit). This puts you back in the editor, right where you left off. Hit Clear-M again, and you're back in the Main menu. Returning to the menu thus, in no way, harms the text. It's still sitting there while you do what you want to do in the Main menu.

There is a demo text file on the Telewriter-64 disk and cassette. The idea now is to read this file into the text buffer so it can be edited. This will demonstrate loading a text file and will also provide a big chunk of text to work with.

15A READ IN DEMO TEXT FILE: CASSETTE

On Side 1 (label side) of the Telewriter-64 tape, the DEMO file immediately follows the program itself. The fifth command down in the menu says READ IN. To invoke it, press the R key (Read In). Next to the Read in command, the word "Sure?" appears followed by the flashing cursor familiar from BASIC. This "query" is a protective device. Reading in text will write over (i.e. destroy) any text in the buffer. The query prevents you from reading in text accidentally (when you don't want to) by forcing you to stop and think.

There's only one way to get by this point. Type a capital Y (for YES), followed by ENTER (a reverse video, lower case y won't work). The cursor will now be flashing beside the Read in command. (If you hit anything other than upper case Y, you'll be back at the start of the menu and you'll have to hit "R" again.)

The flashing cursor is waiting for you to type in the name of the file you want to Read in. If you just type ENTER it will Read in whatever file is next on the tape (the same way it works in BASIC when you do CLOAD or SKIPF without a filename). So, put your recorder on Play and hit ENTER. After a second or so, "F DEMO" should appear at the top of the screen. (You could have typed in "DEMO" for the name, as well.) The demonstration file is being read in. When it's done, the blank editor screen will reappear, with the flashing cursor at the upper left.

Hit Clear Up Arrow (the Top-of-text command) and the demo text file you've just read in will unfold on the screen with the cursor at the top of the text.

15B READ IN DEMO TEXT FILE: DISK

To access files on Disk, you need to use the DISK I/O menu. First, make sure (a backup copy of) the Telewriter-64 disk is in drive 0 (all you really need is the file S/XXX in drive 0). If it's not there, you'll get an "NE" error. If no disk is in drive 0, you will get an "I/O" error. Section 38 below will give full details on this.

Hit D (Disk I/O) in the Main Menu and the disks will churn and the Disk Menu will appear. (Note: If the disk controller is not installed or the connection is bad, nothing will happen, and the Main Menu will just flash (rewrite) quickly.)

Hit the R Key in the Disk I/O menu. As with the Main menu, this will cause the Basic flashing cursor to appear next to the Readin command. This indicates that the name of the file to be read in is expected.

This filename should be in standard RS DOS format: Maximum 8 letters in the filename, 3 letters in the extension, with slash separating the two. The disk drive is specified by a colon followed by the drive number. If no drive number is given, the default drive (or drive 0 if you only have 1 drive) is used. If no extension is given, the "BIN" extension is assumed, as in the RS DOS. (Unlike RS DOS, the BIN extension is assumed for the Name Change and Kill commands as well, if an extension is omitted.)

To read in the demo file, type in DEMO (or DEMO/BIN:0) followed by ENTER. The file will be read in (if found), and once the drive wheezes to a halt, a blank editor screen will appear with the flashing cursor at the upper left. Hit CLEAR Up Arrow at this point (the Top of Text command) and the new text will unfold on the screen.

16 SCROLLING

Hit the Shift and the Down Arrow simultaneously. The cursor runs quickly down the lines at the left margin and when it gets to the bottom of the screen, the text scrolls up a line at a time. Keep holding the two keys down for a few seconds. Now press Shift Up Arrow. When you get to the top of the screen, the text scrolls down.

17 PAGING

Scrolling allows you to move quickly through the text and still watch it as it goes by line by line. Paging moves even faster.

Hit Clear Up Arrow to get back to top of text. Now hit Clear-P. This is the Page command. The screen will pause (about a 1 or 2 second delay during which time anything you type in will not be acknowledged) and you will be at the next screenful of text in your file. For reference, what was formerly the bottom line on the screen is now the second line down from the top of the new page. Thus, you don't miss seeing anything. It's exactly like going to the next page in a book.

Hit Clear-P again. The cursor appears at the second screen line on the new page. Hit P again and you'll be at the End of Text marker, because no more text is left.

To Page in the reverse direction, hit Clear - (i.e. Clear minus sign). The cursor will appear at the bottom of the screen (until you get back to the first page). For continuity sake, this bottom line was the top line of the preceding page.

18 FINDING A PATTERN

Jump to the top of text again (Clear Up Arrow), and hit Clear-F. This is the Find command. The text will disappear and the BASIC lo-res screen will appear as it did for the Main menu. You can now type in any pattern of characters you'd like to Find in the text. Type in

the

This is called the "search pattern" because it's the pattern you're searching for in the text. These letters should appear in reverse video, indicating lower case. (If they are not, hit the Left Arrow key. As in BASIC, this moves you back and deletes the previous character. Delete all the letters this way and hit shift zero to get into lower case mode. Now, type "the" again and make sure it's in lower case (reverse video).)

Hit ENTER after the lower case word "the" has been typed in. The text will return to the screen and the cursor will be sitting at the first instance of the word "the" that it's found. Now, to find the Next instance, hit Clear-N. The cursor will be sitting at the next "the".

Hit Clear-N a few more times. The cursor will jump to the next occurrence of the pattern if it's on the current text page. If not, it rewrites the screen with the page it finds the next occurrence on (not necessarily the next screen page). When it can find no more instances, it stays put. If you jump back to the top of text (Clear Up Arrow), and hit Clear-N, it'll move through the text once again finding each successive occurrence of the search pattern "the".

The pattern you use can be anything -- a single letter, a word, a group of words, and each Clear-N will jump to the next occurrence. However, if you are looking for a pattern of, say, 3 words, like "the time for", and there's an occurrence in the text like:

> When in the course of human events is the time for all good men to come to the aid of their country...

this pattern will not be found. That's because there's a CR character (see ref. man. section 2.6 or section 7 above) after the word "time" in the example. The search pattern typed in, however, calls for a Space character in this position and, therefore, won't match. Similarly, if an instance in the text has, say, 2 spaces between "the" and "time" ("the time" vs. "the time"), this instance won't be found by the pattern given. (But see section 20 below, for a solution to the first problem.)

The Find and Next commands start their search at the current cursor position and search in the forward direction through the text. This means that to search the entire text, simply move the cursor to the top of text first (Clear Up Arrow), and then search.

The Find and Next commands are useful, not only to find a certain word or pattern for modification, but also as a way to move quickly to a given point in the text. It's infinitely faster than paging or scrolling if you know a pattern that'll get you where you want to go.

19 GLOBAL SEARCH AND REPLACE

A simple extension of the Find feature just described is called Global (or selective) search and replace. Essentially you search for a given pattern as in "Find", but now, each time you find one, you have the option of changing it to any other string of characters (or deleting it altogether) with a single command.

Go to the top of text again. Hit Clear-G (for Global search and replace). The lo-res screen will appear as it did in Find. Now type in

the

and hit ENTER. Instead of returning to the editor screen as in Find, the cursor, goes down to the next line, and waits for a "Replace pattern". So now type in the word

these

and hit ENTER. You're back at the editor screen and the cursor should be sitting at the first found instance of the word "the". Hit Clear-N a few times and it'll jump to the next few instances of "the". Now, sitting at one of the "the"s, hit Clear-R (for Replace). The word "these" will replace "the" (and the screen will rewrite from the line of the change, down). At this point you may have some fragmented lines. You can either leave them alone, or re-Align them with the Clear-F command.

You can continue moving through the text with Clear-N to find each "the" and, every time you want to replace one with "these", simply hit Clear-R. In this way, you can change every instance of the search pattern to the replace pattern, but you also have the choice of not replacing some of the instances.

If you want to delete the same pattern a number of times, you'd follow the same procedure, but simply give no replace pattern. So try this now. Jump to the top of the text and hit Clear-G, then type in:

the <ENTER>
<ENTER>

Now, back in the editor screen, hit Clear-R. The word "the" will be removed. Do this (i.e. Clear-N followed by Clear-R) a few more times. Notice that there's an extra space leftover after "the" is gone. If you want to avoid that (which in most cases you would), simply include a space in the search pattern "the ".

(To move rapidly through a global search and replace, hold the Clear key down constantly with the right hand, and alternate between N and R with 2 fingers of the left hand.)

20 WILD CARD SEARCH

You may specify an "incomplete" pattern for global search and global search and replace. If an Up Arrow (^) is included in the search pattern (by hitting the up-arrow key on the Color Computer), that position in the pattern will match any character. (And you can use any number of these "wild cards" in a given pattern.)

For example, if you hit CLEAR-F and type in the search pattern:

the^e

Then, when you go through your text with CLEAR-N finding instances, you will match words like "there, these, theme, the end, etc." This feature is also quite useful for finding multi-word combinations which might be separated by a CR rather than a space, and thus would not match a search pattern where a space was used. In general, you should use an up-arrow in place of space(s) when searching for multi-word combinations.

21 BLOCK DELETE

To Delete a Block of text, you'd first move the cursor to the end of the desired block. (A "Block" is any chunk of text -- it can be many paragraphs long or it may be a few words or letters in a single sentence.) For demonstration purposes, start at the top of the text and cursor down 3 lines with the Down Arrow key. Jump to the end of the current line with Clear Right Arrow.

Now, press Clear-E to mark the End (don't worry if wordwrap occurs because the End Mark exceeds the right margin). A dark bracket will appear, pointing left. This designates the End of the block you will be deleting. (This "End Mark" can go anywhere -- at the beginning, middle or end of a line). Now move the cursor to the beginning of the block (move it up, say, 3 lines from where you've put the End Marker) and press Clear-X. This deletes the marked block (from the cursor to the End Mark) and the mark as well. The screen will rewrite from the point where the deletion was made, down to the bottom of the page.

22 BLOCK COPY

To Copy a block of text, move the cursor to the beginning of the block. Pick someplace in the middle of the text. Press Clear-B (Begin). A dark bracket will appear pointing to the right, and at the immediate left of the first character of the block. Now move the cursor down about 5 lines and hit Clear-E to mark the End of the block exactly as was done for the block delete command (21). Now move the cursor to the top of the text. Hit ENTER a few times to make some space, hit Up Arrow a few times, and then press Clear-C (Copy). The block will be copied to the current cursor position. The screen will be rewritten, the Begin and End marks will be deleted and the cursor will be sitting at the beginning of the block as copied. The original of the copy is left intact.

23 MOVE

To effect a "Move", simply copy a block as described in 22, then go back to the original block and delete it (using Block Delete as described in 21). To facilitate finding the block again quickly, you could mark its beginning and end (to the left of the Begin mark and to the right of the End mark) with an uncommon pattern like "QQ". Then, after you've copied it, go to the top of the text, hit Clear-F, type "QQ (ENTER)" and jump immediately back to the beginning of the original block and delete it.

24 SPEED MODE

As long as you are typing at or near the end of the text, you can touch type as fast as you want and the system can keep up with you. However, once you have a large amount of text in the buffer and you go to make changes in the earlier parts of the text, the response to inserts and deletes may be a little slow.

(Note: this will not occur in 16K systems as the text buffer in these cannot hold enough text to slow the system down. If you have a 16K system you don't really need to use Speed mode but it might be good to know about it in case you accidentally hit Clear-S one day.)

Inserting and deleting start to get slow when there's 3000 or more characters (3 manuscript pages) after the cursor. If you have only a few letters to change, this doesn't matter too much, but if there's more than that, the Clear-S (Speed mode) command will allow you to insert and delete at maximum speed.

To demonstrate this, quickly quadruple the amount of text in the buffer (you won't have room to do this in a 16K system). Go to the top of the text (Clear Up Arrow), mark this as the beginning of a copy block (Clear-B), then jump to the bottom of the text (Clear Down Arrow), and mark this as the end of block (Clear-E). Now, with the cursor sitting directly after the End of block mark, hit Clear-C. This copies all the text to the end of the buffer, effectively doubling the amount of text. Now repeat this process to quadruple the text.

Hit Clear-M and look at the bottom of the menu where it says "Space". This tells you how much room is left in the buffer (how many more characters can fit in). When this number is 10,000 or less (in a 32K system), inserting or deleting near the top of the text will be slowed down. Hit E to return to the editor.

Jump back to the top of the text and hit ENTER twice to make some room for inserting. Use the Up Arrow to get back to the empty line at the top. Start typing in something like "Now is the time for all good men, etc." as fast as you can. You will probably notice a few characters missing from the line you've just typed, and you will also notice that it takes a little longer for the letter to appear on the screen (normally it's instantaneous).

Now hit Clear-S. The screen is rewritten and the whole lower part of the text is gone. The End of Text Mark (the black square, EOT) is now only 300 characters away from the cursor (about 6 or 7 full screen lines). This is really only a temporary marker now, used by Speed mode. Essentially the bulk of the text has been pushed out of the way. Now go back to the top, hit a few ENTERs to clear some lines, go back to the top and type something in. The response is fast again.

25 USUAL MODE

This is fine as long as you want to insert or delete in the area you're at. Hit Clear Down Arrow (or Shift Down Arrow) and notice that you can't get past the temporary End of Text Marker. To return to the Usual state, hit Clear-U. The screen will rewrite, and the rest of the text will reappear. Clear-U eliminates the temporary End of Text marker and re-establishes the real one.

(Note: In situations where you have a lot of text in the buffer, you may do some inserting at the middle and use Clear-S to speed things up. But then, when you move up and do some changes at the top of the text, things may get slow again, and you will need to hit Clear-S once more. In cases where you are already in the Clear-S mode and you hit Clear-S again to go higher in the text, the screen will rewrite twice.)

Since it's quite easy to forget that you're in Speed Mode, it's not uncommon to jump to the bottom of the text and be shocked because, suddenly, something's missing -- like a big chunk of your text. Whenever this happens hit Clear-U before hitting the ceiling. Similarly, if you're searching (with the Clear-F command) for something you know is there and you're not getting to it.

26 SETTING TAB STOPS

You can set tabs for columns 0 - 99. The tab settings are stored in the text as an embedded command line (82-90 below).

To establish tab stops, the very first line of the text (starting with the very first column) must be ^T followed by a space and a series of 2 digit numbers, also separated by spaces. (The numbers must be exactly 2 digits, no more no less.) So, jump to the top of the text and type the following line.

^T 05 15 25 36 42

This will set tabs at 5, 15, 25, 36 and 42. Remember that the ^ character is generated by hitting CLEAR and period (CLEAR-.) at the same time. Also note that, since the first character on a text line is considered to be at column 0, a tab stop of 5 will actually be the 6th column on the line.

The tab key is CLEAR-ENTER. Hitting it will jump you to the next tab stop on the line. If no tabs are set, it will do nothing. So jump to the bottom of the text, hit ENTER a few

times, and hit the tab key (CLEAR-ENTER). The cursor will jump to column 6. Type a 4 digit number and hit the tab key again. The cursor will jump to column 16. Type another number and tab over. When you've typed 5 numbers on the line, hit ENTER and type another line of numbers using the tab key. As with a typewriter tab key, the columns are aligned at their left.

Hit ENTER at the end of the line and do another line, but this time don't type in numbers, just hit the tab key. Notice that it won't go past the last stop. Hit CLEAR left arrow to jump to the start of the line. Now hit CLEAR Right-Arrow to jump to the end. Notice that, even though the line appears blank, it is filled with spaces. If there is nothing on a line, then each time you hit the tab, spaces are inserted into the line until you get to the next stop. These spaces remain there, unless they are explicitly deleted with the line or character delete commands.

This may not matter, but a few problems are associated with it. For one thing, a delete won't eliminate the line (but a line Kill (CLEAR-K) will). Keep this in mind, then, should lines of spaces mysteriously appear in your text.

Now hit the up arrow once to get to the start of the line with numbers on it. Hit the tab key and notice that it jumps to the established tab columns. If there is already text in the line, spaces will be inserted only when the end of the existing line has been passed and only upuntil the next tab stop.

Note that if a tab stop exceeds the current screen width (e.g., a tab stop of 60 in the 51 column mode), tabbing to it will create an overflow line, with the cursor at column 60 on the line, but at column 5 on the screen (because of the overflow). If, however, you're typing in an entry, and you exceed the right screen margin, wordwrap will pull the whole entry down to the next line and insert a CR, thus destroying the integrity of the line you're tabbing across. If you now continued tabbing, you would be essentially starting a new line.

So, in general, if you are doing a lot of columnar work, disable word wrap (CLEAR-D), or do it in a screen mode that contains your farthest margin.

It is also wise to protect tabular material from the destructive effects of alignment and right justification, by surrounding it with the disable alignment/justification embedded command (see 75 below).

27 OTHER EDITOR FEATURES

Several features which are available in the editor mode have not been discussed. Since these features relate more to printing and page formatting, they will be taken up in the sections dealing with those topics. These features include: The high density screen modes (64 X 24 and 85 X 24), two character sets, setting screen line width, hyphenation, page finder and comment lines (see 77-81 and 84).

Below is a summary of Editor commands and special characters:

One key commands:

ENTER = Terminate line (Carriage return)
BREAK = Delete character at cursor
Up Arrow = Move cursor up 1 line at left margin
Down Arrow = move cursor down 1 line at left margin
Right Arrow = move cursor right one character
Left Arrow = move cursor left one character

Commands preceded by CLEAR ("control") key:

A = Align text lines to fit current line width
B = Begin Text Block Marker
C = Copy block
D = Disable wordwrap
E = End Text Block Marker
F = Find a search pattern
G = Global (Selective) Search and Replace
K = Kill line
M = Menu (Return to Main Menu)
N = Next instance of search pattern (after Find or Global)
P = Page forward through text
Q = Search for special characters (block & format codes)
R = Replace search pattern with replace pattern
S = Speed Mode
U = Usual mode (to get out of Speed Mode)
V = Vertical Tab (Page Break Finder)
W = Wordwrap mode Enable
X = Block delete
Z = Delete all Block Markers (Begin & End)
- = Page backward through text
0 = 51 column mode
: = 64 column mode
@ = 85 column mode
; = Toggle character set

ENTER = tab
BREAK = delete character before cursor
Up Arrow = top of text
Down Arrow = bottom of text
Right Arrow = end of line
Left Arrow = beginning of line

. = embedded format code
/ = underline delimiter for MX-80 only
, = backslash
1-9 = user defineable control codes, upto 15 values each

SHIFT CLEAR =]
SHIFT @ = [

28 MAIN MENU -- CASSETTE HANDLER

So much for the editor. Clearly it's the heart and soul of a word processing system, but without the ability to easily store and retrieve all the work you've done in the editor, you might as well use a typewriter (almost).

One of the functions of the Main menu, then, is to give you easy access to the storage and retrieval of your text. The following sections will discuss doing this on cassette.

(NOTE TO DISK USERS:

Though disk is obviously faster and easier to use than cassette, people with disk Telewriter-64 may do well to read these sections, since all the cassette capabilities discussed are available on the disk version. Cassette, it turns out, makes an excellent (and in many ways "surer") backup medium for anything you put on disk. It's quite easy for a DOS to simply wipe out your entire disk at any random moment, and then wipe out your backup disk when you try to use that. On cassette, you, yourself, personally and unequivocally, must position the cassette at important data, and then command it to record over it. Though the tutorial examples used below assume that you have the Telewriter-64 tape, it is easy to create a tape with a few files on it to test out the features. Section 32 discusses how to do this.)

Since you've already used the Read In Command in section 15, the next sections will deal with the remaining cassette operations.

29 VERIFY

The Verify command will find a file and read through it til the end, but it will not read it into memory, and its use will not effect anything already in memory. If a tape error is found, it will stop the tape and report "I/O ERR".

To use Verify, get back to the Main menu, from the editor by hitting Clear-M. Rewind the tape, set the counter to 000, and put the recorder on Play. Hit V to invoke the Verify command. The flashing cursor will appear beside the word Verify on the menu. It's waiting for a filename or for ENTER. If you give it a filename (terminated by ENTER), it will search the tape for a file with that name. If you give it no name and only hit ENTER, it will find the next file on the tape.

So hit ENTER. S will appear at the upper left indicating that a file is being Searched for. It will not be "flashing" because the tape is blank at this point (it only flashes when there is data on the tape and a file has not yet been found). Eventually it will find the file TELE64 and F TELE64 will appear at the top of the screen.

When it gets to the end of the current file the tape will stop (as it does with SKIPF) and the menu will return. If, however,

there was something wrong with the file before it got to the end you would get an I/O ERR message beside the filename. Section 32 below explains how to deal with this.

30 AUTO-RETRY

Now, fast forward the tape about 10 counter numbers (to approx. 21 -- note that these numbers assume a CTR-80A recorder -- they may differ for other models). This will put you in the middle of a second TELEWRITER file. With the recorder still on Play, hit V again, followed by ENTER. The S in the upper left will be flashing, indicating that it's searching for the start of a valid file (and that there's data on the tape where it's currently searching). Eventually it will find DEMO, and skip to the end of it. (If it doesn't find it by 35 on the counter, stop the tape, back it up to about 29 or 30 and put it back on Play. See Ref. Man. Appendix B, Section 5.)

This illustrates the "auto-retry" feature. If you were doing a CLOAD or SKIPF in BASIC, what here appears as the first flash of the S would have given you an I/O error and you'd have to type the command again (and again (and again)). Essentially Telewriter does this for you, and does it until it finds a file or is aborted.

Now back the tape up about 12 counter numbers (so you're back in the middle of the TELEWRITER file at about 021) and repeat what you've just done (hit V followed by ENTER). But this time, after the S has flashed a few times, hit the BREAK key. Hold it down for a second or two and the program will return to the Main menu. The filesearch has been aborted, and you're free to issue any menu command at this point. This means of aborting a filesearch will work the same with the Append and Read in commands as well. It will not work if the S is not flashing (i.e. no data on the tape at that point). Put another way -- if you want to abort a filesearch, you need to have it running over a tape with some kind of data on it, so the S is flashing.

The Verify command, combined with the auto-retry feature, is extremely useful when you're lost in a tape. It works essentially as it worked above. (Without it you'd have to type SKIPF in BASIC an infinite number of times, or else rewind and fast forward to find your place.)

31 APPEND

Your tape is still positioned in the middle of the Telewriter file. With the recorder on Play, hit A. This is the Append command. As with Verify and Read in, the flashing cursor appears next to the word Append waiting for a filename. Hit ENTER. This will cause the next file found on the tape to be Appended to the text already in the buffer. That is, the text in the new file will be tacked on to the end of what you already have in memory.

The S should flash a few times while the program searches for the first valid file. Eventually, it should find DEMO ("F DEMO" will appear at the top of the screen). When the file finishes loading, the editor screen will reappear, blank except for the flashing cursor at the top left. As in 15, hit Clear Up Arrow to get the text to unfold on the screen.

Hit Clear Down Arrow to jump to the bottom of the text and note the new text that's been added at the end. Scroll up (using Shift Up Arrow), or page back (using Clear -), and notice that there are two copies of DEMO in the buffer -- the one you'd been working with, and the one you just appended.

Go back to the Main menu now, and notice that after the word FILE:, the filename DEMO appears. The Main menu always shows the name of the last file Read in or Appended in this place.

(Note: If you have a 16K system you will get a "TOO BIG" error message because there is not enough room in the buffer for 2 copies of DEMO. Hit BREAK to get back to the menu, go back to the editor and delete all the text except 1 or 2 lines. Rewind the tape about 3 counter numbers and do the Append command again. The demo file should fit this time.)

32 SAVE

Save (and % Save) is the only command that writes to the cassette. As such, it can destroy anything already on the tape that's "in its way". Since we want to demonstrate saving a file to tape, take out the Telewriter-64 tape and put in a blank one.

Rewind to the beginning, reset the counter to 000 and fast forward past any leader on the tape (going to 5 on the counter should do it).

Press Record and Play down together. Return to the menu (Clear-M) and hit S (for Save). Again the cursor asks for a file name beside the appropriate command. When saving a file it is always important to give it a name (whereas, when Retrieving or Appending, as long as you know your position on the tape, it's easier to just hit ENTER). So type in "TEMP1" for its name and hit ENTER.

As soon as you hit ENTER, the tape will start. All the text in the buffer is now being stored on the tape in a file with the name TEMP1. (The filename can be no longer than 8 characters, and should be chosen in such a way as to give you information about its contents. In this instance, TEMP1 indicates the file is only temporary -- i.e. not too important.)

When it's done, the recorder will stop, the Menu will appear to flash once, quickly, and the name you typed in will be gone.

Now rewind the tape, and Verify that it's been properly stored. Put the recorder on Play and hit V ENTER. The S will appear at the top of the screen (not flashing because the tape

should be blank at this point) and eventually it'll get to the file you've just saved and F TEMP1 will appear at the top of the screen.

If the save has been successful, the tape will go to the end of the file, stop and the menu will return to the screen. If not, the tape will stop and "I/O ERR" will appear after the filename at the top of the screen.

Hopefully, you didn't get an I/O error, but, if you didn't, it's worthwhile to simulate one. Rewind the tape to the beginning, put the recorder on Play, hit V (ENTER), and wait til the TEMP1 file is found. Then stop the recorder before you get to the end of the file.

Now hit Play on the recorder again. It'll run for a second or so and then "I/O ERR" will appear beside the filename at the top of the screen. At this point, hit BREAK (ENTER will work too), and you'll be returned to the Main menu, no harm done.

These same procedures hold true for the Append and Read in commands. If an I/O error is encountered it'll be reported in the same way, and BREAK will get you out of the error condition, and back to the Main menu. Similarly, if you wish to abort a Read in or Append at any point after the file has been found, simply stop the tape, restart it again, and hit BREAK when "I/O ERR" appears next to the filename.

If you do get an I/O error during Verify, you should try to store the text again. (You might try to save it again in the same place and if an I/O ERROR appears again when you verify, this indicates there is probably something wrong with the tape at this point. Re-store the file someplace else on the tape or on another tape.)

33 PARTIAL SAVE

The % command in the menu allows you to save a part (%-age) of the buffer. Use Clear-E to mark the end of the block to save (as you did in block delete), and move the cursor to the beginning of the desired block (as in block delete). Avoid changing anything in the block once the End Mark is set. If you do make changes, delete the End Mark and re-do it.

Now, return to the Main menu and hit %. The marked block will be saved to cassette exactly as with the Save command.

34 OTHER MENU OPTIONS: WORD COUNT

In the Main menu, hit W. The number of words in the text buffer will be displayed next to the word "words". The number of lines is also displayed. A line is counted from CR to CR. You can use the Line count to give a rough idea of how many pages you've got, though this can be done much more accurately with the Page Finder feature (see 81 below). Chars/Lin must be set to the printed line width to get an appropriate number for page counts.

35 STATUS INFO

The name after FILE shows the last file read in or appended (even if there was an I/O error, or the function was aborted).

The number after SPACE is the amount of space left in the buffer for characters. On initialization, this number is around 16000 in 32K systems, (and about 2800 in 16K), indicating that 16000 characters can be typed into the buffer (approx. 8-9 pages). This number cannot go below zero. When an insert would cause this to happen, a "NO ROOM" error message will appear on the screen. As with the other error messages, BREAK is used to exit the error condition and return to the menu.

36 NEWFILE

You've used N in the beginning to start editing. NEWFILE starts a new text buffer. In so doing, it wipes out whatever was there before. Since it's so destructive, it queries first, the same way Read in did. And the response works the same. Capital Y followed by ENTER will start a new buffer, anything else will return to the menu. If there's no text in the buffer, it won't query (which is what happened (didn't happen) the first time you used it).

37 RETURN TO BASIC

In cassette based Telewriter-64, hitting B in the Main menu will get you back to Basic. When you do this, the screen blanks and OK appears at the top. If you type a BASIC command, now, it will execute.

This feature allows you to use BASIC's arithmetic powers while in the midst of text editing. It also allows the loading and running of Telewriter-64 utilities like the ASCII I/O program described below, and the forthcoming Telewriter mail merge program.

You can even run your own small programs (provided they don't interfere with Telewriter's memory areas), though, once you start doing that, we can't promise that your return to Telewriter will be either clean or complete. If you plan to do this, make sure you test it first on inconsequential text.

When you wish to return to Telewriter, simply type EXEC (provided you have not changed location 157-8), and you will be back in the editor, facing a blank Hi-res screen with a flashing cursor. Hit CLEAR Up Arrow to unfold the text.

38 DISK MENU

When you want to save, retrieve or append something on disk, you need to use the DISK I/O Menu. This was described briefly in section 15. To access it, go to the Main menu and hit D (the Disk I/O command). Drive 0 will thunk a few times and a new menu will come up, similar to the Main menu, but with "BINARY DISK I/O" at the top. (On a cassette based system, hitting D will do nothing.)

To get the Disk menu, there must be a diskette in drive 0 with the file S/XXX on it. This can be the Telewriter-64 disk itself (which comes with S/XXX on it) or any other disk with an S/XXX copied from the Telewriter-64 disk. (To copy the file easily, first, turn on the machine and LOAD"S/XXX". Then take any number of formatted disks and SAVE"S/XXX" to each. From then on, if you use only these diskettes for text storage, you will avoid a lot of NE errors. Make sure, however, that you don't create your own file and name it S/XXX.)

If there is no disk in the drive, it will thunk 4 or 5 times and give you an I/O Error. If there is a diskette in Drive 0, but no file "S/XXX" on it, you will get an NE Error.

The solution is to put a diskette with S/XXX on it into Drive 0, then type CLEAR 10 (ENTER) and then RUN "S/XXX" (ENTER). If you don't do the CLEAR 10 to recover from the I/O error, you will get an OS error.

The disk menu works exactly like the other Telewriter-64 menus. The conventions followed are those described in section 1 of the Telewriter-64 Reference Manual: Hitting the first letter of any menu item will invoke that command. The flashing BASIC cursor means the menu is waiting for a filename to be input, followed by (ENTER). Additionally, the dark square beside a command is waiting for a single keypress, representing a drive number.

39 EDITOR

Hitting E will return you to the editor as it does in the Main menu. The difference is that returning to the editor from the Disk menu will present you with a blank editor screen. Simply hit CLEAR Up Arrow and the text will unfold (the same procedure you follow after reading in a file from cassette or disk).

40 SAVE

Hitting S in the Disk menu allows you to SAVE a file to disk. The flashing cursor will appear next to the Save command waiting for a filename in the Standard RS DOS format. Type in the file name and hit ENTER. The file will be saved to disk and the Disk menu will reappear. From here you may do other disk I/O or return to the editor.

41 % SAVE

This allows you to save a part of the text to disk. First, in the editor, you must mark the section of the text to be saved. As in Block delete (2.11), Partial save to cassette (3.3), and Partial print (5.10), CLEAR-E is used to mark the end of the desired block, and the cursor is placed at the beginning of the block. Then, return to the Main menu with CLEAR-M, and go to the Disk menu with the D command. Hit the % (per cent) command and give a valid filename, followed by ENTER. This works exactly like the Save command except only the marked portion of the text is saved instead of all of it. Note that %-SAVE disk

files should not be used in print chaining (see 72 below). You should also avoid making changes to the block once it's been marked. If you do, delete and re-set the mark before saving.

42 APPEND

The A command allows you to append a file from disk to what you have in the buffer already. Simply hit A and give the file name. If the file to be appended plus the text already in the buffer will exceed the space available in memory, a TOO BIG ERROR will result. The number following the error message indicates how many characters must be deleted to allow the file to be appended properly. Eliminate this number of characters in the text and try to append the file again, or else read in the append file and delete the parts of it you don't want, and then append the original file (previously saved).

43 NAME CHANGE

The N (Name change) command is roughly equivalent to the RS DOS "RENAME" command. To change the name of a file on any drive, hit N. The flashing cursor will appear to the right of the Name Ch command. Type in the original name of the file to be changed and hit ENTER. The disk will run while the presence of that file is verified (if it doesn't exist a "FILE NOT FOUND ERROR" will appear at this point -- saving you the bother of also typing in the "target" filename).

If the file is found, the flashing cursor will now appear slightly below and to the right of the first name. It's waiting for the new name. Type it in and hit ENTER. The disk will churn a little and the file will be renamed.

The one thing to note here is that, unlike the RS DOS "RENAME" command, if no extension is given with the filename, "/BIN" will be assumed.

44 KILL

This is the same as the RS DOS "KILL" command. Hit K, and give the filename. Unlike RS DOS, if no extension is given, the "/BIN" extension will be assumed.

45 FILES

This is the same as the RS DOS "DIR" command. Hit F. A black square will appear to the right of the File command. Then, hit the number of the disk drive you'd like listed. The screen will blank and the names of the files in the directory will be written out in the standard RS DOS format. At the end of the Directory listing, Telewriter-64 will print the number of free granules remaining on the particular diskette being listed.

If the listing exceeds the size of the screen, you can pause it with the usual Basic SHIFT-@ command. Hit any key to continue the listing. Hitting BREAK will cause a break in the program, and "OK" will appear on the screen. By hitting BREAK again, or ENTER, you will be immediately returned to the Disk menu.

When the listing is done, the flashing cursor will appear below the free grans number. Hitting BREAK or ENTER at this point will return you to the Disk menu.

46 BASIC

Hitting B in the Disk I/O menu will allow you to return to BASIC. This is for convenience and for those people who might want to PEEK or POKE something, do some arithmetic in Basic, or take advantage of any disk operations not contained in the Telewriter-64 menu (like turning VERIFY ON or OFF).

But beware. Anything you do in BASIC (aside from the operations just mentioned) you do at your own risk. Doing a COPY for example, will mess things up, so don't do it. To return to the Disk menu, simply type RUN. This will bring the disk menu back as long as you have not read in another BASIC program or destroyed any of the lines currently in memory. If you have, typing RUN "S/XXX:0" should get the menu back. If you get an "OS error" when you try to Run S/XXX, type in CLEAR 10 (ENTER), and try again.

(If it doesn't work, hitting reset should get you back to the editor. If that doesn't work, try to remember what you've done, and don't do it again. Note that hitting reset will cause the system to reboot in a 64K Color computer. The only hope at this point is to do a SAVEM"DUMP", 7790, 32000, 32000. This will save the text buffer and then some, in a form that can be read back in as a normal Telewriter text file.)

47 PRINT DIR

It's hard to believe that RS DOS doesn't give you a simple way to print the directory to your printer. Hitting P in the Telewriter-64 Disk menu will allow you to do this. When it's first hit, the screen will go blank and the word "ID" will appear at the top. This will allow you to label the printed directory listing so you know what disk it's a listing for. Type in whatever you want that will serve to identify it for you.

After you hit ENTER, the printer will do 2 line feeds, and the ID will be printed. When this is done, the word "DRIVE" will appear on the screen with a black square to its right. As with the FILES command, you simply hit the key whose number corresponds to the number of the drive you want the listing to come from. As soon as the key is hit, the screen will go blank, the disks will whine and, if you are fortunate enough to have a drive that doesn't frequently wipe out your directory, a listing will be produced, showing the number of free granules at the end. Then, the Disk menu will reappear on the screen.

Note that you must have first set the appropriate BAUD rate for your printer, either in the Telewriter-64 Format menu, or by returning to Basic and POKEing the appropriate number (see 59 below) into location 150 (decimal). You must also have explicitly sent any printer control codes (e.g. for emphasized printing) beforehand. This can be simply done using the Format menu's Direct command (See 69 below), with the printer on line.

48 DEFAULT DRIVE

The default drive is set in the Disk menu with the D command. A flashing cursor will appear at the right of the word DEFAULT. Hit the Drive number desired (0, 1, 2, 3) followed by ENTER. The value set in this fashion will continue to appear at the bottom of the menu as a reminder.

49 CONVENTIONS

Note that in the Disk menu, the following convention is followed. After a command key is hit, a flashing cursor on the screen indicates that input is to be given, followed by the ENTER key. If a black square appears, this means that a single key press will initiate the function. The latter is generally used for drive numbers, the former, for filenames. (The one exception is setting the default drive, which requires ENTER even though it's a single number.)

50 STATUS INFORMATION

As in the Main menu, SPACE indicates how much room is left in the buffer, and FILE indicates the name of the last file loaded or appended from either cassette or disk. (At the end of a chain print from disk, however, there will be no filename info.)

51 ERRORS

Telewriter-64 Disk error messages appear on the second line of the Disk menu, just below the words "DISK I/O". A flashing cursor will also appear to the right of the error message. As usual, BREAK or ENTER will get you out of the error condition, and back to the menu.

Error conditions will occur:

1) If you type in a filename that is larger than 8 characters, or an extension that is larger than 3 characters ("NAME TOO BIG" error).

2) If you try to read in, append, change the name of, or kill a file not on disk or the drive you specify ("FILE NOT FOUND" error).

3) If you try to append a file which would exceed the capacity of the text buffer ("FILE TOO BIG" error).

4) If you specify a drive # higher than 3, or higher than the number of drives you have, ("DRIVE # TOO HIGH" error). (See 52 below for the way to customize the program for the number of drives in your system.)

5) If your disk directory has been destroyed, or there is a problem reading or storing a file to disk. In these instances, you will get an error message from the RS DOS. This will either be an I/O error or a DF (disk full) error. These messages

will break into the Disk menu when they occur and will be followed by the word OK and the flashing Basic cursor. As above, simply hit BREAK or ENTER and the Disk menu will reappear.

52 CUSTOMIZING

To avoid the unnecessarily long wait that occurs when you try to access a drive you don't have, Telewriter-64 lets you customize the disk interface program to suit the number of drives that you actually do have.

To do this, turn on the computer and LOAD "S/XXX from the disk. Type EDIT 2 , and change the value of the variable MX in line 2. Set MX to the highest drive number that you have. E.g., if you only have 1 drive, that's drive 0, so set MX = 0 in line 2. If you have 2 drives, set MX to 1, etc.

Then do SAVE "S/XXX, saving this customized version to this and other disks you intend to use for text storage. (Note: don't do this until you've backed up the original Telewriter-64 disk, and put the original away.)

Once MX is properly set and saved in the program, accidentally hitting the wrong number in the Files command, for example, will get you a quick "DRIVE# TOO HIGH" error message, rather than a 2 minute churn from your drives, or a BASIC DN error.

53 ASCII I/O OPTION

The standard Telewriter-64 format for disk and tape is the "binary" format (i.e. compatible with files created by SAVEM and CSAVEM, and read by LOADM and CLOADM). On tape, at least, this is the most streamlined format in terms of speed and space.

There are, however, situations where it is desirable to have files in the standard ASCII format (the format of BASIC data files, and BASIC programs saved with the "A" option). Such uses include writing Assembly programs, "C" programs, PASCAL programs, BASIC programs, running text files through spelling checkers and preparing and editing text to send to or receive from other computers over the phone lines.

For these applications, a utility program is provided on disk and tape that will save Telewriter-64 text files in ASCII format as well as Read or Append an ASCII format file into the Telewriter-64 text buffer (after which it can be re-saved in either ASCII or binary format).

54 ASCII I/O: CASSETTE

On cassette, when you wish to save, append or read in an ASCII format file, do the following: First, return to the Main menu and, once there, hit B. This returns you to BASIC. Now put side 1 of the Telewriter-64 tape in the recorder. If you loaded Telewriter-64 from side one, you need only do a SKIPF to get past the DEMO file. Then, type CLOAD, and the filename "A" will appear on the screen as the ASCII I/O program is read in.

(If you are lost in the Telewriter-64 tape, use the Main menu Verify command before returning to BASIC, to get to the start of the next "A" file. "A" is stored 11 times, one after the other, on side 1 of the Telewriter-64 tape, so once you are past the DEMO file (or after reading in "A"), you will always be sitting at the beginning of the next "A" file.)

Loading "A" will only take a few seconds. When it's done, type RUN. A menu will appear. Treat it exactly as you would the Telewriter-64 Main menu for reading in appending or storing files. The same conventions hold for filenames and the auto-retry feature. (However, if you abort a file search while the "S" is flashing, you will be returned to Basic rather than the menu. Simply type RUN and hit ENTER, to get it back.)

When you are done, hit E in the ASCII menu, and you are returned to the Telewriter editor. Should you break out of the program, EXEC or EXEC 8505 (EXEC 57105 in 64K), will get you back to the editor. (Or, type RUN to get the menu back and then hit E.) You will be facing the blank editor screen. Hit CLEAR then Arrow to reveal the text.

55 ASCII I/O: DISK

On disk, there is a file called S/ASC. It is quite similar to the file S/XXX, that normally does disk I/O for Telewriter, but S/ASC does the I/O in pure ASCII.

To do ASCII I/O on disk, then, do the following: Hit D in the main menu to get the Disk I/O menu. Now hit B so you return to Basic, and then type RUN "S/ASC". (Note, if the file "S/ASC" is not on the disk currently in the drive, you will get an error from Basic. Put a disk in this drive that has the file on it, and run it. If you copy S/XXX onto new disks, you should copy S/ASC at the same time.)

A new menu will appear, similar to the regular one, but it will say "ASCII I/O" at the top. Simply read in, append, or save you would with the regular "Binary" menu. The ASCII saves and loads will take a little longer and will give you more visual entertainment on the screen, but otherwise, everything else will be the same as in the binary menu. However, ASCII files saved with no extension will default to the "DAT" extension, where Binary files will default to the "BIN" extension.

This is all pretty simple, but, if you use both formats, it now becomes incumbent upon you to keep track of which files are ASCII and which are BINARY, and which I/O system you are using at the time. If the Binary program tries to read an ASCII file or vice versa, you will get an unmitigated error.

In general, you should do everything in Binary format because it is faster. The only things that need to be in ASCII are programs, or files you want to send through a spelling checker, or over the phone lines. You might want to keep your ASCII files on separate disks.

If you are doing a lot of I/O that must be ASCII (especially if you are programming either in BASIC or in Assembly language), it is a nuisance to have to return to Basic and run the ASCII program everytime.

Since Telewriter-64 looks for the Program S/XXX, it is a simple matter to RENAME"S/ASC"TO"S/XXX". Then, whenever you hit Disk I/O in the Main menu, you will get the ASCII Disk I/O menu everytime. Of course, you've now lost the original S/XXX.

But don't fear. The Telewriter-64 disk contains a file S/BIN, as well as S/ASC and S/XXX. If you plan to use both file formats (ASCII and binary) on a given disk, you should put all 3 of these files on it. When you do ASCII I/O, simply go into Basic, LOAD "S/ASC and then SAVE "S/XXX and RUN. From then on, you will always get ASCII I/O when you hit D in the Main menu. When you want to do Binary I/O for a while, return to BASIC, LOAD"S/BIN and SAVE"S/XXX. This is a little easier than using RENAME, which requires that you do 2 sets of renames. Of course, the best solution is to maintain separate disks for ASCII with S/ASC saved as S/XXX on all of them.

But, whatever you do, make sure you maintain copies of S/ASC and S/BIN independent of S/XXX.

56 ASCII UTILITIES -- IMPERFECTIONS

It should be noted that the ASCII programs are considered utilities ancillary to Telewriter-64 and are provided for the convenience of the user. Though they have been brought to a reasonable level of completion, there are a few rough edges, and the programs differ in a few minor ways from their binary counterparts.

In the cassette ASCII program, there is an auto-retry feature similar to the binary one. However, if you BREAK out of this while it's searching for a file, you'll be returned to Basic rather than to the menu. At this point, it's simply a matter of typing RUN to get back to the menu.

Similarly, the Disk ASCII I/O program will occasionally fail to trap an error. When this happens, you will be returned to Basic, from which, typing RUN will again get you back to the menu. BREAKing out of a Directory listing will also put you back into Basic. Typing RUN does the trick here too.

The only real problem is this. The Cassette ASCII program must use BASIC's "INPUT" command (rather than LINEINPUT) in a Color Computer without Extended Basic or in a 64K system (where Extended Basic is disabled). The INPUT command has its own idiosyncracies, one of which is to generate an FD error whenever it reads in a line beginning with a quote and having a second quote in the line. If this happens while inputting an ASCII file, you will get an "FD ERR in 448" message, and input will stop. The solution is simply to type GOTO 451. The program will then continue reading in. The rest of the line, after the second quote, will be lost, however.

57 FORMAT & PRINT MENU -- SWAPPING

All the sophisticated full-screen editors in the world won't do you any good if you can't get your words off the screen onto paper (and onto paper in the way you want). The Print/Format menu gives you access to these capabilities. To get to it, go to the Main Menu and hit "F".

Note: For increased text space, the print and format routines may be non-resident -- i.e. they can be read in automatically from disk or tape when you want to print your text (called "swapping"). This frees up an additional 3K of memory. It is a feature which you may chose to use or not if you have 32K, and a feature you must use if you have 16K. If you're in the latter category, please read the remainder of this section. If not, you might want to skip to section 58, and read this section at a later time.

DISK: On disk, the "swapping" takes place automatically and quickly so it is virtually unnoticeable. When you hit F in the Main Menu to go to the Format Menu, the disk will churn briefly before the Format Menu appears. The only thing you need to keep in mind is that the file "F/" (which is provided on the Telewriter-64 disk) must be on the diskette in drive zero when F is hit in the Main Menu. If it isn't, an "NE ERROR" will flash briefly on the screen and you'll be returned to the editor, without seeing the Format Menu.

Thus, if you have 16K or if you want an additional 3K of text space in your 32K system, it is a good idea to copy the file "F/" to all the disks you copy S/XXX to (see section 38). (Also note that the file has no extension. To copy it, do a COPY"F"TO"F:1" (or COPY"F" TO "F" in a 1 disk system).)

If you wish to take advantage of the non-resident printer driver in a 32K system, you must type LOAD"U" instead of RUN"U when you start Telewriter-64. After "U" is loaded, type:

```
2 SW = 1
```

and hit ENTER. Now type RUN and proceed as normal. You will notice that the SPACE value in the menu is now around 19000 instead of 16000. If you opt to use this, it is recommended that you do so consistently since, if you have some large files saved from a 19K run, they may be too large to load into a 16K run. Chain printing will also not work properly if you try mixing files from 2 different size systems.

CASSETTE: In a 16K cassette system, when you hit F to go to the Format menu, you will be prompted to "LOAD THE FORMAT TAPE". To do this, place the original Telewriter-64 tape into the recorder -- <u>on side 2</u>. This side contains Telewriter-64 from approximately counter # 6-30. Following that, there are repeated copies of a file called "F", til the end of the tape. This file contains the print and format routines. If you load Telewriter-64 from side 2 of the tape, then you will be sitting at the beginning of the first "F" file. If you have already read one in, then the tape will be sitting at the beginning of the next one, and so on.

Thus, if you have a 16K system, you should always load Telewriter-64 from side 2 of the tape (the copy on side 1, serving as a backup).

So insert the Telewriter tape on side 2 and hit PLAY (if you've loaded Telewriter-64 from side 1 of the tape, simply turn it over and rewind 30 or 40 counter numbers). Now hit ENTER in response to the prompt on the screen.

The tape will spin, and "F" will be loaded. If you're not sitting at the exact beginning of one of the "F" files, or if one is damaged, then auto-retry will keep going til it finds the first good one. (Note: As long as the S is flashing on the screen (without a file name beside it), you may abort the search by hitting the BREAK key, which will return you to the Main menu.) It takes about 15 seconds for the print routines to load, from the start of the file. When it's done, the Format Menu appears.

58 FORMAT MENU

The words in reverse video (dark background) on the top half of the screen are the format parameters -- i.e. the aspects of the final printed page over which you have control. The numbers to the right of these parameters indicate their current values.

You can immediately print everything you have in the text buffer at this point, if you want. To do this, 1) make sure your printer is properly connected to the RS-232 connector at the back of the computer (exactly as it would be for BASIC listings or any other use), and 2) make sure the baud rate is properly set.

(Note: also make sure that the number following QUEUE in the menu is set to zero. This value should only be non-zero when you are "chain printing" (see 71 below), and can make the program appear to "hang" after a print, if it is non-zero.)

59 BAUD RATE

The baud rate is the rate at which data is sent from the computer (via the program) to the printer. The program transmits at a certain rate, and the printer receives at a certain rate, and the two rates must match. So first determine the baud rate your printer is running at and then simply set Telewriter-64 to match it.

(If you use your printer with BASIC and have never set its baud rate and have never had to think about it before, then you are probably running at 600.)

If your printer is at 600 baud then you don't need to do anything. The program's baud rate is initially set to 600, and will remain at that value unless explicitly changed by you. (However, if the number to the right of the word XMIT RATE in the Format menu is not 87, then follow the instructions to set the baud rate.)

To set the baud rate in the Format menu, type X. (This stands for Xmit Rate which in turn means "transmit" rate.) Now refer to the chart below. Find your baud rate, and type in the number to the right of it.

printer baud rate	Xmit rate value
110	498
120	458
300	180
600	87
1200	41
2400	18
4800	6

Then hit ENTER. The menu will display the new transmit rate.

60 PRINT -- ABORT

Now that the baud rate of the program matches the baud rate of the printer, simply make sure your printer is on-line and, then, hit P (Print). The word "Print" will appear on the screen to the right of the print command.

The printer will do a few lines spaces at the beginning and the text will start to print out. (It'll be relatively narrow on the page because it still has the same line length it had on the screen -- only 61 characters wide.)

It's probably a waste of time, paper and printer-life to let this printout continue, so abort it by hitting the BREAK key at any point (hold it down for a few seconds if the printing does not stop immediately). The printing will stop, the word "Print" will disappear from the screen and you'll be back at Format menu level where you can issue further commands and change format parameters.

61 PARTIAL PRINT

If you hadn't aborted the printing, it would have continued for a page or so until everything in the buffer was printed out. But there are often times when you would like to see only a small section of the text printed.

To print out only part of the buffer (a %-age), the "%" command is provided. Before using it, though, you need to designate the section you want printed. This is done in the same manner as marking a block of text for deletion or partial save (see 21 & 33).

First hit R. This Returns you to the Main menu. From here, hit "E" to get back into the text. (In a 16K system, the R "R"eturns you directly to the editor. At this point, just hit CLEAR-Up Arrow to unfold the text.) Assume that the part of the text you want to print is the last page. So simply jump to the bottom of the text (Clear Down Arrow), and hit Clear-E to mark this as the end of the block to be printed.

Now move the cursor up to the beginning of the block. Move it up about 10 lines from the bottom and leave it there. The cursor will mark the point where the program will start printing when you do a partial print (%).

Return to the Main menu (Clear-M), and get back to the Format menu (F). Put your printer on-line and hit %. The word "print" will appear on the menu until it's done. The last 10 lines of the buffer will be printed out.

(Note: you can abort a partial (%) print the same way you abort a regular print. The only difference is that, afterwords, you must go back into the text and delete the End Mark that was set. This also holds if you set the End Mark to do a partial print and then accidentally hit P instead.)

62 FORMATTING

So far, in the two rounds of printing you've done, things have come out pretty much as they've appeared on the screen (lines that are 51 characters wide). That format has been adequate for writing and editing, but most typewritten manuscripts have 60 or 70 character lines, are often double spaced, and centered on the page.

The format parameters (which occupy most of the Format menu) allow you to control very precisely the way your text will appear in the final printed copy.

63 SPACING

In the Format menu, hit S. The BASIC cursor will appear beside the "Spacing" parameter. This is the line spacing and it is initially set to 1 (i.e. single-space). What you've printed out before has all been single-spaced. So type in the number 2 and hit ENTER. This sets the printout for double-spacing (you can set it to 3, 4, 5 etc. if you want). Now hit P and watch a few double spaced lines get printed out. Abort when you've had enough. (A linespace in the text will only be spaced once, even if multiple spacing is set.)

64 SIDE MARGINS

Now set the left Margin. Hit M and type in 6 then ENTER. Initially the left margin was set to 0 which meant that the printer started each line as far left as it could go. By setting this value to 6, you'll have the printer space over 6 spaces at the beginning of each line, giving you a significantly wider margin.

The value of M can be any number from 0 to 127, but on an 80-column printer, if the sum of the left Margin and Chars per Line is greater than 79, a print line overflow may result. The symptoms of this are broken lines and top and bottom margins and page numbers falling in the wrong place.

To change the actual width of the printed line, hit C. This is the "Characters per Line" parameter. Type in 63 and hit EN-

TER. There will be a slight pause as the text is rearranged into "text lines" that are as close to 63 characters long as they can get. (On the screen, in 51 column mode, non-overflow text lines are no greater than 51 characters long.) The word "Queue" will disappear from the menu, and will reappear when the realignment is done, and the menu will appear to flash once, quickly.

Since you've set the left margin to 6 and the line width to 63, the right margin will, of necessity, be 11 spaces from the far right (assuming an 80-column printer). So M and C are used to set the side margins of the printed page. Remember, though, that if M + C is greater than 79, you may get printer overflows on some lines.

65 UPPER/BOTTOM MARGINS

In a similar fashion you should set the Upper (U), and Bottom (B) margins. These values are both initialized to 4 (that's why both times you printed above, it did 4 line feeds before it started doing the actual text). You could leave them at these values (which gives about 2/3 inch at the top and bottom of the page) or increase one or both. Try 5 and 5. This'll leave about an inch on all sides except the right.

The Lines per page (L) parameter also plays a key role in determining the top and bottom margins and how the printer pages. For standard 8 1/2 X 11 printer paper and standard 1/6 inch line spacing, L is set to 66. Since it's initialized to that value by the program, you generally don't have to set it. (If you're using non-standard paper and/or non-standard line spacing then you will have to determine this number yourself by measuring or counting the # of line feeds it takes to span the full page.)

Note: For the page breaks to come where you want them (i.e. before and after the perforation of the computer paper), it's important that you start printing with the paper set at just the right point -- with the print-head of the printer somewhere at or slightly above the top of page.

You can quickly work this out by trial and error by setting Upper Margin (U) to 0 and doing a print (P) and then abort (BREAK) after a line or 2 has printed. If the top line of text is on the very first available line after the perforation, then use this position every time you start a print.

66 NUMBER PAGES

You're ready to print the formatted text now, but there's one more touch you can add simply. At the upper right of the menu, is the Number Pages command. If this is set to zero (as it is initially), then pages will not be numbered as they are printed. If this is set to any number other than zero, then pages will be numbered consecutively, beginning with whatever number you give it (usually 1, but, if you're writing a long novel, numbers upto 32700 can be handled.) Set it to 1, and reset the Spacing parameter to 1 if you'd like to waste less paper.

The page number is always printed 3 lines up from the bottom of the page. For this reason, if you want pages numbered, the Bottom Margin parameter must be set to a value of 3 or greater.

67 WHERE -- PAGE NUMBER POSITION

The WHERE parameter allows you to position the number where you want on its line. To center it, for example, take one half the Chars/Line, add the Left Margin value, and set WHERE to that (in the usual fashion). For the present example (Margin=6, Chars/Line = 63), set WHERE to 37 (63/2 + 6).

(The page number may also be put at the top of the page and/or included in the header. This will be discussed in section 89 below.)

Now move the paper in the printer so the perforation is at the printhead (as described above) and hit P. Let the printing run its course this time so you can see the page break and page numbering. Notice the difference between the way the page looks now and the way it looked without formatting in the first two examples above.

68 ONE PAGE -- FRICTION FEED

If you have a friction feed printer and use single sheets (as opposed to perforated fan-fold computer paper), you will want to use the ONE PAGE command (O). This is initially set to zero, meaning that the printer will not wait at the bottom of a page. If ONE PAGE is set to 1 (or any non zero number less than 128) then each time the printer gets to the bottom of the page, it will pause and the flashing cursor will appear next to the Print command. Simply hit BREAK or ENTER after you've set the next sheet of paper in place, and the printing will start again for the next page, and halt again at the bottom.

The ONE PAGE command can sometimes be useful even if you're not using a friction feed printer. You might simply want to stop and look over each page before you continue printing.

69 DIRECT OUTPUT

If you have an intelligent printer that uses control codes to change various internal states or perform certain functions, the Direct Command (D), will allow you to send any sequence of numbers (less than 256) directly to your printer. The printer must, of course, be on-line when you do this.

As a quick demonstration that any printer can use, first, hit D. When the cursor appears next to the word Direct, type in the number 10 and hit ENTER. Your printer (regardless of its name) should do a line feed. (If nothing happens, type in 13 and hit ENTER. If nothing happens make sure your printer is on line and type in a few more 10s.) Notice that each time you enter a number, the flashing cursor returns to the right of the word Direct, and waits for another number. Hitting BREAK, will get you out of this mode and back to the command level of the Format menu.

The number you type into "Direct" is a decimal number, and though you can type in as many as you want, you must enter them one at a time (I.e. each value followed by ENTER). The Direct command will keep looking for numbers to send out to the printer until you hit BREAK. Then you'll be returned to the command level of the menu.

70 TYPEWRITER

The "Typewriter" feature works in a similar fashion. Hit "T". The screen will blank with the flashing Basic cursor at the upper left. With your printer on-line, type in a line ("Now is the time for all etc."). The whole line will be printed out as soon as you hit ENTER to end the line. This means that you can edit the line at any point before you hit ENTER (see 1.3 in the ref. manual for the BASIC line editing conventions that apply here).

After each line is entered, the cursor moves to the next line on the screen and waits for you to type in another line. To exit this mode, simply hit the BREAK key and you'll be returned to the Format Menu.

71 CHAINING FILES: CASSETTE

In a 32K system, there's room in memory for about 8-9 manuscript pages. This will be enough for many applications, and, in those cases, the entire text can be contained in one cassette file.

For situations where the amount of text exceeds 1 file, the Queue feature (also called chaining or auto-link) is provided. It allows for any number of files to be loaded from tape and printed without user intervention. The files must first be stored in the proper sequence on tape. You might, for example, have a sequence of files like:

CHPT1A, CHPT1B, CHPT1C, CHPT2, CHPT3, CHPT4A, CHPT4B, CHPT5

To get some more practice using the editor and to create some files to demonstrate the chaining feature, go to the Main menu. Hit N, respond with Y (ENTER) to the query, and a fresh buffer will be started. Now, type in 4 paragraphs of anything and clean it up using the editor. Leave a linespace between each paragraph.

Then, start with the second paragraph, and use the %Save feature to save each paragraph in a separate file on a blank cassette. Save them in the proper order on the tape, so you now have 3 new files there. Then rewind the tape to the start.

Go back to the editor and use the Block Delete feature to get rid of paragraphs 2, 3 and 4, so only the first paragraph remains.

Hit Clear-M to get back to the Main menu, and hit F to get back to the Format menu.

Put the cassette recorder on Play and put your printer on-line, and at the top of the page. Set the Queue parameter to 3. This means that 3 consecutive files will be read in and printed, one at a time. The current contents of the text buffer will be printed first, before the program starts going to the tape for text.

Set the upper margin (U) to 4, then hit P. The program will do 4 line feeds for the upper margin of the page. It will then print whatever's in the text buffer. When it's done with that, normally it would terminate.

Since, however, the Queue parameter is set, it will go instead to read in the next file from tape. Then it prints that, reads in the next file, prints that, and so on, until it's read in and printed 3 files. It will do as many files as you tell it to do when you set the Queue parameter. In this case, after it's read in and printed the third file, it'll return to the menu. Your 4 paragraphs have been printed out, and this method will similarly work for 400 pages. Don't let anybody tell you you can't do "serious" word processing with a cassette based system.

At any point during all this, you may abort the printing in the usual manner (see 60) and get back to the menu or you may abort the file readin (see 32). Both return you cleanly to the menu, but you should set the Queue value back to zero unless you want to do chain printing immediately again. Be aware that if Queue is set to a non-zero number and the recorder is not on or there are no files left on the tape, that, after a print, the program will not return the user to menu level. It will be trying to find the next file on the tape to print. If you get hung up after a print and the Queue value is, in fact, not zero, turn on the recorder and intentionally abort a file (see appendix B, section 3).

Also note that cassette chain printing will not work on Telewriter-64 ASCII format files.

72 CHAIN PRINTING -- DISK

File chaining is a little more flexible with disk than it is with cassette (see 5.18 in Ref Man). To print out a document which is contained in several disk files, you would do the following:

Assume you have a 40 page document to print and it takes up 6 separate files on disk: DOC1/BIN, DOC2/BIN, DOC3/BIN, DOC4/BIN, DOC5/BIN, and DOC6/BIN.

By using the ^Q embedded command (^ is generated by hitting CLEAR-period -- see section 82 and following, below), you effectively tell the print routine to read in and print the file whose name follows the Q, once it's finished printing the current contents of the buffer (upto the ^Q line).

Thus, in the example given, you would start with DOC1/BIN in the text buffer. At the end of the buffer, you would put:

^Q DOC2/BIN:1

This will cause the file DOC2/BIN to be loaded and printed from drive 1 after the text currently in the buffer (DOC1/BIN) has finished printing. DOC2/BIN:1, must, in turn, end with:

^Q DOC3/BIN:1

so that the third file will be read in and printed once DOC2 is done -- and so on. DOC6, will not have a ^Q command at the end because it is the last file to be printed in the document.

The filename that follows Q must be separated from the Q by a space, and the extension must be specified. The drive number need not be specified, but, if it's not, the current default drive will be searched for the file. Including the drive number will enable you to chain print one document from files stored on different drives.

Note that the chain will be broken and the printing stopped if the file is not found or an I/O error is encountered. In these instances, you will be returned to the editor with a blank screen. Hitting CLEAR Up arrow will reveal the file last read in and printed before the error was encountered.

The restrictions on chain printing disk files are: 1) they cannot be files that have been partially saved (with % SAVE). They will work OK if they have been partially saved starting from the very top of the buffer, but not otherwise. 2) You can't mix files from 32K and 64K systems, or from format-swapping and non format-swapping systems. 3) You can't properly chain print files from one size system on another.

These restrictions may be easily overcome, however, by taking all files to be printed in the chain, reading them in one at a time, and then re-saving them exactly as is. If they are all re-read in and saved with the size system that will print them, there should be no problem.

To avoid problems, if you are planning to chain certain files together, and you do do % SAVEs for some of them, indicate this somehow in the filename -- put the partial saves in reverse video or else stick a % sign in the extension or at the end of the name. Then, before chain printing, simply read in each one of these files and resave.

72A CHAIN PRINTING -- DISK & CASSETTE

Three things: 1) When you are saving files that will be chain printed, always end each file at a paragraph boundary. This is important only if you are right justifying the text. By convention, the last line of a text buffer will not be right justified. This means that, if the next file simply began with the next line in the same paragraph, on printout, you'd have a paragraph with an unjustified line in the middle.

2) If you are using headers (see 89 below), the header definition line must appear at the top of each file in the chain, and it must appear at the top of the first buffer printed. If the print starts with an empty buffer, then the header will not start until page 3.

3) After a chain print in either system, the screen may be slightly garbaged on return. Doing a Shift Down cursor (scrolling the text up) will get rid of the garbage, as will returning to Basic and then back to Telewriter-64.

73 EPS/OKI/LF

The EPS/OKI/LF parameter in the Format menu is used to either select Epson Special features (1), indicate Graftrax present on the MX-80 (2), or indicate special carriage return/line-feed handling (4,5). Epson special features include underlining without Graftrax and menu Font control. These will be explained in the Reference Manual (5.20, 5.7)

Different printers handle Carriage Returns differently with respect to auto line-feeds. There appear to be 3 main classes:

1) Okidata and Centronics printers can be set (via dip switches) to provide a line feed with each carriage return, but consecutive carriage returns (not preceded by any other character) will not generate auto line feeds.
2) Terminet 300s, TI Silents, Teletypes, and some Daisy Wheels, (and others) require that the software send explicit line feeds with each carriage return.
3) LP's, DMP's, MX-80s, Prowriters, NECs, and most others, can be set to provide a line feed with any and all CRs.

Setting EPS/OKI/LF to 4 will handle the first category above. Setting it to 5 will handle the second. For an MX-80 without Graftrax, the setting should be 1. With Graftrax, it should be 2. For all other class 3 printers -- 0.

74 RIGHT JUSTIFICATION

To right justify your text on printout, simply set the "Justify" parameter in the Format menu to 1 before you print. If your printer can handle it, you can do boldface, underlining, subscript, super script, foreign characters, mathematical symbols, etc., all within perfectly justified text (see section 87 on printable embedded codes).

Changing font size within a line will destroy the justification for that line. If you really want to use different sized print within a single line of text, you're better off setting justify to zero and using rag right margins or trying to "hand-justify" that particular line. There will, of course, be no problem in using enlarged headings with justified text as long as different sized fonts are not mixed in a line that, itself, must be justified. Isolated lines (like headings), and lines at the ends of paragraphs, don't get justified, because they are generally too short to not look absurd after justification.

75 DYNAMIC ENABLE/DISABLE JUSTIFICATION

Right justification is nice, but sometimes you want sections of your document left alone. This is especially true of lists and columns and other things that are better formatted by hand. By the same token, you usually want these sections left alone by the align command as well.

The embedded code

^;

placed around a block of text, will prevent both alignment and justification from operating on that block. The embedded command syntax will be explained more fully in sections 80 & 81 below. For now, as an example, take the list from section 74 and surround it with the non-align code so the screen looks like this:

```
^;
1 Carrots
2 Apples
3 Oranges
4 Artichoke
5 Cauliflower
^;
```

(the ^ is generated by hitting CLEAR-. (CLEAR-period), and the ^; command should be put on its own line, as indicated above)

Now hit Clear-A and notice that the list isn't aligned. Delete the two non-align codes and hit CLEAR-A again. This time the list will be combined into 1 line by the alignment routine.

Note that the codes must be used in pairs. The first one turns the alignment/justification off, and the second one turns them back on. If there is no second one, the remainder of the text will go unaligned when you hit CLEAR-A, and unjustified when you print.

76 NON-BREAKABLE SPACE

To generate a backslash character on the screen, hit CLEAR-, (CLEAR-comma).

For use with right justified text, the backslash serves as a non-breakable space so you can prevent certain words from being separated when the line is padded with spaces. Though it appears on the screen as a backslash, it prints as a space on printout as long as the line it's in gets justified.

When text is not justified, the backslash prints as a backslash. This is provided so Telewriter-64 may be used to write "C" programs, which require a backslash character in their I/O syntax, and are never rigt justified. (See section 89 below for still another use of the backslash character.)

77 ON-SCREEN PRINT FORMATTING -- HIGH DENSITY DISPLAYS

Telewriter-64 provides you with a number of features that allow you to see, on-screen, most of the relevant aspects of your final printed output.

You can switch between 3 display formats (51 X 24, 64 X 24 and 85 X 24) with a single command. Hit CLEAR-: to go into the 64 column mode. Hit CLEAR-2 to go into the 85 column mode, and hit CLEAR-0 (zero) to return to the 51 column mode.

When you switch modes, the screen will rewrite and text lines will be realigned according to the current line width. Get a good chunk of text on screen and try switching between the 3 modes to see what each looks like.

Though less easily readable than the 51 column mode, the 2 high density modes are ideal for on-screen formatting, doing columns, and hyphenating, since they let you see the full width of your page layout at one time. And, since these modes give you all the editing features of the 51X24 mode, you can make the change as soon as you see the need. You can also switch instantly back to the 51 column mode to do further editing and then immediately back to the 85 or 64 column mode to see what the page layout now looks like.

When you switch between display modes, the cursor and the display stay put so you don't lose your place and so you can move quickly between viewing and editing. Be aware, however, that when you switch display modes, the text is automatically re-aligned to the current line width (Chars/Line). So, if you have text you don't want "aligned" (BASIC and assembly programs are outstanding examples of this), you should protect it by using the ^; embedded code (see section 75), or by setting "Chars/Line" to 0 in the Main Menu (see 79 below - especially paragraph 5) before doing any mode switching.

Also be aware that, for example, if your line width (i.e. the Chars/Line parameter in the Main menu -- see 79) is 63 in the 64 column mode, it'll still be 63 if you suddenly switch to the 51 column mode. This means that now, when you do an Align (CLEAR-A), most of the lines on the screen will be overflow lines approaching 63 characters in length. Thus, if you plan to stay in the 51 column mode for any length of time, you should set Chars/Line to 50, to optimize editor functioning.

78 TWO CHARACTER SETS

Notice that for the 2 high density modes, a different character set is used. This second set is only 3 pixels wide as opposed to the standard Telewriter character set which is 4 wide. When you switch to 64 or 85 mode, the 2nd character set is automatically used, and the 1st one is used when you switch back to 51.

You can, however, select either of the 2 character sets in any display mode. This is a matter of preference. In the 64 mode,

character set 1 is larger and more readable, but there is no space between letters as there is with character set 2. In the 85 mode, character set 1 characters are clipped at the right edge but they are larger than set 2.

Hitting CLEAR-; "toggles" the character set. That is, this command changes the character set used to the one not currently in effect. Thus, you can switch back and forth easily to see which you prefer.

79 LINE WIDTH

You can change the line width displayed on the screen at any time. It can be set to any value from 0 to 127 (but 84 is the maximum that can be fit on one screen line at one time).

To set line width, go to the Main Menu and hit "C" (for Characters per Line). Now type in the desired value and hit ENTER. The old number is destroyed as you type in the new one, and, when you hit ENTER, there is a brief delay, and you are returned to the editor screen, now fully rewritten to reflect the new line width.

Generally, during writing, Characters per line should be set to 50 and the 51 column display mode should be used. This is far and away the most readable format, but, if you're comfortable with the 64 column mode for writing, then set Chars/lin to 63, and use that mode.

The line width setting is useful mainly when formatting text prior to printing. At this point, for viewing purposes, you should use a display mode which is at least one greater than your desired printout line width.

Keep in mind the distinction between display mode and line width. The "display mode" number -- 51/64/85 -- indicates the maximum number of characters that can fit on one line on the screen. The "Characters per line" parameter (line width) indicates the actual length of the line of text as it will be printed. If Chars/line exceeds or equals the display mode line size, then the lines may "overflow" on the screen.

Thus, when viewing, if line width is greater than 50 and less than 64, use the 64 column mode. If it's greater than 63 and less than 85, use the 85 column mode. This will allow you to see the full width of all your lines on the screen at one time.

During normal typing, the lines will extend to the width of whichever display mode you're using (51, 64, or 85). But, whenever the Align command (Clear-A) is given, the width of the lines will be adjusted to the current value of Chars/Line (line width), so you can instantly see your text at the width you've set. Because of this, CLEAR-A will only eliminate overflow lines if Chars/Lin is set to a value lower than the width of the display mode used (51, 64, 85).

Note that if the line width is set to zero, then no alignment will take place. This is quite useful if you are writing Basic

or assembly programs, where an accidental align command can totally destroy your work by running lines together.

80 HYPHENATION

By hyphenating, you can get better looking right justified text. Hyphenating puts more letters on a line which means that fewer spaces need to be inserted. It also packs more text onto a single page. (Take a look at a newspaper or magazine or book, and notice how many lines are hyphenated.) The Telewriter-64 high density screen displays combined with the Align command and the global search and replace feature, make hyphenation simple.

Hyphenating is the last thing you should do with your text (with the exception of finding page breaks -- see 81). Finish and proof-read it first, then hyphenate (if you wish), and then print.

When you're ready to hyphenate, set the Characters Per line parameter in the Main Menu to your desired printed line width. You will be returned to the editor. Now select a display mode that is wider than your line width. (I.e., Hit Clear-1 if it's 63 or under, hit Clear-2 if it's over 63.)

The text lines as they now appear on the screen are exactly the lines as they will be printed (minus justification's padding and special printer features).

Scan down the right margin and look for lines that are short. Your goal is to put more into these lines so they will have to be padded less. When you've found one (usually not too hard), look at the next line and see if the first word in it can be hyphenated (i.e. if it has more than one syllable). If there is, move the cursor to the hyphenation point (a syllable boundary that looks like it will fit on the preceding line).

Now type a hyphen (-), followed by a space. Then hit clear-A. If there is room for the hyphenated fragment, it will jump up to fill out the preceding line as the text is aligned. If it doesn't, simply delete the "hyphen space" combination by hitting CLEAR-BREAK twice. See if there is an earlier place in the word to insert a hyphen and, if so, repeat the procedure.

Or, if it does work, and the preceding line still looks too short, see if there is a later place in the word to hyphenate and try that. Hitting CLEAR-A will always show you what fits in your established margins and what won't.

Remember, you should only hyphenate once you've decided on the line width you want to print at. You should also work from top to bottom when finding hyphenation points, because each hyphenated line changes the arrangement of all the text that follows it (in its own paragraph). If you do make changes after hyphenation, you should do a global search and replace (delete) for the "hyphen space" combination. If you find some that were former end-of-line hyphens but are now sitting in the middle of a line, simply delete them by hitting CLEAR-R. This search

should be begun at the top of the text and is something you should probably do religiously right after you finish hyphenating, and right before you print.

81 PAGE FINDER -- VERTICAL TAB

Before you print, it is often useful to know where the page breaks will fall. This helps you avoid having certain lines occur at inappropriate places on the page (e.g., headings falling at the last line of a page, or the last line of a paragraph falling at the top of the page).

To find page breaks, first, go into the Format Menu and set the main print parameters you will be using: Upper Margin, Bottom Margin, Spacing, Lines per page, and Characters per line. These are the values the Page finder needs to properly determine page breaks. If you change these between the time you scan for page breaks and the time you print, the results will, of course, not match. (See section 89 for special considerations when using headers.)

Now return to the editor and move the cursor to the very top of the text. Hit CLEAR-V. The screen will rewrite, and the cursor will be sitting at the beginning of a line at the center of the screen. This is the first line of the second printed page. Hit Clear-V again and the screen will rewrite, and the cursor will be sitting at what is the first line of printed page #3. You can continue this way through the rest of the text.

Once you're sitting at the page break, you can instantly see the surrounding lines. Say the line just before the cursor is a heading. This means it'll be the last line of the preceding page, rather than at the "head" of its section. But, since you know this in advance of printing, you can simply insert a CR before the heading (if it were 2 lines before the top of the next page, you'd insert 2 CRs). This will push it down so that it now falls at the top of the next page, where it belongs.

Verify this change by jumping back to the top (or to any previous page break), and paging forward with successive CLEAR-Vs until you're at the change. The cursor indicating top of page, should now be at the heading. If it's not, add or delete CRs to get it there. (Depending on the aesthetics of the particular page, it may be more practical to delete CRs so the heading is followed by a few lines at the page bottom.) In this way, you can cause page breaks to fall whereever you wish (though sometimes you have to do more than just add or delete CRs -- sometimes you need to change whole paragraps to get the page breaks where you want them.)

Keep in mind, though, that, if changes are made, you should re-scan the text with CLEAR-V to make sure that any change in page breaks is not detrimental. Remember that the cursor must always start at the very top of the text, or at page breaks found by CLEAR-V, in order to find the next page break. It must also start at the beginning of the line. CLEAR-V jumps the cursor exactly one page distance from whereever it starts.

Scanning for and verifying page breaks should be the very last thing you do before printing, even after hyphenation. In fact, you only really need to do it when you're printing out the final draft.

Note that when the page finder encounters an embedded format command that changes the Chars/Line parameter (^C -- see 82 below), it will actually set Chars Per Line to that new value, and realign all subsequent text in accord with it. Thus, the page finder can show you dynamic changes in the page width as they will occur during printing.

When it does set Chars per line to the new value, that value will hold until CLEAR-V encounters the next embedded "C" command. If you cursored or paged down after this, you'd find all lines following the change will be set to the new value, (as well as Chars/Lin in the Main menu). This width will be wrong for all lines following any subsequent embedded "^C" command, but will be set right as CLEAR-V "passes" over those. For this reason, the line widths preceding the cursor (after a CLEAR-V is done) will all be correct, while the ones following, may not be. For this reason as well, starting at a page break in the middle of the file may give erroneous results. A "final" scan should always be done, starting at the very top of the text. All scans should also be started with the cursor at the left-margin of the line it's on.

If you want a permanent record of where the page breaks are, you can optionally mark them with a non-printing comment line (84). This line will not be counted when the page finder finds breaks and it will not effect the printout.

The recipe for this is: Set up as described above and hit CLEAR-V. Bang -- the cursor is sitting at the first line of printed page 2. Now hit ENTER so the line is pushed down one. Then hit up arrow to move the cursor up 1, and then type

^T P1

This line now marks the end of page 1 and the begining of page 2. If you go back to the top of text and hit CLEAR-V now, the cursor will jump to the ^T line. Since it's a non-printing comment line, however, it is not counted or printed, so the following line is (still) the actual top of page. If you mark your text in this way, it will not effect anything in the printing and you will now be able to scan through the text and always know which printed page you're on. Of course, if you make changes to the text after the page breaks are marked, they may no longer be valid and you should re-scan the text with CLEAR-V and change any that have been moved. If the vestigial ^T page break markers created in this way are not removed, they may leave short lines in the final printout as alignment will not work around them. Also note that if only 1 line space separates 2 paragraphs (which you're delimiting with line spaces rather than by indentation), the ^T line should not be inserted into it. This would effectively eliminate the paragraph break and the line before the ^T will be justified when it shouldn't be. So always insert the ^T on a new line.

If you are chain printing files and wish to use the CLEAR-V command across the entire document, the recommendation is this: End each file at a page break and make sure the page break is also a paragraph break (this may take a little re-manipulation of your text, but usually it's just a matter of throwing in a few line spaces or cutting some verbosity out of a paragraph or two). The paragraph break is necessary because the last line of a text buffer will not be justified. If it should fall in the middle of a paragraph, it will ruin its appearance. When using this approach, make sure that the last CLEAR-V gets you to the ^Q FILENAME/EXT line if you are disk chaining or to the End of Text marker if you are chaining from cassette. Partial saves can be used as long as they are started from the very top of the text buffer. If this is done, make sure the last CLEAR-V gets all the way to the End Marker. (See 89 below for special considerations when using CLEAR-V across chained files that have headers.)

82 EMBEDDED FORMAT CODES

The Format menu allows you to flexibly specify the appearance of your text when it's printed. One short-coming of this "Menu-driven" approach, is that, once the parameters are set, they stay set for the duration of the printing.

But there are times when you'd like to indent certain passages more than others, or suddenly start a new page, or change font (if your printer has that capability). One way to do this would be to use the partial print feature (section 61 above) and have the printing stop at each point where you want to change some parameters.

This can be anything from inconvenient to offensive to impossible. Embedded format codes provide a way to change format parameters during printing. They are very similar to the parameters set in the Format menu, but, as their name implies, they are "embedded" in the text.

During printing, whenever these special codes are encountered, they will not be printed, but instead, will instruct the program to change the appropriate format parameters. These new values will hold until they are changed again by another embedded format code.

The key to embedded format commands is the embedded command marker. Hold CLEAR down and hit the period (.) key. An up arrow character (^) will appear on the screen. That's the embedded format marker. A line beginning with this character will not be treated as normal text on printout, instead, it will be treated as a command. Usually a command to change a print parameter.

For example. Say you have a long quote and you want to indent it to set it off. You need to change the left margin and the line width (temporarily). The embedded command:

^M15 C40

will change the margin to 15 and the Chars/line (line width) to 40. When the quote is over and you want to return to your regular margins (assume they were Left Margin=6 and Chars/Line=63) just use:

^M6 C63

Embedded format commands must be on their own line. The parameter to be changed is represented by the same key letter used in the Format menu. The parameters which can be changed dynamically with the embedded commands are:

M = Left margin S = Spacing C =Chars/Line
U = Upper margin B = Bottom Margin L= Lines/page

In the format command, each Command letter (M,S,U,C,B,L) must be followed immediately by a number indicating the new value. No intervening spaces are allowed. There must, however, be one or more spaces between individual commands. The six commands listed above may all be put on the same line or they may each be put in separate lines. You only need to specify the ones you want to change. Any parameter not set by an embedded command, is taken care of by the values you've set in the menu. Thus

^M6 C63 S1

would set Margin to 6, Line width to 63 and Spacing to 1, as would:

^M6
^C63
^S1

The lines may be "stacked" as indicated in this last example. That is, placed one right after the other. However, no embedded command line may exceed a single line. In general an embedded command line should be restricted to one line on the screen. If it does exceed a screen line (i.e. if it's an overflow line), however, you must make sure there are no hidden CRs in it -- that is, the embedded control line must always be no more than one text line. If you exceed a screen line, make sure that word wrap doesn't put in a CR behind your back. (See Ref. Man. Section 2.6 for more on "screen lines" vs. "text lines".)

An embedded command line should begin with the ^ character, and **should not** be preceded by any spaces. Alignment will have no effect on embedded command lines, and will not combine them with other lines.

(Note that changing the bottom margin does not take effect until the next page after the command is given. Upper, Bottom and Lines per page are rarely changed dynamically. If they are used, the Page Finder (section 81 above) will not work properly.)

83 OTHER EMBEDDED COMMANDS

The following features all use the embedded command syntax, but they cannot be combined on the same line the way the o above can, and they deal with functions other than changing format parameters. They can, however, be stacked and freely intermixed with other embedded command lines. The remaining embedded commands will be discussed briefly here, and more thoroughly in the reference manual.

84 COMMENT LINES

You can now put lines in your text files that will not be printed -- i.e. "comment lines". These can be especially useful for telling you what your other embedded control commands mean (see 86 below). You can save commented "glossaries" -- collections of control codes and their definitions -- that can then be loaded independently at the start of each file, freeing you from the need to remember complex code sequences everytime you want to do something jazzy with your printer.

The ^T code is used at the beginning of a line to mark a line that will not be printed. The only time this doesn't work is for the very first line of the text file. In that spot, ^T is treated as a tab stop definition line (section 26).

^T, used as a comment line marker, is only good for one line at a time. Each comment line must begin with it's own ^T, and the ^T must be followed by a space. An example would be:

```
^T Left Margin = 6, Chr/lin = 63, Spacing = 1
^M6 C63 S1
```

As described in section 81 above, ^T comment lines are also quite useful for marking page breaks once they are found by the Page Finder.

85 FLUSH LEFT LINES

If a ^T command line comes at the very top of the text, it sets tabs. If it comes anywhere else and the ^T is followed by a space and then text, it is a non-printing comment line. However, if the ^T is not followed by a space, then the line will be printed, but it will not be subject to alignment. That is, when alignment happens, this single line (from the ^T to the CR that ends the line) will not be combined with either the preceding line or the subsequent line when the text is aligned.

This is useful for things like letter headings where you have a few short lines all of which need to be kept flush left on the printed page.

Keep in mind, then, the big difference a space makes:

^T double-strike condensed enlarged print

will not be printed.

^TMr. Arthur Rimbaud
^T72 Market St.
^TVenice, Ca.

will be printed, but won't be effected by alignment. The disable align/justify embedded code (^; - section 75) will also achieve the same effect (on a block of text rather than on individual lines). It is frequently prefered, as ^T lines will be justified if they are greater than half the current line width (i.e. the value Chars/Lin is set to). This is generally not desirable.

86 EMBEDDED CONTROL CODES

Many of today's printers have "intelligent features". They can change type style and size, underline, superscript, subscript and so on. To access these, the printer must be sent special codes. Telewriter-64's "Embedded control code" feature allows you to send any sequence of special codes to the printer, dynamically, at any time during printing.

There are two parts to embedded control codes:

1) First, there is the definition line (placed preferably at or near the start of the text). This uses the embedded command character ^ to begin the line, followed by the letter D, followed by a number from 1 to 9 (with no spaces separating the D and the number). The embedded code definition line:

^D1 27 88

says: "Define the 'special character' 1 to send the codes 27 and 88 when encountered in the text during printing." 2) The code is now defined, but it takes a "special character" to actually send the code. The "special characters" are the numbers 1-9 but with a significant difference. So try this now. Hit CLEAR and 1 at the same time and look at the screen. Notice that a miniature 1 (¹) appears.

Try this with the other numbers from 1-9. These miniature numbers are the infamous "special characters." On the screen, they look almost like superscripts, but they are not. They will not be printed as either numbers or superscripts. Instead, when the print routine encounters them, it will look at the embedded control definition line for the corresponding number (in this case 1), and send the numbers that appear on the line (in this case, 27 and 88 -- or ESCAPE X -- a sequence that starts underlining on a C.ITOH Prowriter). Thus, if your screen looks like this:

^D1 27 88
^D2 27 89

this is some dummy text. This is some more dummy text.

more dummy text demonstrating ¹embedded control² codes.

it will print as follows, on a Prowriter:

this is some dummy text. This is some more dummy text.

more dummy text demonstrating embedded control codes.

(27 89 is the C.Itoh code to stop underlining.) Printer control codes should be described for you in the manual accompanying your printer. It is upto you to be certain that you're using the proper codes to acheive the desired effects.

87 PRINTABLE CONTROL CODES, FOREIGN SYMBOLS

Embedded control codes are usually just that -- non-printing codes that get sent to the printer to control a function. When justifying text, the program does not count these codes because they do not put additional characters on the line. But embedded control codes can also be used to print foreign, math, and other strange symbols which many printers provide.

If you wish to print these special symbols in justified text, you need to let the program know they are printing codes rather than control codes. To do this, simply change the D in the definition line to DP, so that, where normally you would use the form:

^D1 94

now you would use the form

^DP1 94

The above line defines the embedded code 1 to send an ASCII 94, (up arrow code). This form must be used if justified lines are to print properly with printing control codes in them. In general, when you use printing control codes in justified text, you should only define 1 character per code.

88 CENTERING

To center a line automatically on printout, simply precede it with ^*. Restrictions on the length of the line are as described in the last paragraph of section 82 above. Centering lines whose characters are a different size than the body of the text, presents a few problems which are discussed in the Reference Manual section 6.11.

89 HEADERS

To cause a header to print at the top of each page beginning with page 2, use the form:

^H3 Jan 3, 1983 Telewriter Tutorial ^

The header definition line must begin and end with the ^. Otherwise, you'll get an error. The letter H after the first ^ indicates Header, and the number immediately after it sets the number of linefeeds after each header before the actual body of the page begins.

The margin at the very top of each page (before the header is printed) is determined by the value of "Upper Margin" in the Format menu (usually 1 or 2 when headers are used). Then the header is printed, and then (in the example just given) 3 linefeeds are sent and the main body of the text begins.

This "header definition line" must occur at the top of text. It can occur in a stack of embedded commands at the top of the text (see 6.3), but this stack must be at the very top. (So, among other things, a tab stop definition line could precede the header definition line as long as they were in the same stack.) No linespace must separate the header, or the stack it's in, from the very top of the text. Otherwise, the header will not start til page 3.

If you follow the H with a Plus sign (+) you will get an immediate header. This means the header will start on the very first page instead of waiting til the next. You might use this if you are printing a document in sections where the first printed page is not necessarily the first page of a chapter or the document itself. You can also use this form if, for some reason, your header definition is not at the very beginning of your text.

The Header is not effected by the Telewriter-set margin, so you must control its positioning explicitly with spaces in the definition line.

The page number can be included in the header simply by putting a backslash character in the header where the page number is to be printed (the backslash character is produced by hitting CLEAR-, (CLEAR comma)):

```
^H3   Telewriter Tutorial                    page \ ^
```

If you use this option, no page number will be printed at the bottom of the page, but the Number Pages parameter in the Format menu must be set to the first page value that you want printed (usually 2, if you're starting at the beginning).

In the same manner, you can select to print page numbers anywhere at the top of the page. Just use the header syntax with no text.

```
^H3                          - \ - ^
```

will print the page number centered at the top of the page, with a hypen on each side.

Note that when using the Page Finder feature, if headers are present, an extra step is required: When the page finder is used to find breaks, the Upper margin (in the Format menu) must

be set to the sum of Upper margin + # of linefeeds after header + 1 (for the header line itself). That is, on printout, Upper Margin would be set to 1, indicating that the header is to be printed 1 line down from the top of the page. But, when you're finding page breaks, if H3 is used as in the examples above, Upper margin should be set to 5 (Upper margin (1), plus the header line (1), plus linespaces after header (3)). Then, when you go to print, remember to set Upper margin back to 1 in the Format menu, otherwise the page breaks won't fall where expected (neither will the headers).

If you are chain printing (71, 72) and using Headers, the Header definition must appear at the top of each file in the chain. If you wish to use the Page Finder across these files, then, in each and every file after the first, the Header must be an Immediate Header (see paragraph 5, this section), otherwise the Page Finder will give wrong results.

90 NEW PAGE

Sometimes, during printing, you want something to start a new page. The ^N command, embedded anywhere in the text (on its own line of course), will cause the printer to jump to a new page when it is encountered during printout. If the N is followed by a number, it will start page numbering using that number (e.g. ^N10 will number the new page 10 (when it gets to the bottom) and the next page 11 and so on).

The ^N command can also be used at the very end of the text, to get the program to number the last page.

TELEWRITER-64

Reference Manual

TELEWRITER REFERENCE GUIDE

0. MAJOR STATES. Telewriter-64 is always in 1 of 4 states. It is either in the Editor, or in 1 of 3 menus. Clean returns to and from BASIC are also supported.

1. MENUS

1.1. CONVENTIONS. There are 3 menus: the Main menu, the Format menu, and the Disk I/O Menu (disk version only). The items on the menus are either commands or status information or both. Status information is distinguished from commands on the menu. A menu item is invoked by its first letter (i.e. press "N" in the Main menu to create a "New file"). A menu selection requires only one key (upper and lower case are treated the same) and is not followed by ENTER. When no cursor is flashing, the program is at the top of the menu, waiting for a one-key menu command.

1.2. COMMANDS. There are 2 kinds of menu commands: action commands and parameter-setting commands. Parameter commands require a value to be input from the keyboard. Action commands do their thing instantly or require a string input from the keyboard first.

1.3. MENU INPUT. Menu items that require further input will display the flashing BASIC cursor directly to the right of the item, whenever it's invoked.

The conventions for entering input to a menu item are the same as those for Color BASIC. I.e.:

1) Lowercase characters appear as reverse video

2) Shift-0 toggles between Lowercase mode and Uppercase only mode

3) When keyboard input is required, the flashing BASIC cursor appears

4) The alphanumeric string or numeric value typed in at this point is not entered until the ENTER key is hit

5) Back Arrow deletes the previous typed letter and moves the cursor back 1 place

6) Shift-Back Arrow deletes the entire line typed in

7) BREAK aborts keyboard input (before its been terminated by ENTER) and returns cleanly to the menu without executing the command

1.4. DESTRUCT PROTECTION. Main Menu commands that destroy text in memory (i.e. "New File" and "Read In") are followed by

a query ("Sure?"), so the user cannot accidentally destroy the text. There is only one response to the query that will allow the function to take place -- an upper case Y (or any word beginning with upper case Y, such as "YES"), followed by ENTER.

1.5. ACCESS. The Main menu appears on initialization (after the copyright notice). The editor, the Format menu, the Disk I/O menu (disk version only) and BASIC (cassette version only), are accessed through the Main menu. From the Main menu, E accesses the Editor, F accesses the Format menu, D accesses the Disk menu (disk version only) and B returns to BASIC (cassette version only).

CLEAR-M returns to the "M"ain menu from the Editor. R "R"eturns to the Main menu from the Format menu. E returns to the "E"ditor from the Disk I/O Menu. In the disk version, B provides access to "B"ASIC from the Disk menu. If the non-resident print routines are used (See 4.2, 5.0), R will "R"eturn to the Editor rather than the Main menu.

2. THE EDITOR

2.1. CONVENTIONS. Typing N (New File) or E (Edit) in the Main menu causes a jump to the Editor (if there's text in the buffer, New File will query first with "SURE???.." since it will destroy what's already there -- see 4.1). The display changes from the BASIC lo-resolution screen to a hi-resolution white screen. A black square marks the End-of-Text (referred to as EOT marker or just EOT). The rate of cursor flashing indicates Lowercase mode if it's slow and "Uppercase only" mode if it's fast. Pressing Shift-0 will toggle the flash rate as it toggles between upper and lower case.

The CLEAR key is used as the "Control key" in combination with another key to generate editor commands or special characters. It is pressed simultaneously with the other key or held down slightly before pressing the other key. For the purpose of editor commands, upper and lower case are treated the same.

There are 4 non-standard keys. Left Bracket ("[") is generated by Shift-8, Right Braket ("]") by Shift-Clear. The Tab key is Clear-ENTER. And backslash ("\") is generated by CLEAR-,.

2.2. INSERT. There are no insert, delete, or cursor-movement modes. Any alphanumeric key hit is immediately inserted at the current cursor location. The cursor then advances one position to the right, so it points to the same character in the text as before the insert was made.

The "SPACE" entry at the bottom of the Main menu indicates how much room is left in memory for text. It decrements 1 for each character inserted in the text. When this number reaches zero, any character typed will cause a "NO ROOM" error, and will not be inserted into the text (see 4.3).

2.3. DELETE. The BREAK key deletes the character at the cursor. Clear-BREAK deletes the character before the cursor.

Deleting a "CR" (see 2.6) causes the subsequent text to be pulled up, filling the current line out to the right margin of the display. Often, this will turn the current line into an overflow line (2.7), and following it with Clear-A (2.8) is recommended. When a "CR" is deleted, the screen is also rewritten to keep the text coherent looking.

2.4. CURSOR MOVEMENT. The cursor will go only where there is text. (Upon initialization, with no text in the buffer, the cursor will go nowhere.)

The arrow keys move the cursor in the expected directions. Up Cursor and Down Cursor keep it at the left margin. The Shift key works as a "Repeat" key when held down along with any of the 4 cursor movement keys. The two can be pressed in either order, and holding down one and intermittently pressing the other allows jumping quickly around in the text.

Shift Up Arrow and Shift Down Arrow essentially scroll through the text. Since the cursor only goes where there's text, no scrolling occurs until there's more than 1 page of text on the screen (or at least 24 or more ENTERs).

Clear Right Arrow moves the cursor to the end of the current line, Clear Left Arrow moves it to the beginning. Clear Up Arrow moves to the top of text, Clear Down Arrow, to the bottom.

2.5. WORDWRAP. Wordwrap takes effect whenever a character is inserted at the far right margin of the screen (e.g. the 51st character on the line in the 51 column display mode). Any text lower than this line on the screen is scrolled down, and the last complete word on the line is moved to the next (which has been left clear by the scrolling). The cursor still remains in its current position at the end of the word being typed (section 2.7 has a little more detail).

Clear-D "Disables" Wordwrap, Clear-W ("Wordwrap on") enables it again. The system is initialized with Wordwrap enabled, and, for most standard text, it should be used in this state. (The sudden appearance of words broken at the ends of lines, may mean that Clear-D has been accidentally hit. Clear-W will get Wordwrap back, then Clear-A will readjust the lines on the screen.)

Wordwrap should be disabled when using Telewriter-64 to create BASIC programs or when creating tables using the Tab feature. Wordwrap's hidden "CR" may be destructive in both cases.

Note: For the optimal functioning of Wordwrap, the cursor should be at the end of the line being typed in -- for the following reason:

When characters are being inserted into an existing line of text, Wordwrap occurs only when the cursor exceeds the right screen margin. Upuntil the cursor and the insertion get to the end of the line, the text overflows freely onto the next "screen line" (see 2.7). When Wordwrap occurs under these circumstances (inserting into a line as opposed to inserting

at the end of a new line), it needs to rewrite the screen, instead of scrolling as it normally does. This requires a little extra time and may cause the loss of one or more letters being typed when the Wordwrap occurs.

This slow-down can be avoided by moving the cursor to the point of insertion, hitting ENTER (to push the rest of the line down, out of the way), followed by Left Arrow (to get back to the end of the preceding line), and then typing the insertion. When the insertion is done, Clear-A (see 2.8) can be used to realign the text and fill out the broken lines.

It is also possible for the condition just described to go unnoticed. That is, it can appear on the screen that inserting is happening at the end of a line while, in fact, extra space characters (unseen on the screen) following the cursor, make the program think it's inserting into the middle of a line. Thus, it will rewrite the screen on Wordwrap (as described in the preceding paragraph) slowing typing speed when Wordwrap occurs.

If this happens, hitting Clear Right Arrow will get the cursor to the true end of line. Extra spaces in the line (directly before the cursor now) should be deleted with CLEAR-BREAK. Now, when Wordwrap occurs during typing, the subsequent text on the page will scroll out of the way -- fast enough so that touch typing speeds are not hampered. If response is still slow, see Appendix B.

2.6. TEXT LINES AND "CR". The only unseen character on the screen besides "space" is the "CR" character (carriage return, line feed) used to terminate each text line. This is the character the cursor is "at" when it's sitting at the start of an empty line, or when text is being inserted at the end of a line (see 2.5).

The CR is used by the Up Cursor, Down Cursor, End and Beginning of Line commands, and the Line Kill command (Clear-K, see 2.12) to recognize where a "text line" ends (and another text line begins). A distinction, thus, should be made between a "text line" and a "screen line".

A screen line is one of the 24 lines of the screen display. It can have any number of characters on it from none to a maximum of 51 or 64 or 85 (depending on the display mode used - see 2.21). A text line is a string of characters on the screen ending with a CR. A text line is normally 1 screen line long, but if it overflows due to inserting text (see 2.7, 2.5), a text line may be more than one screen line, and have more than the maximum number of characters for that display mode.

2.7. CONTINUATION LINES. Normally, when new text is typed in, the cursor is sitting at the end of the text line, at the CR. When inserting text takes the cursor to the end of the screen line (the far right end of the screen), Wordwrap shifts the current word down to the next line and leaves a CR at the end of the last line (and the cursor at the end of the new line). This, thus, fits one text line into each screen line.

Under these conditions, every line on the screen will have a CR at the end of it, put there by Wordwrap, and each text line will be 1 screen line long. However, inserting characters into an existing line will eventually cause the end of that line to overflow the right margin of the screen onto the next screen line. This is an overflow or continuation line. It means this one text line is longer than one screen line. It means the CR is more than 51 characters away from the beginning of the text line (in 51 column display mode). A continuation line may be any number of screen lines long (greater than 1, of course).

2.8. ALIGNMENT. The editor can deal with these overflow lines, but they tend to mess up the appearance of the text (e.g. words broken at random letters at the end of a line). They can also be confusing to a new user, and they slow down certain editor functions (scrolling down, Wordwrap). Things go a lot smoother if each text line is no longer than a screen line.

Clear-A, quickly "Aligns" the text on the screen so this is the case. It inserts CRs and fills out lines so no text line overflows the screen margins. Essentially it fits everything back into 51 character lines (in 51 column mode).

This same feature is useful when deletions within lines leave short text lines that chop up sentences. Clear-A puts the sentences back together. Its liberal use is recommended when changes leave short or overflow lines that can confuse the editing.

Because the Align routine is often doing a major rearrangement of text on the screen, it rewrites the display each time it is invoked. When this happens, the cursor stays pointing where it was in the text.

Hyphenated words (like counter-insurgent) are treated as a single word by the Alignment routine, and not broken.

2.9. PARAGRAPHS. The Alignment routine (in its effort to fill out short screen lines and eliminate overflow lines) feels free to move any line that is flush against the left screen margin. However, if a line is preceded by a blank line (i.e. a screen line with only CR on it) or, if a new line begins with one or more spaces, it will not be moved by the Align command.

This means that, to indicate a paragraph, it is only necessary to precede the new paragraph with an empty line (i.e. hitting ENTER twice in succession will always start a new paragraph) or to indent 1 or more spaces at its start. Since this is the way paragraphs are indicated in normal writing, it seems the most natural and efficient way to do it here.

Besides keeping lines from being joined on the screen, a preceding blank line or indentation will also indicate the paragraph breaks when it comes time for formatting and printing.

2.10. ALIGNMENT DEFEAT. To keep lists and other non paragraph-oriented text from being aligned (i.e. short screen lines not filled out), the same single space indentation preceding each line will do the trick. But, for situations where this is not appropriate (e.g. a flush left line is desired), two other methods are provided to disable alignment.

The ^; embedded command (see 6.1 and 6.7 for syntax requirements), placed in pairs, around a block of text, will cause the alignment routine to completely ignore that block.

The embedded command ^T, followed immediately by a line of text, and with no space between the T and the line, will cause the alignment routine to completely ignore that text line until the next CR. If this line exceeds the width of the screen, care must be take to avoid a word wrap-inserted CR that would terminate the line before expected. ^T is intended mainly for short lines. If the line exceeds half the value of Chars/Lin, it will be right justified on printout -- something that's probably not desired.

2.11. BLOCK DELETE. A block of text to be deleted is indicated in the following way. First, the cursor is moved to the end of the block and Clear-E (End block marker) is hit. A bracket pointing left will appear, marking the end of the block (see 2.13, 2.14 for more about this special character).

The cursor is then moved to the beginning of the desired block. When Clear-X is hit, all text from the cursor to the End Block marker is deleted, and the screen is rewritten from the cursor on. The End Block mark is deleted as well. Any size block may be deleted in this fashion, from 2 characters on up to the entire buffer.

Block Delete will not work if the Begin Mark for the Copy operation (see 2.13) is set (this is to protect the user from setting up a block to copy and then accidentally hitting X rather than C, thus losing the whole block).

Block Delete will also not work if the cursor is positioned farther along in the text than the End Marker and it will not work if no End Marker is set. Because Block Delete can be so destructive, it should be used with some measure of caution.

2.12. LINE DELETE. Hitting Clear-K (Kill) will delete from the cursor position through the end of the current text line, including the CR which marks the end of the line. It may, therefore, delete more than a single screen line if the cursor is sitting in a continuation line.

Successive line deletes starting in the middle of a line, will usually turn the current line (the one with the cursor in it) into an overflow line. Since Line Kill deletes the CR at the end of the line, the screen is rewritten, and the following line is tacked on at the cursor, often resulting in a line exceeding the display width. Line Kill is thus optimally used with the cursor at the left margin of the line to be deleted.

Line Delete will not work on a line containing a Begin or End marker.

2.13. BLOCK COPY. To Copy a block of text, the cursor is moved to the beginning of the block (anywhere in a line). Pressing Clear-B (Begin) will put a dark bracket on the screen at the start of the block, marking the beginning. The end of the block is marked as in Block Delete, using the Clear-E key.

The cursor is then moved to the position where the block is to be copied. Pressing Clear-C will cause a copy of the block to be created, starting at the current cursor location. The screen will be rewritten, the Begin and End marks will be deleted, and the cursor will be sitting at the beginning of the new copy. Text can be inserted or deleted anywhere, even after the copy block has been marked, and before the actual Copy is executed.

If a block to be copied is too big to fit, an error message will appear reporting how many characters it's too big by. BREAK or ENTER will exit the error condition and return to the menu. From here the text editor is re-entered in the normal way (E). The Begin and End Marks will remain set. Deleting the number of characters indicated by the error message, will allow the block to be copied. The Begin and End markers should be deleted (Clear-Z, see 2.14) if no copy is to be executed.

Block Copy will do nothing if the Begin and End marks are inverted in order, or if only one or neither of them are set. It will, similarly, do nothing if all or part of the marked block is past the temporary EOT in Speed Mode (see 2.20).

Block Copy will not copy a block into itself. Again, it will do nothing, if this is attempted. It is up to the user to make sure the marks are properly set. (See 2.25 paragraph 3 for restrictions on using Block Copy in Speed Mode.)

2.14. BLOCK MARKS. Once a given Block Mark (Begin or End) is set in the text, no others (of the same type) may be set. (If it's tried, nothing will happen.) To set a new one, the old one must first be deleted. This may be done in the normal way (moving the cursor to the Mark and hitting the BREAK key) or by hitting Clear-Z.

The Clear-Z command will delete all Begin and End markers in the text without the need to search for them. When the command is done, the screen will rewrite, and the cursor, still pointing at the same character in the text, will be moved to the middle of the screen.

(Clear Q, can be used to search quickly for either of these block markers (See 2.18).)

2.15. MOVE. To effect a "Move", the desired block of text is marked and Copied to the new location (2.13). Then, the original block is returned to and deleted, using the Block Delete command (2.11). (Marking the block with some oddball pattern like "QQ" after the End Marker, enables a quick return

to it for deleting (after the copy is done), by using the "Find" command for that pattern.)

2.16. PAGING. Clear-P pages forward through the text a screen at a time. Clear - (Clear minus) pages backward through the text. After a Clear-P, the cursor is sitting at the second screen line of the new page which was formerly the bottom line of the old page. This helps provide continuity when paging through the text.

Similarly, after a Clear minus, the cursor is at the bottom of the screen at what was previously the top line of the last (forward) page. The cursor will not be at the bottom of the screen if there is less than a page to go in the reverse direction when Clear minus is executed.

2.17. FIND. Clear-F is used to Find a specific pattern in the text. When Clear-F is hit, a blank lo-res screen appears along with the flashing BASIC cursor at the upper left.

The desired pattern is typed in at the cursor, terminated by ENTER. As in the menus, BASIC conventions hold here for input, screen display, and line editing of the pattern (e.g. lowercase appears as reverse video -- see 1.3). When the pattern is ENTERed, the editor screen jumps back and, if the pattern was found, the cursor is sitting at the start of it. If no instance of the pattern was found in the text, the cursor stays put. The search starts at the current cursor position and moves in the forward direction through the text. (Because ENTER is used to terminate the pattern, CR cannot be a part of the pattern searched for. Thus, an instance of a multi-word pattern will not be found if it crosses a text line boundary.)

Hitting Clear-N will move the cursor to the "Next" instance of the pattern. When there are none left to find, the cursor will just stay where it is. Clear-N always searches in the forward direction. Jumping back to the top of the text will enable finding all the same instances all over again.

The characters "[" and "]" are typed in the search pattern using the BASIC conventions, Shift Down Arrow and Shift Right Arrow respectively. In the Telewriter-64 editor, these characters are generated by Shift-J and Shift-Clear (2.1). Since the tab is not a single character, it cannot be searched for as such. To search for the backslash character, use the BASIC backslash key combo SHIFT CLEAR. In the Telewriter-64 editor, backslash is generated by CLEAR-,.

2.18. SEARCH FOR SPECIAL CHARACTERS. The search routine described in 2.17 works for patterns of standard alphanumeric characters but does not work for "special characters" like the Begin Block marker, the End Block marker, the Embedded command marker "^" (see 6.1), and the direct code symbols (see 6.8) which are not included in the Color Computer's normal character set. The Clear-Q command is provided to search for these special characters. It searches in the forward direction and does not distinguish between them. It'll find whichever one is next until there are no more left.

2.19. GLOBAL (SELECTIVE) SEARCH AND REPLACE. Clear-G invokes Global Search and Replace. This works like Find at first (2.17), but when the search pattern is terminated with ENTER, the Editor screen doesn't reappear. Instead, the cursor jumps to the next line on the BASIC screen and waits for a "replace pattern". This is terminated by ENTER, and, as in Find, the editor screen returns and the cursor points at the first found instance of the search pattern. As in Find, Clear-N will find all subsequent instances of the pattern.

But now, pressing Clear-R at any found pattern, will "Replace" it with the replace pattern. This means each instance can be examined and selectively replaced with one quick command. If no replace pattern is given (i.e. double ENTER after the search pattern is typed in), then the found pattern is deleted.

Hitting BREAK while inputting either a search or replace pattern, terminates the command and returns to the editor screen wihout a change in the cursor position. The size of the replace pattern cannot exceed 127 characters.

2.20. WILD CARD SEARCH. The BASIC Up Arrow key, included in a search pattern, will match any character in that position in the found pattern. Any number of such "wild cards" may be used in a single pattern. This also provides a way to search for multi-word patterns where a CR may exist at any of the space positions.

2.21. HIGH DENSITY DISPLAYS. The normal 51X24 Telewriter-64 display can be instantly switched to a 64X24 display by hitting CLEAR-:. Hitting CLEAR-9 will switch to an 85X24 display. Clear-0 will return to the original 51X24 display. A 6 pixel wide character set is used by the 2 high-density modes and a 4 pixel character is used for the 51 mode. The CLEAR-; command allows the character set to be manually switched between the two. The readability of the high density displays is marginal, but they provide an excellent way to view the full width of the text lines as they will be printed.

2.22. PAGE FINDER - VERTICAL TAB. The CLEAR-V command will jump the cursor from its present position to a position exactly one printed page length away. This allows finding the page breaks as they will occur on printout.

To do this, the cursor is started at the very top of the text buffer or at a previously found page break. When CLEAR-V is hit, the screen rewrites and the cursor is now sitting at the first character of the very first line of the next page. The format parameters: Chars/Line, Spacing, Lines per Page, Upper margin, and Bottom Margin must be set to the values they will have during printing. Otherwise, page breaks found will not match the printout. U and B must be set in the Format menu.

(The only time this is not true is when headers are used. In that case, during scanning with CLEAR-V, Upper Margin must be set to a value which is the sum of the Upper Margin at print time plus the number of linefeeds after the header, plus 1.

Then, before printing is to be done, Upper margin must be set back to its lesser value to accommodate the header.)

The cursor must always start at the very beginning of the current line when scanning for the next page break. When CLEAR-V passes an embedded C (Chars/Line - see 6.2 below) command, it changes the line width of all subsequent text to that value. Line width is changed back upon passing the next embedded "C" command with CLEAR-V. In this way, once the whole text hass been scanned with CLEAR-V, all linewidths shown on screen will exactly match their printed counterparts.

The recipe for scanning for page breaks across files that will be chain printed is: 1. Load the first file and scan it for breaks (optionally mark them with ^T lines (see 6.5)). 2. Save the file. 3. Keep the last page (from the very last break that CLEAR-V finds) in the buffer. 4. Delete ALL preceding text (and from CLEAR-Q til the End Of Text marker, if it's disk chaining). 5. Append the next file in the chain. 6. Start the cursor at the top of the text buffer (which is the page break from the previous file), and continue paging forward with CLEAR-V. 7. Repeat the process with all subsequent files.

Remember to save files changed in this process, and remember to first delete the extra page garbage at the top of the text buffer before saving them. Files which will undergo this process should be kept small enough so that at least 2.5 to 3 K of SPACE is left in the buffer when they are saved (so they can be appended onto text buffers with upto a page of text already in them).

Alternatively, all files can be saved so that they begin and end at page breaks. In this case, Clear-V must make it to the last line of the file (the EOT or, in the case of disk versions, the ^Q line -- see 6.14), which means that the next file will start at the very first line of the next page. The last line of a file should also be the last line of a paragraph, as it will not be right-justified.

Page-break scanning should, generally, be done only when the document is completely finished and ready for printing. Clear-V can, however, serve in a somewhat looser fashion to quickly tell at any point in the writing, how many printed pages the document will run, and/or, whether it'll fit on, say, 1 printed page.

Note that, if headers are used in files to be chain printed, for CLEAR-V to function properly, the immediate header form (H+ - see 6.13) must be used in all files after the first.

2.23 ADJUSTABLE TAB STOPS. A ^T embedded command line (see sections 6.1 and 6.6) is used in the text to set tab stops. It must be the very first thing in the text buffer. The T must be followed by a space and a series of 2 digit numbers. A CR will terminate this tab setting line, and any subsequent number will be treated as text. Each number in the line must be 2 digits, and each successive number must be higher than the preceding one, otherwise hitting the tab key will cause

strange things to happen (lots of unexpected inserted spaces, for example).

The tab stop number refers to a column on the screen. The first is called 0, so a tab stop of 05 would be at the sixth character or space on the screen line.

The tab key is CLEAR-ENTER. If no tabs are set, hitting it will do nothing. With tabs set: 1. Hitting tab key on an empty line will insert spaces til the next stop. 2. If there is text at the next stop, the cursor will simply jump to it. 3. If there is text ahead of the cursor but the line falls short of the next stop, when tab is hit, the cursor will jump to the end of the existing line and then insert spaces til the next stop. Used carelessly, the Tab feature can leave excess spaces lying around unseen, that may cause later confusion.

When creating tables using the tab key, word wrap should be disabled (CLEAR-D) and the table should be preceded and followed by the embedded non-align command (^; - see 6.7) so it will not be destroyed if CLEAR-A is hit accidentally or intentionally. If word wrap is not disabled, and a table entry is made that exceeds the right margin of the display (51, 64, or 85), then wordwrap will occur, and a CR will be inserted, destroying the integrity of the line.

2.24. SPEED MODE. Once the text buffer contains more than 6,000 characters (approx. 3 manuscript pages -- possible only with a 32K or 64K system), going back and inserting or deleting text in the early part of the buffer may get a little slow. To return to higher speed response, the "Speed" mode is used. It is invoked with Clear-S.

First the cursor is moved to where the insertion or deletion will happen (or near it). Then, when Clear-S is hit, the screen will rewrite and the End of Text mark (EOT) will appear not far down from the cursor (though, if the cursor is near the bottom of the screen, the EOT may be out of view).

Essentially, most of the text buffer has been pushed out of the way so insert and delete can work fast again. (The new EOT marker is, of course, only a temporary one for the purposes of Speed mode, and does not represent the actual End of Text.) In this mode, inserting, deleting and moving the cursor, work as always. The only difference in Speed mode is that, when it's time to move down in the text, the cursor will not go past the temporary EOT marker.

2.25. BUSINESS AS USUAL. To get back to the "Usual" state from Speed mode, Clear-U is hit. This eliminates the temporary EOT and resets the real one so that all the text is accessible again (of course, inserting or deleting in the early part of the buffer will again be a little slower). When Clear-U is invoked, the screen will rewrite from the cursor down.

Usual mode is automatically invoked whenever there is a return to the Main menu from the editor. (In this case, the screen will rewrite before returning to the menu.) This means that

any command issued at the menu level will always be dealing with the entire buffer. Other commands (Clear-Z, Clear-S) will automatically exit the Speed mode if required to carry out their functions. If Clear-S is hit while already in Speed Mode, the screen below the cursor will rewrite twice, and the temporary EOT will be moved accordingly.

The Copy Block function will not work if the block marked for Copy is all or in part beyond the temporary End of Text established by Speed mode. In this case the user must explicitly invoke the Clear-U command and then do the copy again. (The block markers (2.14) remain set, however.)

If a Clear-A is done while in Speed mode, the realignment will be done only upto the temporary EOT.

Note: It is sometimes possible to lose track of being in Speed mode. If a large chunk of text seems to have suddenly disappeared, hitting Clear-U will often be all it takes to make it miraculously reappear.

3. CASSETTE HANDLER

The cassette I/O routines are accessed through the Main menu, and all follow the same format. When the first letter of the command is hit (S, %, R, A, V) the Color BASIC cursor will appear immediately to the right of the one selected, waiting for a filename to be input. Again, BASIC conventions are followed for giving the filename (see 1.3). It can be a maximum of 8 alphanumeric characters long, terminated by ENTER. When ENTER is hit, the command executes. If BREAK is hit at any point before ENTER, the whole command is aborted and the Main menu is re-entered and ready for a new command.

If only ENTER is hit and no filename given, then the next file on the tape is used. This should not be done with a Save command, since nameless files will cause problems. For Read in, Append and Verify commands, though, it generally makes life easier, as long as the tape is positioned at or just before the desired file.

3.1. AUTO RETRY. In normal BASIC usage, attempting to load or skip a file on tape will generate an "I/O error" if the tape is not positioned either at the exact beginning of a file, or at a totally blank part of the tape immediately preceding a file. This can make it difficult (and frustrating) to jump freely around in the tape using fast forward and rewind. It can also be quite annoying if you lose your place in a tape.

Telewriter-64 solves this with its auto-retry feature. Stated simply, it just keeps retrying the given command until a valid file is found. While it's searching for the file, an S will appear alone in the upper left of the blank screen, blinking between normal and reverse video. When a file is found, its name will appear to the right of the S (which becomes an F if it's the right file). It then carries out the desired operation (Verify, Append, or Read in). (Occasionally, this approach might miss a file. See Appendix B for more on this.)

While the S is flashing (and upuntil a file is found), the search may be aborted by hitting BREAK (holding it down for a second or two if there's no immediate response). The search will stop and jump back to the Main menu. Once a file is found, however, an abort (if desired) must be done as described in 3.4 below. If the S is alone on the screen and not flashing (and the recorder is running), then the tape is at a blank spot. An abort cannot be done at this point. To abort it, the tape should be moved forward or back to an area of the tape with data (non-blank). When the recorder is put on play, and the "S" flashes, the BREAK key will abort the file search and return to the menu.

The auto-retry feature is quite important for word processing with cassette, especially if a large body of text is involved. With many text files to make up the document, and many revisions, and fragments of the files, it is essential to be able to access any of them quickly. This can be done by keeping a record of the files on a tape along with their counter numbers. Then all it takes is fast forward or rewind to a position anywhere near the desired file. At that point, it is easily accessed with a single command, issued only once. This feature goes a long way towards making cassette based word processing competitive with disk systems. It can also be a life-saver when the position on the tape gets lost or confused.

3.2. SAVE. The Main menu command S will Save a file to cassette. The filename typed in stays put on the screen until the operation is done. Then it goes away and the Main menu is ready for the next command.

Once ENTER is typed at the end of the filename, the store starts. It can't really be aborted, but it will ultimately run its course and return to the menu even if the tape is turned off (it'll run from a few seconds to more than a minute depending on the size of the file). None of the text will be changed or harmed by this. Only the tape. When the Save filename disappears from the menu, the Save is done.

3.3. PARTIAL SAVE. "%" will save a part (%-age) of the text in memory. The part to be saved is marked as in block delete (2.11): The cursor is moved to the end of the block and Clear-E is typed, putting the End of Block marker there. (See 2.14 for End Mark conventions.) Then, the cursor is moved to the start of the block to be saved. When "%" is hit in the Main menu and a filename given (followed by ENTER), the marked block of text is saved to cassette with that filename.

After the %save, the End Mark is automatically deleted from the text. %save will do nothing if the End Mark has not been set.

3.4 READ IN-FILE. The R command Reads a file into memory from tape. Since executing this command will destroy the text currently in memory, it queries first with "Sure?", before accepting a filename. The only positive response to this query is upper case Y or a word beginning with uppercase "Y" (like Yes). All others will cause a return to the menu level when ENTER is hit, and the Read in will not be executed.

A "Y"(ES) response sets the cursor back to the Read in command waiting for a filename. From here on it works like the other I/O commands. When ENTER terminates the name, the tape starts, and when the file is found, it's read into the text buffer.

Once a file is found, a Read in may be aborted by simply stopping the tape momentarily (this is true even if the file is not the one named, and there's an S to its left rather than an F). When it's started again, "I/O ERR" will appear to the right of the found filename at the top of the screen, and the recorder will stop. BREAK or ENTER will return to the Main menu. If the text buffer was empty at the time of the Read in, it'll still be empty. If there was text in it, some data from the text being read in will appear in the buffer.

Sometimes a "natural" I/O error may occur during Read in, Append or Verify (e.g. from garbage on the tape). In this case the same thing will happen. The tape will stop and the screen will say "I/O ERR" and ENTER or BREAK will return to the Main menu from the error condition. Returning to the editor and hitting Clear Up Arrow will reveal that some of the new text has been read in. Appendix B describes a method for salvaging at least some of a file with an I/O ERR.

3.5. APPEND. A will Append the file whose name is given. There is no query because this is non-destructive, it simply finds the file and tacks it on to the end of the text already in memory.

If the file is too big to fit in available memory with the text there, a "FILE TOO BIG" error message will appear soon after the file is found. The name of the file will appear along with the number of characters the file was too big by. This allows deleting the right amount of text so the Append can be done successfully. (The sum of the "Too big" value and the "Space" value in the Main menu, is the amount the "Space" value must exceed for the appending of that file to occur without error.)

An Append may be aborted in the same way as a Read in (see 3.4), however, upon returning to the editor, nothing will have been appended, and the original text in the buffer will be unharmed.

3.6. VERIFY. The V command works like BASIC's SKIPF. It can be especially useful for verifying that a file just saved has been saved successfully (i.e. Rewind to slightly before the file and type V ENTER. If there is a problem, the tape will stop, and "I/O ERR" will appear to the right of the filename. ENTER or BREAK will return to the menu.). It is also useful when the current position of the tape is unknown, and it can be used repeatedly for making a "directory listing" of a given tape. As with SKIPF, if it's given a filename, it will search until it finds that file, stopping at the end of it, otherwise, if no name is given, it will go to the end of the next valid file it finds, reporting its name at the top of the screen.

Verify is aborted in the same manner as Append and Read in.

4. MAIN MENU

The I/O functions described in section 3, occupy most of the Main menu. The remaining functions are described now.

4.1. NEWFILE. The N command clears out the text buffer so a New file can be started. Since it is so destructive, it queries the user first with a "Sure?" prompt, before wiping out all old text. (If the text buffer is empty, however, it will not query.)

Uppercase Y (ENTER) (or any word beginning with uppercase Y) is the only way to get past this point to the editor. Any other response will cause a return to the menu.

4.2. FORMAT MENU. The F command jumps to the Format menu, described in detail in Section 5. In 16K systems, the format routines are "swapped" from disk or tape. In 32K systems, this swapping is optional and will provide an additional 3K of memory. The "swappable" print/format module (called "F") is supplied on the Telewriter-64 disk, and 10 copies of it are provided on side 2 of the Telewriter-64 tape, stored end to end from the 2nd Telewriter file, to the end of the tape.

4.3 STATUS INFORMATION. SPACE indicates the number of characters that there is still room for in the buffer. When this number goes to zero, a NO ROOM error message will appear whenever an attempt is made to insert text (including special characters like the Begin and End Markers and the Format codes (see 5.)). As usual, ENTER or BREAK will cause a return to the Main menu from the error condition.

The FILE entry at the bottom of the menu indicates the name of the last file that was read in or appended (whether or not the load was aborted or ended in an I/O error).

Pressing W invokes the WORDS command. The value which then appears to the right of the command is the total number of words in the text buffer. At the same time a value will appear next to the word LINES. This is the number of text lines in the buffer. If the End Marker is set in the text, the WORDS command will display the number of words between the cursor and the End Mark as well as the number of lines in this block. By setting the CHARS/LINE parameter, and then executing the WORDS command, the number of pages in the final printout can be estimated.

4.4 CHARS/LINE. The C Command allows setting the line width to be displayed on the screen. If this line width exceeds the maximum width of the screen (51, 64 or 85 depending on the display mode), then the line will wrap around. During normal usage, C should be set to 50 (its value on startup). When viewing the page format prior to printing, C should be set to the desired line width of the printout, and a display format at least one column wider should be used to view the full width of the lines as they will be printed. The command is essentially the same as the C command in the FORMAT Menu.

Whenever CLEAR-A (Align) is hit, the width of the adjusted lines will be determined by the value in Chars/Line. If it's kept at 50 during normal writing (in the 51 column mode), Align will always produce clean, non overflow lines. If it's set for, say, 60, or even 51, many lines will overflow the 51 column screen boundary, causing a confusing appearance. Of course, these wider lines can be accommodated by the higher density display modes (64 and 85).

5. FORMAT MENU

The Format Menu is used to format and print the text in memory, to send control codes to the printer and to set the baud rate. It is accessed from the Main menu by typing F.

In 16K systems, the format routines are non-resident. That means that for A) disk systems: a diskette with the file "F" (supplied on the Telewriter-64 disk), must be in drive 0 when the F command is hit, and, B) for cassette based systems: side 2 of the Telewriter-64 tape should be in the recorder, with the PLAY button depressed. In disk systems, hitting F automatically calls in the format routines. In cassette systems, hitting F gives a prompt to "INSERT FORMAT TAPE" (i.e. Side 2 of the Telewriter tape). When ENTER is hit, the format routines load from tape, and the Format menu appears.

The bulk of the Format menu consists of the format parameters. The current value of each parameter is displayed immediately to its right. These are initially set to standard values by the program. A value is changed by hitting the initial letter of the parameter and then typing in the new value. The old value is obliterated when this is done.

Hitting BREAK at any time before hitting ENTER will abort the change and keep the old value for that parameter.

5.1. SPACING. The S parameter will take values from 0 to 255. A value of 1 means single spacing, 2 is double spacing, 3 is triple, etc. A zero here will get single spacing as well. Numbers larger than 255 will yield unexpected results.

5.2. MARGIN. M sets the left Margin. The printer will indent every line this many spaces. Assuming an 80 character line, common values would be somewhere between 0 and 12. The default value is 0. (Note: by doing multiple passes over the same page, and shifting the margin each time, it is possible to get pages printed in columns. See Appendix A.)

The largest value for left Margin is 255, but high values can cause problems. If the sum of the Left Margin and the Chars/Line parameter (5.3) exceeds the maximum line width of the printer (normally 80), then the print line will overflow, causing wrap-around lines on printout and top and bottom margins and page numbers in the wrong places.

5.3. CHARACTERS PER LINE (CHARS/LINE). C will set the line size (maximum # of characters in a line). This, combined with the M setting, will determine the right-hand margin. Thus, on

an 80 column printer, if Left Margin is set to 6 and Chars/Line is set to 66, then the right margin will be 8. Note that with the same line length (66) and a Left Margin of 14, the printer line would overflow causing the problems mentioned in the last lines of 5.2 above.

The maximum value for C is 127. Any greater number will not alter the margins. Since most reasonably priced printers don't exceed 132 columns wide, this maximum should be more than sufficient for all needs.

Setting the Chars/Line parameter will realign the actual text in memory, so that the length of each text line (2.8) will get as close to that value as possible without exceeding it.

The Chars/Line parameter should always be set to the desired line width before printing and especially after changes have been made to the text. The value in the menu reflects the last size the parameter was set to, and additions to or deletions from the text since that time will undoubtedly leave some lines chopped up. Thus, the desired value for Chars/Line may appear in the Format menu, while actual segments of the text don't conform to this number.

There will be a slight delay (2 seconds or so) when the new value is set. Both the cursor and the word QUEUE will disappear until the realignment is done. Then the updated menu will appear, ready for further commands.

5.4. UPPER MARGIN. U sets the number of line spaces the printer will skip at the top of each page. Default is 4 which provides a fairly small border at the top.

5.5. LINES PER PAGE. A standard 9 1/2 X 11 page holds exactly 66 lines at the standard 1/6 inch line spacing. The default value for the L parameter is therefore initialized to 66 and generally won't need to be changed. It is wise, however, to verify that it's 66, otherwise the tops and bottoms of pages will not be properly found.

(If the printer's line spacing constant is to be changed, then this number must take that into account. Thus, if an MX-80 is set to do 1/3 inch on every line feed, there would be only 33 lines on a page. Similarly if a 1/2 page form were used.)

It is also important to set the top of the paper in the printer so that the first line printed corresponds pretty closely to the very first line on the paper following the perforation.

5.6. BOTTOM MARGIN. B sets the amount of space (number of lines) left (empty) at the bottom of the page. The default is 4. This value must be at least 3 in order for a page number to be printed (if page numbering is specified (5.14)). This is because the page number is printed 3 lines up from the bottom, centered on the text itself.

5.7. FONT. The F command is only valid if the printer used is an MX-80 (or a printer with the exact same font control

codes). If no font is set, the menu shows nothing after the word FONT.

The MX-80 has 4 print modes, which can be combined to form 12 legal fonts (not all combinations of the 4 are legal). Graftrax adds a few more, for combinations approaching infinity. For the purposes of setting the font parameter, each mode is assigned a number, as follows:

emphasized = 1
double = 2
condensed = 3
expanded = 4

Each invocation of the F command can set only one of these numbers, but each new number set in this way, will be added to the list, so that "combined" fonts can be easily specified (e.g. expanded condensed (3 4), double emphasized (1 2), emphasized expanded (1 4), Double emphasized expanded (1 2 4), etc.).

This list is cleared (i.e. no special font will be used), by giving F a zero for its value. To get from 3,4 to 2,3, for example, it is necessary to clear the list first, with F 0 ENTER, then to hit F 2 ENTER, F 3 ENTER, to set the new list.

If the Enlarged font is selected in the menu, either by itself or in combination, it will be maintained from line to line. In normal MX-80 usage, the enlarged font code is cancelled after a line is printed, but with the Telewriter MX-80 driver, the enlarged mode, once asserted, will remain in effect until explicitly cancelled. This feature is provided mainly to allow easy access to the enlarged condensed font (3 4 in the Format menu).

The EPS/OKI/LF parameter (5.19) must be set to 1 or 2 for the font feature to work.

5.8. BAUD RATE. X (for XMIT RATE) sets the baud rate. The number input is the internal number used by BASIC to determine baud rate. The following table shows the value of this number for some standard baud rates.

baud	number input
110	498
120	458
300	180
600	87
1200	41
2400	18/17
4800	6

The default is 87 (= 600 baud), because that's the standard baud rate used by Basic to write to the printer. By setting the appopriate switches in the MX-80 serial interface, it is possible to run at 4800 baud. The same should be true for other printers as well.

5.9. PRINT. Hitting P causes all the text in memory to be printed out. The word PRINT will appear next to the print command. This provides a way of knowing if P has been accidentally hit with the printer off-line. Without it, the program could hang indefinitely in the Format menu without taking any commands and without indicating why.

Hitting the BREAK key will abort the printing (the printer must be on-line at the time). It may be necessary to hold it down for a few seconds before it takes effect. In the event that P is hit accidentally, holding the BREAK key down while putting the printer on-line, will return to the Format menu. At most, one line will be printed out. (This should work even if the baud rate is not set to the right value).

It is generally advisable to save a text buffer before printing it out, just in case.

5.10. PARTIAL PRINT. The "%" key in the Format menu works more or less like partial save (3.3). In the editor, the end of the block to be printed is marked with Clear-E, and the cursor is moved to the beginning of the block. Then, in the Format menu, the % key is hit, and the text between the cursor and the End Mark will be printed. %Print will do nothing if the End Mark has not been set.

5.11. RETURN. The R command will Return to the Main menu from the Format menu. If the non-resident print routines are used, R will return to a blank editor screen. At that point, Clear-Up Arrow will reveal the text again.

5.12. CONTROL CODE OUTPUT. The Direct command (D) allows sending control codes directly to the printer. When D is hit, it waits for a decimal number. When ENTER is hit at the end of the number, that value is sent to the printer. The cursor will return to the immediate right of the word DIRECT, waiting for another number. It'll do this after each number input. The BREAK key here will exit the Direct output mode and return to the top of the Format menu.

The largest number that can be sent is 255 (127 in Color Computers with Color Basic 1.0). The printer, of course, must be on-line when these values are sent.

(Example: hitting D, then 11 ENTER BREAK will send the number 11, causing a line feed on most printers.)

5.13. TYPEWRITER MODE. The TYPEWRITER command makes the printer work like a typewriter. When T is hit, the screen will blank with the BASIC cursor at the top. Whatever is now typed at the keyboard will be printed out when ENTER is hit. The cursor will continue down the page with each CR hit. As with Direct mode, Typewriter mode is terminated with the BREAK key.

Each line typed in this fashion may be edited (using the BASIC editing conventions enumerated in 1.3) at any point before ENTER is hit and it's sent to the printer.

5.14. NUMBER PAGES. To number pages, the N parameter is set to a non-zero number. Each printed page will be numbered at the bottom center, starting with the number given, and incremented by 1 for each successive page. The number is printed 3 lines from the bottom of the page, and the Bottom Margin parameter (5.6) must be set to at least 3 for the page number to print.

Setting the NUMBER PAGES parameter to zero (its initial value) will suppress page numbering (as will setting Bottom Margin to 2 or less).

5.15. WHERE. The page number may be positioned anywhere at the bottom of the page by setting the W (Where) parameter to the position number desired. Centering the page number involves taking half the Chars/Lin parameter and adding the value of M. (Note that if the printer is supplying the left margin, the Telewriter margin should be set to zero and only half of Chars/LIn should be used to center the page number). On a 78 character line with no margin, the center position is, of course, 39.

5.16. RIGHT JUSTIFICATION. Setting R to 1 (or any non-zero number) will cause the text to be right justified on printout. Justify works on the lines as they are established by the Align or the Chars/line commands. That is, after Chars/Lin is set in the Main (or Format) Menu, the lines on the screen represent the full lines as they will be printed. Right justification will add spaces to pad out these lines, but it won't change how much actual text is on the line (optimally viewed in a display mode at least one greater than Chars/Lin).

If printing control codes are used (e.g. special foreign or math symbols available on some printers) they must be defined as such (see 6.9) for the lines they're in to be accurately justified. If print size is changed within a justified line, that line will not justify properly, though boldface, underlining, subscript, superscript, etc. present no problem as long as the printer has codes to do them.

In justified text, the last line of a paragraph will not be justified, and the last line of a text file will not be justified. For the latter reason, when chain printing, it is a good idea to end each file at a paragraph break, otherwise a line that should be justified will go unjustified.

The justify routine calls a paragraph anything preceded by a space (indented one or more) or preceded by a line space. When looking at the next line, it ignores any embedded command lines (section 6). This means that, normally, an embedded command line can not substitute for a paragraph break. If one is put into the only linespace separating 2 paragraphs, the linespace break will appear to be gone and there will be no break between paragraphs on printout (and a line that shouldn't be justified, will be). If, however, an embedded command line is indented one space, the preceding line will not be justified, regardless of what the next line is.

5.17. ONE PAGE. It is sometimes useful to have the printer pause at the bottom of each page (especially on friction feed printers that can use 1 sheet at a time). Setting the One page parameter to 1 (or any non-zero number) will accomplish this. When the printer gets to the bottom of the page (after completing the bottom margin), the printer will stop printing and a flashing cursor will appear beside the word PRINT on the Format menu. At this point, the printer will not continue until either ENTER or BREAK has been pressed.

5.18. QUEUE. The Queue feature allows printing a number of consecutive cassette files without user intervention. The Queue parameter is set to the number of files to be printed, and the recorder is put on Play.

When the Print command is given, all text currently in the memory will be printed out (a % command will begin by printing the marked block of text). When that's done, the next file will be read from cassette into memory and printed out in full. When that's done, the next file is read in and printed, and so on, until it's done as many as the Queue number requested. If the Queue value is set to 3, then 3 cassette files are read in and printed after the contents of the text buffer are printed.

The transitions between files on the printout will be unnoticeable as long as each cassette file begins and ends at a paragraph boundary, and as long as no excess line spaces are included at the top or bottom of the text files.

When not chain printing a number of files, Queue should normally be set to zero. If it is not set to zero, after a print, the program will "hang" looking for a cassette file, and will not respond to commands. If the recorder is off (not on Play), the filesearch S will appear at the upper left of the blank screen, and nothing will happen.

(The solution is to turn on the cassette recorder and hold the BREAK key down (3.1), aborting the file search, and returning to the menu. If a file is found before the search is aborted, stopping and starting the tape will generate an I/O error, from which BREAK will return to the menu (3.4).)

Note: In files to be chained, it's a good idea to set the main format parameters (S,C,M) using embedded format commands (6.2) at the top of each file, to insure consistency. Each file read in during a chain print will be automatically set to the current value of Chars/Line (5.3). This will, however, be overridden by an embedded format command (i.e. C) placed at the top of the text read in.

5.19. EPS/OKI/LF. This should be set to 1 if an Epson MX-80 without Graftrax is the printer being used. If it's an MX-80 with Graftrax, E should be set to 2. Setting E to either 1 or 2 will enable the EPSON font feature (5.7) as well as MX-80 underlining (5.20) with or without Graftrax.

For use with an Okidata or Centronics printer, E should be set to 4. This will supply required line feeds where the OKI and

the Centronics won't send them (on consecutive CRs or lines of 0 data). The printer should be set to supply auto-linefeeds.

E should be set to 5 for all printers that do not supply an auto line-feed when sent a CR. GE Terminet 300,s, some daisy wheels, Teletypes, and a host of others do this, and most other printers can be set to do it.

5.20. UNDERLINING. To underline any chunk of text with any Epson MX-80, the underline character is simply put on each side of the chunk. The underline character is generated by hitting CLEAR-/ (CLEAR slash). It appears on the screen as a dark bar beside the lowest part of the character.

These underline marks must be balanced. That is, they must always be used in pairs -- one to indicate the start of the underline, another to indicate the end. If there is an extra one, or one short, the underlining can go on forever.

This feature is for the Epson only, and the Epson parameter in the format menu must be set to 1 or 2. Many printers, including the newer MX-80's and 100's, and those upgraded with Graftrax Plus, have a code for underlining, and this can be accessed in Telewriter-64 with the embedded control code feature (see 6.8). These printer-generated underlines are usually solid and generally preferable to the broken underline that Telewriter-64 provides for Epson printers.

When using a printer's "true underlining", it might be wise to set the Telewriter margin to 0 and use control codes to instruct the printer to generate the desired margin. The reason is this: If more than one word is to be underlined, and, on printout, that sequence of words extends beyond a single line, then the printer's underline feature will treat the Telewriter margin as characters to be underlined and this will put lines out in the margin. But if the printer is generating the left margin, it will know better. This is only of concern where more than a single word at a time is underlined.

6. EMBEDDED COMMANDS

Telewriter-64's Embedded commands provide a means to change both printer and page format parameters dynamically as the text is being printed. These "embedded" command lines exist in the text like all other lines, but they are not printed. Instead they send instructions to the print routines or to the printer itself. The embedded command syntax is used to access other features as well.

6.1. EMBEDDED COMMAND SYNTAX. The caret symbol (^) must begin an embedded command line at the left screen margin (not preceded by spaces, though this generally won't hurt). A letter or symbol follows the caret, indicating the function.

An embedded command line should be completely separate from the text and have no text on it. It will not be aligned with other lines, and Align won't work around it (i.e. if a short line

precedes an embedded command line, and a short line follows it, these lines cannot be combined by the Align routine because the embedded command is in the way, even though the two lines will appear consecutively on printout).

The caret (^) is generated by hitting CLEAR-period (.). An embedded command line is terminated by CR.

(Note: It is best to keep embedded command lines from exceeding the screen width. If they do, however, care should be taken to avoid getting CRs in them. Since align won't effect them, word wrap is the only way to get an unexpected CR in an embedded command line.)

If there's a syntax error in an embedded line, it will be printed out, starting with the code that's in error.

6.2. EMBEDDED FORMAT LINES The parameters discussed in 5.1 - 5.6 above may be changed dynamically, at any time during printing, using embedded format commands in the text.

The parameters are specified by the same (upper case) letter that invokes them in the Format menu. That is, M sets the left Margin at menu level and when used in an embedded format line. S sets the Spacing, C sets Chars per Line, U sets Upper margin, B sets Bottom Margin, and L sets Lines per Page.

The embedded format commands may be combined in any order on the same line. Thus

^M6 C63 S1

would set the Margin to 6, Chars/Line to 63 and Spacing to 1.

The parameter code (M, S, U, L, C, B) must be followed by a number which is its new value, and there must be no spaces separating it from its value (e.g. M6), but at least one space must separate each group of parameter and value (e.g. M6 C63 S1). The following example shows, first, what appears on the screen, and, second, what the resultant printout looks like:

```
^S2 M3 C59
This is a double spaced body of text (59
characters per line, left margin of 3 spaces)
leading up to a quotation.  In the original
Sanskrit, this appears as:

^M11 C40 S1
...this is an indented quotation, single spaced
and compressed to 40 characters per line and
indented (left margin =) 11 spaces.  This is a
line of padding .
^M3 C59 S2


This is a return to the normal body of the text.
This line is padding to make it long enough to
show the double spacing.
```

resulting in:

This is a double spaced body of text (59 characters per line, left margin of 3 spaces) leading up to a quotation. In the original Sanskrit, this appears as:

> ...this is an indented quotation, single spaced and compressed to 40 characters per line and indented (left margin =) 11 spaces. This is a line of padding .

This is a return to the normal body of the text. This line is padding to make it long enough to show the double spacing.

The first command line above establishes the Spacing (S) as double, and the left Margin (M) as 3. The second command line above says essentially: Change the left margin (M) to 11 (from 3), change the length of a line (C=Chars/Line) to 40 (from 59), and do Single spacing (S=1, formerly S=2). This will essentially pull the text in on both sides and single space it -- a standard way of including long quotes in text. After the quote, another format line appears to return the left margin, linesize, and spacing back to what they were before.

Note that if the U and L parameters are embedded as the first line of the file, or "stacked" (see 6.3) with others at the very start, they will be in effect for the first printed page. If they are embedded after the first printable line, they will not take effect until the next printed page.

6.3. OTHER EMBEDDED COMMANDS. The embedded commands described in the following sections will use essentially the same syntax as the embedded format commands, but they cannot be combined more than one to a command line. They can, however, be "stacked" one line after the other. Thus, the following "stack" (whose unfamiliar components will be discussed in subsequent sections) ressembles the one used to generate this document:

```
^T 05 20 35 43
^D1 27 33
^D2 27 34
^D3 27 88
^D4 27 89
^DP5 94
^DP6 16
^D7 27 83 48 48 48 56 0 0 0 0 0 18 31
^D8 27 76 48 48 56
^C63 M0
^H3 Telewriter Tutorial                                    page \ ^
```

6.4. NEW PAGE. The ^N embedded code causes the printer to start a New page when it is encountered during printing. No number needs to follow the N but if a number does, that will be the first page number used on the new page. If no number is given, the page numbering will continue as it had been (and if no numbering before, then none now).

The ^N embedded command can be used at the bottom of a text buffer to force the printer to number the last page printed. Normally printing will just stop after the last character is printed on the last page, but if the text buffer ends with ^N, then the printer will space to the bottom of the page, and put the page number there if required. The option of what to do at the end of a print, is, thus, left to the user.

6.5. COMMENT LINES. The versatile ^T embedded code provides a way to place "non-printing" lines anywhere in the text. This can be only one line long, terminated by CR, begun, of course, by ^T. The T must be followed by a space, and the ^T line must not be the very first thing in the buffer (see 6.6).

6.6. TAB DEFINITION LINE. The embedded ^T code is also used to define Tab stops. It is followed by a space and a sequence of 2 digit numbers which represent the column on the screen where the tab stop will be set. It must come at the very beginning of the text buffer (see the example in 6.3 above), otherwise it will be treated as a comment line.

6.7. NON-ALIGNMENT. If ^T begins a line and is not followed by a space, that line will be printed, but it will be unaffected when the text is aligned. Thus, no matter how long or short it is or what comes before or after, it'll always start flush left on the printed page and will not be combined with other lines. However, if the ^T line is longer than one half of Chars/Line, it will be justified.

Alternatively (and usually preferably), if the ^; codes surround a block of text, nothing in that block will be effected by alignment or right justification. The ^; is placed alone on its own line, usually immediately before and immediately after the block to be passed over.

Note that embedded format control lines that change the value of Chars/lin, must not be included within a non-alignment block delimited by ^;. Any change in Chars/lin must come either before the first ^; marking the start of the block, or after the second ^; marking the end of the block (and return to normal alignment).

6.8. CONTROL CODE OUTPUT. The ^D (Define Direct) command allows the definition of special characters which, when encountered in the text during printing, will send ASCII codes directly to the printer. These ASCII codes can be printable characters, or non-printing control codes (to change font, superscript, underline, etc.). In the example from section 6.3 above, the line

^D1 27 33

defines the special character ¹ so that when it is encountered during printing, the numbers 27 and 33, will be sent to the printer. This code would put a C.Itoh Prowriter into bold print mode. (^D1 27 69 would do the same for an MX-80.)

The special characters such as ¹ are generated by hitting the Clear key along with one of the numbers 1-9. They appear on the screen as a miniaturized version of that number. The number used in the definition statement (the number after the D) is one of the standard numbers 1-9.

Each of the 9 special characters can be defined to send upto 15 decimal numbers to the printer. The ^D line only defines a code. It sends nothing. The special characters send the code when encountered in the text. If the number (of the special character) has not been defined in a ^D line, then nothing will be sent.

^D definition lines may occur at any point in the text (on their own line, of course), but they must occur somewhere before the actual use of the special character they define. The same special character may be redefined and reused any number of times throughout the course of the text, and, for any given instance, the most recent definition will be the one used.

The special characters (¹, ², etc.) can be put almost anywhere in the text, freely mixed into text lines, and even included in Embedded Format lines. (In the latter situation, they must be separated from any format parameters in the line by at least one space (on each side), and they cannot be included in any other type of embedded command line (like ^; or ^N), only those described in section 6.2 - M, S, U, L, C,, B.)

If a special character is put alone on a line, it will send its defined code and the CR at the end of the line will also be sent. If this form is used, it is a good precaution to indent the special character by one space so there is no risk of it being aligned with a preceding line. If the special character is alone on the line, the extra space won't make any difference to the printout.

This latter form is useful (for some printers) when using these codes to change font size. If the code is placed at the beginning of a text line, the margin will be generated before the font size takes effect and the first line will be indented the wrong amount (this is similar to the complications that arise when centering and changing font size simultaneously (6.11 below)). But if the special character is placed on its own line preceding the actual text where the size change takes place, it can be invoked before the margin for that line is done, so that it will come out right. (It can usually be invoked at the end of the last line before the change begins, but some printers (e.g. pre-Graftrax MX-80s) have problems with this.)

It is sometimes desirable to send a control code alone on a line, without also sending the CR for that line (so no extra line space is printed, yet the code will still take effect). To do this, the form ^1 can be used. This is essentially an embedded command line (and therefore non-printing), without any standard embedded commands on it.

6.9. PRINTING CODES. Some characters and symbols not available from the keyboard or in Telewriter, can still be printed by sending the appropriate ASCII code to the printer. The curly bracket symbol, {, has an ASCII code of 123, so a line that said ^D1 123, would define ¹ so that whereever it appeared in the text, a curly bracket ({) would print. Since some codes print and some don't, the justification routine needs to know which are which so it knows which to count. For printing codes like 123, DP replaces D in the definition line, i.e. ^DP1 123, defines the special character ¹ to be the "printing" control code, 123, or left curly bracket.

For best results, only one such printing symbol should be defined per special character. However, if 1 special character is used to stand for more than 1 printing symbol, these will be counted properly by the justify routines, but, if too many are used in a given line, it will overflow the normal margin.

6.10. CENTERING. The ^* embedded command will cause the remainder of its line to be automatically centered on the page. Thus,

^* Telewriter-64 Tutorial

on the screen will produce:

Telewriter-64 Tutorial

on printout.

6.11. CENTERING AND FONT SIZE CHANGE. Centering and changing font size at the same time, can get a little sticky. This is because the size of a space is different in the two different font sizes, and Telewriter's centering routine uses spaces to center the line. This change in the size of a space must be accounted for, but different printers have different repetoires of print sizes and different control codes to access them, so it is difficult for the software to accommodate everybody, automatically.

This can, however, be taken care of explicitly by the user, without too much pain. Changing the line width (Chars per line) and the left Margin (M) just before the centered line, and then changing them back right after it, does the trick.

The change is based on the ratio of the maximum characters per line for the 2 sizes.

For example, on an MX-80 and for most printers, the normal print style can fit a maximum of 80 characters on a print

line. For the "Enlarged" font, it's 40 (i.e. 40 Double-width characters will fit on a printed line -- any more will overflow onto the next). So that ratio is 2:1 (80/40). Whatever is done to 80 (mathematically) to get the new max line size, is now done to the left margin and chars per line parameters.

Therefore, centering a "Double width" line in a body of normal size text, requires the current left Margin and Chars/Lin parameters to be cut in half (40 is half of 80, so cut the parameters in half -- centering condensed print in normal would require increasing the parameter values). This is done just before the centered line, and then the original values are reinstated immediately afterwards. Thus, in a text where these defining codes had already been given:

^D1 14
^D2 15
^M6 C63

(I.e. the embedded special character 1 will send 14, which is the Double width font command on a C.Itoh Prowriter (and many other printers), and 2 will send decimal 15 which, in a Prowriter, turns off Double width.)

the screen:

normal font size text sixty-three characters per line normal size font sixty-three characters per line.[1]

^M3 C31
^* Center this enlarged line
^M6 C63

[2]back to normal font size and original margins.

will print as:

normal font size text sixty-three characters per line normal size font sixty-three characters per line.

Center this enlarged line

back to normal font size and original margins.

So it's simply a matter of surrounding the line to be centered with an embedded format command that first halves the left margin and linewidth parameters, and then, after the centered line, returns them to normal.

For most centering involving changes in font size, the above approach would be used, but with an MX-80 (only) the following may be done:

When going from an 80 column font size to either Double width print, Condensed print, or Double width Condensed print, an equal (=) sign may be used in place of the asterisk (*), to center the given line. If the equal sign is used instead, then the margin change (M & C) will be taken care of automatically

by the program. So, for MX-80's, what, on the screen, would look like:

```
normal font size text sixty-three characters per line normal
size font sixty-three character per line.
 ↓
^= Center this enlarged line

^back to normal font size and original margins.
```

would print as shown in the example above, without the need to use ^M3 C31 and ^M6 C63, as required when the * was used.

Notice in the MX-80 form above that the ↓ is put on its own line and indented 1 space, to keep it from being combined with the text line. Nothing prints, but a CR is sent which puts a line space before the centered line. This form is shown mainly because some printers, like the early MX-80s without Graftrax, could not properly handle the first form above, which is preferred because it's a little neater to decipher on the screen. In the early MX-80's, font changing codes placed at the end of a line, frequently effected the whole line rather than taking effect "after" the line, as usually desired. So these printers may sometimes require a font changing code placed on its own line.

6.12. HEADER. To have a header printed at the top of every page except the first, the ^H embedded code is used. Its syntax bears many similarities to the other embedded codes but there are some differences.

The ^H signals a header definition line, and this line should be placed at the top of the text file, either as the first line, or in the very first stack of embedded commands (see example in 6.3). If any text, even a CR, precedes the Header definition line, the header will not start until page 3.

The line not only begins with a caret ^, it must also end with one. The Telewriter-set margin does not effect the header and it will start printing at the printer's column 0. Spaces must therefore be explicitly provided in the header definition line if the header is to have a margin.

The number after the H in the definition line sets the number of line feeds that will be sent after the header prints -- i.e. the spacing between the header and the text. This number must be one and only one digit long. If headers are used, the number of spaces from the top of the text to the header is determined by the value of Upper Margin in the format menu (usually 1 or 2). The example of a header definition line in section 6.3 above, would generate a header followed by 3 linefeeds, and then the text, on each page.

Generally, embedded control codes will work properly in the header, but there are some additional considerations for pre-Graftrax MX-80s. These exceptions will be discussed a few paragraphs hence.

It is possible to do multi-line headers by simply defining a special character (e.g. ¹, ², etc.) to output the 13 (carriage return) code wherever a new line in the header is desired. The header itself is not terminated til the ^ is encountered at the end, so every 13 that's encountered will do a CR/LF inside the header.

It is also possible to include the page number in the header by putting the backslash character (generated by hitting CLEAR-,) in the header definition at the point where the number is to be printed (see example in 6.3). When page numbers are printed in headers, the numbering starts with the number set in the Number Pages parameter in the Format menu.

If the header definition line is not terminated with the ^ character, unexpected results are promised. The Header definition line should not exceed 127 characters long.

When a header is to be used with chain printed files (5.18, 6.14), the header definition must appear at the beginning of each separate file. And, if the page finder feature (CLEAR-V) is to be used with the files, the header at the start of each one (after the 1st) should be an immediate header (See 6.13).

Since the pre-Graftrax MX-80s use the last control code sent in a print line to determine the print mode of that line, it's a little tricky to, say, turn emphasized on at the beginning of the header, and then off at the end. Normally, this line would not print emphasized at all because the off code sent last would dominate the line (Graftrax corrected this problem).

So, to handle this, the following syntax has been specially designed. The rule is: The ^ code terminates the actual content of the header and, at this point the header will be printed (a 13 code will be output), but after that, if there are any control codes on the header definition line before the CR that ends the line, then those control codes will be output as part of the header. That would allow turning off emphasized after the line had been printed as emphasized, so that the rest of the actual text would not be printed in emphasized mode. Again, this form is probably only needed on pre-Graftrax MX-80s.

Note that the Telewriter-provided underline feature for the MX-80 will not work in a header, but the Graftrax one will.

6.13. IMMEDIATE HEADER. A plus sign after the H in the header definition will cause the header to begin on the first page. This may be used if a document is printed in sections or to test the header quickly. It is required in all files after the first when the Page finder (CLEAR-V) is used across files that will be chain printed (see 2.22). It can also be used if the header form is used to number pages at the top.

6.14 DISK CHAIN PRINTING. The ^Q embedded command line, when encountered during printing, will cause the program to load the file whose name follows the ^Q, and then continue printing from the start of that file. In this way, any number

of disk files can be chain printed to generate one long document.

The file name must be separated from the Q by a space and it must include the file extension (even if it's BIN). It should be in the standard form (FILENAME/EXTin) where, if no drive number is given, the currently set default drive will be searched. If the file is not found, an NE error message will flash briefly on the screen, and then the program will return to a blank editor screen, with the last file still intact in the buffer.

6.15 NON-BREAKABLE SPACE. The backslash character, when used in justified text, will not print as a backslash, but, instead, as a space. This means that if 2 words are separated by a backslash (on the screen) rather than a space, the words will not have additional spaces stuck between them when justification pads the line. Thus, it prints as a space, but cannot be "broken". The backslash character is generated by hitting CLEAR-, (CLEAR-comma).

Appendix A: Optimizing tips

Speed:

Telewriter-64 generally responds fast enough for touch typing speeds, but there are situations where it can slow down. This section describes ways to avoid them.

1) When insertions are made into the middle of a line (meaning anyplace other than the exact end of the line), Telewriter-64 must rewrite the rest of the line after the cursor, with each insert. Usually this won't slow things down noticeably but, if the line is long (especially a lengthy overflow line), it may.

The real slowdown, though, comes when the cursor gets to the end of the line and wordwrap takes over. When this happens while inserting into a line, all the text on the screen below the current line will need to rewrite. Telewriter will not accept any characters from the keyboard until the rewrite is done. Even though it's just a second or two, this can be annoying if the insert is more than just a few chararacters.

The solution is, simply, to make the middle of the line stop being the middle of a line. Go to the place you want to insert at and hit ENTER. This pushes the end of the line down to the next line. Then do a left cursor and you'll be sitting at the end of the line you want to insert into (formerly the middle of the line). Now type in the insert.

Since you're at the end of the line, when wordwrap happens, the lines will scroll down out of the way, rather than rewriting, so you won't be slowed down. Then, when you finish typing the insert, hit Clear-A (Align) and the line you inserted into will be joined back together again.

A similar problem can occur behind your back. Sometimes it may appear that you're inserting at the end of the line when, in actuality, there are a few spaces after the cursor (which are unseen). If this is the case, wordwrap thinks it's inserting into a line and will rewrite the screen when it takes effect. The result is the same type of slowdown just described. The solution here (when you notice that the screen rewrites whenever wordwrap happens) is to hit Clear Right Arrow (the End of line command) to get to the true end of line. Then, just delete the extra spaces which are now behind the cursor by using Clear Break. Wordwrap will be fast again. (See 2.5.)

2) When the text buffer starts to get full, things can get slow if you insert or delete in the early part of the buffer. Hit Clear-S (Speed mode), and inserting and deleting will be fast again. (Clear-U gets back the rest of the text. See 2.24, 2.25.)

3) If there are overflow lines (2.7) in your text, and you're doing a Shift Up Arrow (so the text is scrolling down the page once you get to the top of the screen), each overflow line that is encountered coming down from above the screen will cause the entire screen to rewrite. This essentially destroys the smooth scrolling and can take a good deal of time to scroll a page by if every line is an overflow.

Two simple solutions exist for this. One, hit Clear-A to eliminate all the overflow lines (2.8) making certain that Chars/Line is set to a value less than the display width you're using. Scrolling will now be smooth and fast. If, for some reason, you don't want to destroy the overflow lines, then simply use the Clear - (clear minus) command to page backwards. This will still happen at its normal speed, whether or not there are overflow lines, and will be infinitely faster than trying to scroll by continuation lines.

4) Generally speaking, liberal use of the Clear-A command is recommended as it eliminates in one swoop, many of the conditions that cause slow-downs.

Proof-reading:

When proof-reading for errors, it can be quite helpful to use the right Auto-cursor (Shift right arrow), intermittently to guide you through the text and from line to line.

Auto Cursor Speed:

In the Telewriter-64 DISK version, location 12280 contains the delay constant for the auto-cursor. The value is initially 10. To speed up the auto-cursor, return to Basic and poke a smaller number into this location. (E.g., POKE 12280,9). To slow it, poke a larger number. In cassette Telewriter-64 the location to poke is 12093. In a 64K cassette system, the location is 60693 and in a 64K disk system, it's 60950.

Display optimization -- background color:

The background color of the Telewriter-64 editor screen is optimized for Black and White TVs. On color TVs, though, this "Buff" background produces "color sparkles" around the letters. On some sets this makes the letters hard to read. One solution is to simply turn the Color down or off on the set. But it is also possilbe to change the background color of the screen in software, so as to eliminate the color sparkles. To do this, return to Basic and do one of the following pokes as appropriate to your system.

CASSETTE	16/32K	POKE 12522,240
CASSETTE	64K	POKE 61122,240
DISK	16K/32K	POKE 12729,240
DISK	64K	POKE 61259,240

To get back to the original background color, poke 248 into the same location.

Columnar printouts

It is possible to get a printout in columns with only a little trickery. Assume you want to do two columns of normal size print on an 80-column printer. First, write out the text, then determine the line width you want (35-38 chars/line would be expected values for this). Go into the Format menu and set the chars/line parameter to that number (assume 36). Also set Upper and Bottom Margin to the desired sizes.

Then go back to the editor and use the Page Finder feature to determine where the first column will end (a full page worth of that column). And where the second column will end. Then do whatever manipulations you want to your text to fit it on the way you want. The Page Finder will always tell you quickly where the columns will begin and end.

Immediately after the final line of the first column, put an embedded format code to shift the margin to, say, 40 (^M40). If there is no paragraph break between the bottom of column 1 and the top of column 2, then there should be no linespaces between the lines, otherwise the bottom line of column 1 will not be justified (see 5.16 for the reasons why.). For example, in a situation where there was no paragraph break between columns, the screen would look something like this

```
these are the bottom lines of column
1, but part of a single paragraph
continuing into the start of column 2,
^M40
and this is the start of column 2,
continuing the same paragraph so the last
line of column 1 should be justified.
```

(A line space would be inserted before the ^M40 line, if there was a paragraph break between columns.)

Now return to the Format menu, set the One Page parameter to 1 (see 5.17) and hit P. The first column will print to the bottom of the page and then there'll be a pause in the printing (caused by "One page") . At that point, remove the paper and reinsert it or just back it up to the start. Then hit BREAK or ENTER to restart the printout. The ^M40 command will be encountered first so that the printer will now space over 40 spaces at the start of each line, effectively giving you a second column.

You may need to do a little playing with this to get it to come out just right (especially re-positioning the paper to get the 2nd column just right). You may also use the partial print feature (5.10) to print out a column at a time, in place of using a New page command.

Appendix B : Pitfalls

Though every effort has been made to make Telewriter-64 "crash-proof" (one of the more important features of a word processor), there are situations which arise accidentally that may give the appearance of a system crash. There are other pitfalls to watch out for as well. These will be briefly discussed in this Appendix.

1) BE CAREFUL

The Read in and Create commands in the Main menu can destroy all the text currently in the buffer. They are both provided with an extra level of security (the "Sure?" prompt), but it's still possible to bull ahead and execute the command, only to regret it moments later. There is only one solution -- think twice before doing a Read in or Create.

This is even more true of the Block delete command (Clear-X), because it does not have an extra level of protection. Since the End Marker for Block Delete is also used for other block operations (partial save, partial print, block copy, partial word/line count) you need to be careful not to hit Clear-X while the End Marker is set for another operation. Whenever possible, try to execute the desired command immediately after setting the End Marker. If you set it, and then decide not to do the operation, delete it immediately.

When doing the Block Copy operation, it is good practice (though not always the most convenient one) to set the Begin Marker first. Once this is set, the Block delete operation will not work.

Also watch out for using the Line Kill (Clear-K) command in the middle of a line. After the first kill, this will usually make the current line into an overflow line and the next Clear-K will delete the entire overflow line, not just the current screen line as you might expect. Use the End of Line or Down Cursor command to make sure you know just what the Clear-K command will delete (though, even if you do blow it here, you won't lose too much).

2) CLEAR KEY

A minor problem arises if you accidentally hit the Clear key while inputting to a menu item (e.g. giving a filename to an I/O command or typing in a format parameter). If Clear is accidentally hit at any of these times, the screen will be Cleared (as happens in BASIC). Though it can make you feel totally lost, the solution is quite simple: Just hit the BREAK key and you'll be returned to the menu.

A good rule of thumb is "When in doubt, hit BREAK" (except, of course, when you're in the editor where this merely deletes the current character).

Note that you can get the same blank screen (and accompanying confusion) if you accidentally hit the Clear-F or Clear-G command without realizing it. The solution is the same -- BREAK.

3) QUEUE

If the Queue parameter is set to a non-zero number (when you're not actually chain printing a series of files) problems will arise. Specifically, if you do a Print command with the Queue value at non-zero, then, after the print finishes, the program will go look to the cassette for the next file to be read in and printed. If this happens, the filesearch "S" will appear at the upper left corner of the otherwise blank screen.

You can hit all the keys you want and not get out of this situation. The solution is simply to turn on the recorder and abort the search (3.1, 3.4).

Try to first position the tape so that it is at some data but not at the beginning of a file. Thus, when you turn on the recorder, the S will be blinking and the BREAK key will abort the filesearch. If you are at the beginnig of a file, part of it will be read in and overwrite the beginning of the text buffer. If you stop it fast though, you will not lose too much. This is one good reason to always save a file before printing it out.

Again, if the Queue value is set to a non-zero number and you do a Read in or Append, the program will immediately go to print it out once the read in is finished. This will not announce itself, and the screen will remain as it was while the file was reading in. If your printer is not on line, you will have no way of knowing what's wrong. The screen will just sit there with F "filename" at the top. The solution here is to connect your printer, turn it on, and put it on line, and then abort the printout (5.9) (this will work even if the baud rate is wrong). You'll be returned to the Main menu with no harm done except a line or two printed out that you probably didn't want.

The real solution to these problems, though, is to avoid them. Always make sure that the Queue parameter is set to 0 when you are not chain printing.

4) CLEAR-D

If you accidentally hit Clear-D, this will disable wordwrap, and, unless you explicitly insert Carriage Returns (ENTER), all the lines you type will be overflow lines. This can severely slow down certain editor features (as described in Appendix B) as well as making the text look strange (words broken in the middle at the end of screen lines). If this starts to happen, hit Clear-W to re-enable wordwrap. Then hit Clear-A to eliminate the massive continuation line(s).

5) AUTO-RETRY

If you start searching a tape in the middle of a file, the Auto-retry feature will usually find the next valid file. Occasionally it will miss one. This is a synchronization problem in the BASIC part of the cassette handling routines (not the Telewriter part) and it can happen with SKIPF or CLOAD as well. If it happens and you know there should be a file there that it's missing, try going back and starting the tape from different places (closer to where you think the file is, if possible).

6) SALVAGING PART OF A FILE WITH AN I/O ERR

You may save a file, and then, later, when you go to Read it in (or Append it) find that part way through you get an I/O ERR. The first thing to do is to try it again (maybe lowering the cassette volume control a little). If it continues to give I/O ERR this probably means that there's something physically wrong with the tape at the point where the error occurs. If this is the case, the file is lost. There is, however, a way to salvage what's in the file upto the point of the error.

First, fill the text buffer to the brim with anything -- find a big file you have on tape or else type or read in some text and then keep doing block copies of the entire buffer until it's full. Then go back and Read in the file that gave the I/O ERR. When the error occurs, return to the menu as usual with the BREAK key, and then go back to the editor. Hit the Clear Up Arrow command, and, when the screen rewrites, the beginning of the Error causing file will appear in the buffer up until the garbage data that caused the error. Following this for the rest of the buffer, will be the dummy text you filled it with in the first place. Delete the garbage and resave the file. You've salvaged as much as you can.

Verifying a file (with the V menu command - 3.6) after you save it can help you avoid this problem.

7) BREAK

Remember, when things look lost, hit BREAK, turn on your cassette, turn on your printer.

TELEWRITER-64 Command Reference card

EDITOR COMMANDS

One key commands:

ENTER = Terminate line (Carriage return)
BREAK = Delete character at cursor
Up Arrow = cursor up 1 line at left margin
Down Arrow = cursor down 1 line at left margin
Right Arrow = cursor right one character
Left Arrow = cursor left one character

Commands preceded by CLEAR ("control") key:

A = Align text lines to fit line width
B = Begin Text Block Marker
C = Copy block
D = Disable wordwrap
E = End Text Block marker
F = Find a search pattern
G = Global (Selective) Search and Replace
K = Kill line
M = Main menu (return to)
N = Next instance of search pattern
P = Page forward through text
Q = Search for special characters
R = Replace search pattern with replace pattern
S = Speed mode
U = Usual mode (exit Speed mode)
V = Vertical Tab (Page Finder)
W = Wordwrap mode enable
X = Block delete
Z = Delete all Block markers (Begin & End)
- = Page backward through text
0 = 51 column display mode
: = 64 column display mode
@ = 85 column display mode
; = toggle character set

ENTER = tab
BREAK = delete character before cursor
Up Arrow = cursor to top of text
Down Arrow = cursor to bottom of text
Right Arrow = cursor to end of line
Left Arrow = cursor to beginning of line

. = embedded format code
/ = underline delimiter for MX-80 only
, = backslash
1-9 = user defineable control codes

MAIN MENU COMMANDS (one key)

N = New text file created (destroys old)
E = Jump back to Editor (non-destructive)
S = Save all text in buffer to tape.
X = Save marked block of text to tape
R = Read in text file from tape
A = Append text file from tape to end of buffer
V = Verify (Skipf) Skip to end of file on tape
C = Characters / Line - Sets Linewidth
F = Jump to Print/Format menu
W = Word and line count for all or part of text

FORMAT MENU COMMANDS

S = Line Spacing
M = Left margin
C = Chars per Line
U = Upper (top of page) margin
L = Lines per page (usually 66)
B = Bottom margin
F = Font (MX-80 only) (see table below)
X = Baud (Xmit) rate (see table below)
P = Print whole text buffer
X = Print marked block of text buffer
R = Return to Main menu
D = Direct (ascii/control code) output to printer
T = Typewriter (keyboard chars direct to printer)
N = Number pages (0= no; val = start page number)
W = Where to position Page Number
Q = Chain print (Queue) files (val = # of files)
E = Epson=1/2; Oki/Centronics = 4; LF=5)
J = Justify (0 = no, 1 = Yes)
O = One page - pause at bottom (1= wait, 0= don't wait)

EMBEDDED COMMANDS

^S = Line Spacing
^M = Left margin
^C = Chars per Line
^U = Upper margin
^B = Bottom margin
^L = Lines per page
^N = New page (optional val= start pg #)
^D = Define direct code output
^DP = Define printing direct code
^H = Define Header
^H+ = Immediate Header
^! = Center Line
^= = Center Line of different size font
^; = Don't Justify or align block
^T = Tab stops if 1st Line
^T = Non printing comment - if not 1st line
^Ttext = Print flush left, don't align (no SP after T)
^Q = Load & print disk file (chain print)

EPSON FONT TABLE

0 = clear special fonts (normal font)
1 = ephasized (ESC E)
2 = double (ESC G)
3 = condensed (SI)
4 = enlarged (SO)

BAUD RATE TABLE

110	498
120	458
300	180
600	87
1200	41
2400	18
4800	6