TI-WRITER©

SUPPLEMENT

FORWARD

Purchasers of this TI-WRITER SUPPLEMENT may find the size of this publication somewhat disappointing if all they are shopping for is volume. My purpose in releasing this supplement is to show users of TI-WRITER, and it's clones, what possibilities exist beyond word processing.

Although similar information may appear in several of the articles, each work contains unique elements which warranted it's inclusion in this supplement. This reference is simply a minuscule representation of the quality, and quantity, of articles written on TI-WRITER by fellow TIers. Additional articles and/or information on TI-WRITER should be available from your local users group.

If there are additons, that you feel may benefit future readers, please bring them to my attention. All correspondence can be addressed to:

TI-WRITER SUPPLEMENT EDITOR
C/O CHICAGO TI USER'S GROUP
P.O. BOX 578341
CHICAGO, IL. 60657

All material contained within this guide has appeared in newsletters received by CTIUG as part of our newsletter exchange program. The Chicago TI User's Group claims no rights of ownership, whatsoever, to any of the articles contained within this guide. If an author does not wish their works to appear in this reference, please inform me and they will be removed from any future editions. To the best of our knowledge none of the articles, contained herein, are copyrighted and they have been authored with the express desire to share them with other members of the TI community. As such, and is our policy, other user groups are encouraged to copy and distribute this information.

TABLE OF CONTENTS

THE EDITOR

TI-WRITER OVERLAY OVERVIEW.....TOM KENNEDY

TI-WRITER MNEMONIC (MEMORY) TRICKS.....UG OF ORANGE COUNTY

A HANDY QUICK REFERENCE TRICK.....FRED DENNIS

TI-WRITER COMMANDS.....JERRY MAXFIELD

THE FORMATTER

TI-WRITER FORMATTER OVERVIEW.....TOM KENNEDY

MORE TI-WRITER TIPS.....BARB BERG

DATE? WHAT DATE?.....GLEN DAVIS

GRAPHICS

PRODUCING ART WITH THE WORD PROCESSOR-PART ONE.....ANNE DHEIN

PRODUCING ART WITH THE WORD PROCESSOR-PART TWO.....ANNE DHEIN

TI-WRITER GRAPHICS.....JAMES STRINGFELLOW

TI-WRITER GRAPHICS WITH THE PROWRITER.....BARB BERG

PROGRAMS, PRINTERS & MISCELLANY

PRINTER COMMANDS.....BRAZOS VALLEY 99'ERS

PRINTING MAILING LABELS WITH TI-WRITER.....JERRY KEISLER

MAKING ADDRESS LABELS WITH TI-WRITER.....KARL BARNA

XB PROGRAM EDITING WITH TI-WRITER.....JONATHAN BLACK

GRAPHS USING TI-WRITER.....JACK COLEMAN

TI-WRITER FONT MAKER.....JAMES STRINGFELLOW

DIS/VAR CONVERTER PROGRAM.....JAMES STRINGFELLOW

A PROGRAM TO HELP TI-WRITER USERS WHEN DOING FORM LETTERS....BILL HARMS

WORDCOUNT.....B. DAVIS

DOWNLOAD CHARACTERS TO YOUR GEMINI.....JIM PATTISON

TI-WRITER TEXT SORTER.....JERRY KEISLER

LETTERFORM.....OLLIE HERBERT

TELECOMMUNICATIONS WITH TI-WRITER.....MAGNETIC UG

HINTS, TIPS & ANSWERS.....BILL SPONCHIA

TIWRITER OVERLAY OVERVIEW
by Tom Kennedy

How many of you have a typewriter, please raise your hand. Keep your hand up if your typewriter has interchangeable text. How about automatic bold and underline? Or some amount of memory storage (for letter heads, etc.)? How about an erase key? Those of you left have probably got a pretty expensive piece of machinery, but TI-WRITER has ten times the functions, or features of the best typewriters. With TI-WRITER, your only limitation is your own creativity.

To start off with, what will you need to operate your Word Processor? You must have the 99/4A console (TI-WRITER won't work with the 99/4), a TV or monitor, the cartridge and disk package, the disk system, memory expansion, the RS232 interface, and a printer. In other words, the whole works. The printer is something you definitely want to be careful in choosing because all of your work will be in vain if you can't print out exactly what you type in, and with an attractive appearance. First, let's look at the command line. That's the line at the top of the screen when you're in the command mode. There are seven commands shown and sixteen sub-commands that are options of the main seven. The commands are selected by typing only the letters that are capitalized in the word. For instance: "F" for Files, "SH" for SearcH, or "LF" for Load File. That's an interesting point: you can access any of the sub-commands from the main command menu. In other words, to ShowDirectory (which is a disk catalog) you would enter the command mode, (FCTN 9), and either type "F" for files, and "SD" for ShowDirectory, or just type "SD" immediately. This feature saves a lot of time and keystrokes.

The first command is Edit. This simply enters you into the text-edit mode in which text is created.

Next is Tabs. When you hit "T", the top part of your text is shown with a scale across the top showing the current tabs and margins. Changes are made by simply typing over existing entries with the appropriate symbol (L,R,T, or I).

"F" for files allows you to work with your text file as a whole. To Load, Save, Delete, Print, Purge, or ShowDirectory. "PF" for print file is not what you'll get when you print out through the text formatter; it just prints a "hard copy" of the whole file, just as you see it on the screen. It doesn't print with any of the modifications made by the format commands (more on those later). "PF" is useful for making a fast copy of a long letter, or whatever, in order to check for errors without having to scroll back and forth or up and down. Purge simply erases the file from memory to prepare for a new entry. It is similar to the "NEW" command in BASIC.

Next is "L" for Lines. This allows you to work with whole lines or groups of lines by moving them to somewhere else in the text, copying to somewhere else and leaving the original intact, to delete groups of lines, or to quickly move the cursor to some line in the text with the ShowLines option.

Search (or "SH") gives you the option of either the FindString routine or the ReplaceString routine. FindString will move the cursor to the first and/or each successive use of the word string you give. ReplaceString searches the text for a given string and replaces all or one occurrence with the new string. This is great for correcting a repetitive spelling error.

RecoverEdit is a failsafe repair in case the text buffer was purged in either the File or Quit command. It will pull back everything but the first line and restore the file. I guess the loss of the first line is the penalty paid for accidentally erasing a file, which can't be done very easily.

Finally, Quit, as the name implies, blows it all apart and leaves you with the title frame. But before it goes, all open files are closed (such as to disk or printer) so no data is lost. Fortunately, it first gives you the option of saving your file (in case you forgot to do that already) or just purging the file and going back to the edit mode. But if you really want to quit, you type "E" for Exit and it shuts down.

Now let's go over the keyboard. TI-WRITER makes extensive use of the FCTN and CTRL keys and uses every possible function of the top line of keys (the numbers). There are also many functions that have duplicate methods of keystrokes to activate them. For instance, to enter the command mode, you either press FCTN 9 or CTRL C. The reason for this duplication is to allow you to choose which is easiest to use depending on where your fingers are at. The problem though, is that it can be very confusing trying to remember the fifty different key combinations that activate the thirty functions. A better method is to just pick which keys you're going to use for what function and ignore the rest. What I do is use the number line keys for anything shown on the overlay strip and just memorize the few functions hidden down in the keyboard. Let's start by going down the overlay strip, left to right as shown on the next page.

The last four key functions to mention are the cursor arrows: UP, DOWN, LEFT, & RIGHT. These stay the same as in console BASIC. Now, if you're still following along you may be quite confused with this onslaught of information. The point is, you can't learn all of this in one sitting, but after using TI-WRITER for a while you start to pick things up as you need them. Rest assured, you do spend the majority of your time typing. The purpose of most of the functions I've mentioned are to manipulate the text which is already in the file. I have simply tried to cover all of this in order to bring something to your attention that you might have missed, or to peak your interest in the capability of the TI-WRITER software.

To review, in the command mode we can choose between Edit, Tabs, Files, Lines, SearcH, RecoverEdit, or Quit. As sub-commands of those seven, we can choose Load File, Save File, Print File, Delete File, Purge, ShowDirectory, Move Lines, Copy Lines, Delete Lines, Showlines, FindString, ReplaceString, or Exit.

OOPS!	CTRL 1 (CTRL Z)	This can be a real lifesaver. It recovers, or "backs up" a functio that you didn't mean to hit. Like if you goofed and hit "Delete Line" instead of "Insert Character", hit "OOPS!" and the line comes back.
Del Char	FCTN 1 (CTRL F)	This is the same as "DEL" in console BASIC. It deletes one character under the cursor and pulls the rest of the line up to fill.
Reformat	CTRL 2 (CTRL R)	This is used to close up the text after using Insert Character. It deletes all spaces between the cursor and the next word in the text Then it draws all subsequent words up through the paragraph until it encounters a Carriage Return.
Ins Char	FCTN 2 (CTRL G)	In Word Wrap mode (solid cursor), 32 blank characters are inserted after the cursor. The bulk of the text is pushed down the line. After insertion of new text,hit Reformat. Any remaining spaces are removed. In the Fixed mode (hollow cursor), this operates the same as BASIC.
Screen Color	CTRL 3	This allows you to choose which of the five color combinations of text/screen you prefer. The default, for no good reason, is white on dark blue. This is hard on the eyes. I prefer to turn down the color on my monitor and use either black on green or black on light blue.
Del Line	FCTN 3 (CTRL N)	Deletes the entire line that the cursor is on, including the space of the line.
Next Paragraph	CRTL 4 (CTRL J)	This advances the cursor to the beginning of the following paragraph and puts the first line at the top of the page.
Roll Down	FCTN 4	This is a "vertical block scroll", meaning the next 24 lines of text of text are shown. Scans quickly down the text to get to some point.
Dupe Line	CTRL 5	Creates a duplicate below of the line the cursor is on. The Move/Copy function can do the same, but this key makes it faster and easier to create repetitive lines such as a double row of '*'s under a title.
Next Window	FCTN 5	A "horizontal block scroll". It jumps across to display the next block of 40 characters, in increments of 20. For example, the screen starts out on column one to forty, then twenty to sixty, then forty to eighty.
Last Paragraph	CTRL 6 (CTRL H)	The opposite of "Next Paragraph"
Roll up	FCTN 6 (CTRL B)	The opposite of "Roll Down"
Word Tab	CTRL 7 (CTRL W)	This moves the cursor down the line to the first letter of each word.
Tab	FCTN 7 (CTRL I)	Just like on a typewriter, this moves the cursor to next setting, defined using the Tab function on the command line.
New Paragraph	CTRL 8	Places Carriage Return at end of current line, then skips down to next line. If you have preset an auto-indent, (by using an "I" in Tabs) then it also indents over to the proper column.
Ins Line	FCTN 8 (CTRL O)	Inserts a blank line above the line the cursor is on.
New Page	CTRL 9	Inserts a blank line with a Np and Cr symbol at the beginning. This causes the printer to feed to the next page.
Command/ Escape	FCTN 9 (CTRL C)	This is how to exit from the edit mode to get to the command line. It is also used to cancel a command already in progress.
Word Wrap	CTRL 0	Switches from the "Word Wrap" mode to the "Fixed" mode. In Word Wrap, upon reaching the end of the line the cursor jumps to the nest line.
*	*	If you're in the middle of a word at the end of the line, the word you were on moves down too. This allows you to just type continuously without looking up to see when to hit enter. In the fixed mode, when you reach the end of the line your letters just pile on top of each other and you hit enter to move to the next line.
Line Numbers	FCTN 0	This removes or displays the four-digit line numbers at the left side of the screen. The numbers are used for reference when manipulating blocks or lines of text, just like when editing a BASIC program, line numbers are needed to refer to where changes will be made.
Quit	FCTN =	Quit is the same as in console BASIC. Use Quit option of the Command line to safely exit TI-WRITER.
Back Tab	CTRL T	The same as Tab except it backs up one setting.
Beginning of Line	CTRL V	Moves the cursor to the beginning of the line you're on.
Del.End of Line	CTRL K	This is just like Delete Character (FCTN 1), except it takes out everything to the right of the cursor.
Home Cursor	CTRL L	Moves the cursor to row 1, column 1, on the screen only. Unfortunately it doesn't move to first line of text, which would be more convenien when at the end of a long document and want to jump to the top. [Fo that, enter S, then enter 1.]
Left Mrgn Release	CTRL Y	Allows you to temporarily back-arrow beyond the left margin when it has been set past zero.

REPRINTED FROM THE ROM NEWSLETTER OF USERS GROUP OF ORANGE COUNTY, OCTOBER 1985

From Out 'N About

TI WRITER MNEMONIC (MEMORY) TRICKS

CTRL	MNEMONIC	FUNCTION	ALTERNATE
A	ADVANCE DOWN	ROLL DOWN	F4
B	BACK UP	ROLL UP	F6
C	COMMAND MODE	COMMAND MODE	F9
F	FLYAWAY CHARACTER	DELETE CHARACTER	F1
G	GET A HOLE FOR CHAR	INSERT CHARACTER	F2
H	HOP BACK TO LAST	LAST PARAGRAPH	C6
I	INDENT	TAB	F7
J	JUMP TO NEXT	NEXT PARAGRAPH	C4
K	KILL TO END OF LINE	DELETE TO END OF LINE	--
L	LEAP HOME	HOME CURSOR	--
M	MAKE NEW PARAGRAPH	NEW PARAGRAPH	C8
N	NO MORE LINE	DELETE LINE	F3
O	OPEN BLANK LINE	INSERT BLANK LINE	F8
P	PAGE BEGINNING	NEW PAGE	C9
R	REFORMAT	REFORMAT	C2
T	TAB BACK	BACK TAB	--
V	VEER TO LEFT	CURSOR TO LINE START	--
W	WORD TAB	WORD TAB	C7
Y	YANK MARGIN CONTROL	LEFT MARGIN RELEASE	--
Z	ZIP BACK	OOPS!!	C1
-	----------	SCREEN COLOR	C3
-	----------	DUPE LINE	C5
-	----------	NEXT WINDOW -->	F5
-	----------	WORD WRAP	CØ
-	----------	LINE NUMBERS OFF/ON	FØ

NOTE: THE ARROW KEYS WORK THE SAME WITH EITHER THE CONTROL KEY OR THE FUNCTION KEY

1	2	3	4	5	6	7	8	9	Ø	
OOPS!	REFORMAT	SCREEN COLOR	NEXT PARAGRAPH	DUPE LINE	LAST PARAGRAPH	WORD TAB	NEW PARAGRAPH	NEW PAGE	WORD WRAP	
DEL CHAR	INS CHAR	DEL LINE	ROLL DOWN ↓	NEXT WINDOW→	ROLL UP ↑	TAB	INS LINE	COMMAND/ ESCAPE	LINE #'s	QUIT

A HANDY QUICK REFERENCE TRICK.....

by Fred Dennis

Quite often when I am typing in TI-WRITER I wish I had a quick reference for the most commonly used features of my Panasonic printer such as italics, underlined, compressed, expanded, italics, etc. with the Editor keystrokes needed to control them. To make things easier I whipped up a little table, printed it out as a quick reference card and laminated it. That worked fine until it started getting lost in the pile of papers on my desk every time I needed it. It was then I remembered a nice feature of TI-WRITER that lets me load in a second text file and append it to the one I am editing. Now when I am editing or creating a document, I load in my little quick reference file using the LF command, typing in the last line number of the file I'm editing before the filename to load, in the form:

```
LF
54 DSK1.PANASONIC
```

(i.e. "Load DSK1.PANASONIC after line 54 in the text buffer")

The reference table is always at the end of the file I am editing and can be quickly viewed by scrolling down with FCTN-4. I have found this method easier than locating my little card when I am in a hurry, and could be applied to numerous other applications as well. I hope you fined some additional uses for this technique and share them with other TI users.

TI-WRITER KEYSTROKES
FOR PANASONIC KXP1091 PRINTERS

PRINTER FEATURE	KEYSTROKE SEQUENCE
Set pica pitch	CTRL-U,FCTN-R,CTRL-U,"P"
Set elite pitch	CTRL-U,FCTN-R,CTRL-U,"M"
Set proportional pitch	CTRL-U,FCTN-R,CTRL-U,"o"
Set compressed printing	CTRL-U,SHFT-O,CTRL-U
Release compressed printing	CTRL-U,SHFT-R,CTRL-U
Set one-line double-width	CTRL-U,SHFT-N,CTRL-U
Release one-line double-width	CTRL-U,SHFT-T,CTRL-U
Set double-width printing	CTRL-U,FCTN-R,CTRL-U,"W","1"
Release double-width printing	CTRL-U,FCTN-R,CTRL-U,"W","0"
Set superscript mode	CTRL-U,FCTN-R,CTRL-U,"S","0"
Set subscript mode	CTRL-U,FCTN-R,CTRL-U,"S","1"
Release sub/superscript mode	CTRL-U,FCTN-R,CTRL-U,"T"
Set italics mode	CTRL-U,FCTN-R,CTRL-U,"4"
Release italics mode	CTRL-U,FCTN-R,CTRL-U,"5"
Select internat'l char. set	CTRL-U,FCTN-R,CTRL-U,"R",(n)
Set emphasized mode	CTRL-U,FCTN-R,CTRL-U,"E"
Release emphasized mode	CTRL-U,FCTN-R,CTRL-U,"F"
Set double-strike mode	CTRL-U,FCTN-R,CTRL-U,"G"
Release double-strike mode	CTRL-U,FCTN-R,CTRL-U,"H"
Set NLQ print mode	CTRL-U,FCTN-R,CTRL-U,"n"
Set underline mode	CTRL-U,FCTN-R,CTRL-U,"-","1"
Release underline mode	CTRL-U,FCTN-R,CTRL-U,"-","0"

THE NORTHEAST IOWA HOME COMPUTER USERS GROUP

TI-WRITER COMMANDS

Loaded from the **DECATUR HC USERS GROUP** V 7 P 4 by TERRY MAXFIELD 8/13/85

EDITOR COMMAND	FCTN	CTRL	EDITOR COMMAND	FCTN	CTRL	EDITOR COMMAND	FCTN	CTRL
BACK TAB		T	INS. BLANK LINE	8	0	QUIT	=	
BEGINNING LINE		V	INSERT CHARACTER	2	G	REFORMAT		2orR
COMMAND ESCAPE	9	C	LAST PARAGRAPH		6orH	RIGHT ARROW	D	D
DEL CHARACTER	1	F	LEFT ARROW	S	S	ROLL DOWN	4	A
" END OF LINE		K	LEFT MARGIN REL.		Y	ROLL UP	6	B
DELETE LINE	3	N	NEW PAGE		9orP	SCREEN COLOR		3
LINE #'S OFF	0		NEW PARAGRAPH		8orM	TAB	7	I
DOWN ARROW	X	A	NEXT PARAGRAPH		4orJ	UP ARROW	E	E
DUPLICATE LINE		5	NEXT WINDOW	5		WORD TAB		7orW
HOME CURSOR		L	OOPS!		1orZ	WORD WRAP/FIXED		0

LOAD FILES = LF (enter) DSK1.FILENAME (load entire file)
LF (enter) 3DSK1.FILENAME (merges filename with data in memory after line 3)
LF (enter) 3 1 10 DSK1.FILENAME (lines 1 thru 10 of filename are merged after line 3 in memory)
LF (enter) 1 10 DSK1.FILENAME (loads 1 thru 10 of filename)

SAVE FILES = SF (enter) DSK1.FILENAME (save entire file)
SF (enter) 1 10 DSK1.FILENAME (saves line 1 thru 10)

PRINT FILES= PF (enter) PIO (print control character and line numbers)
PF (enter) C PIO (prints with no control characters)
PF (enter) 1 PIO (prints 74 characters with line numbers)
PF (enter) F PIO (fixed 80 format)
PF (enter) 1 10 PIO (prints lines 1 thru 10)

CANCEL PRINT FILES= FCTN 4
NOTE: If your printer uses RS232 switch PIO with RS232.

DELETE FILES= DF (enter) DSK1.FILENAME

SETTING MARGINS AND TABS (16 tabs maximum)
L - Left margin R - Right margin I - Indent T - tab
Use ENTER to execute or COMMAND/ESCAPE to terminate command

RECOVER EDIT= RE (enter) Y or N

EDIT = E (enter) (enter edit mode)

LINE MOVE = M (enter) 2 6 10 (moves lines 2 thru 3 after line 10)
M (enter) 2 2 10 (moves line 2 after line 10)

COPY = Same as move except use C instead of M

FIND STRING = FS (enter) /string/ (will find string)
FS (enter) 1 15 /string/ (will find string in lines 2 thru 15)

DELETE = D (enter) 10 15 (deletes line 10 thru 15)

TIWRITER FORMATTER OVERVIEW

by Tom Kennedy

Now I want to cover the Text Formatter, which prints out the document. Most importantly, the special symbols, called Format Commands, that the formatter uses to alter the print-out of the document, which are installed in the Text Editor.

In other words, you put these commands into the text when you write it and as the formatter comes across them it changes the text accordingly but doesn't actually print the symbols.

There are six groups of formatter commands that are all applied in a similar manner. All commands must be in caps and must be on a line that starts with a period.

The use of these commands in your text is what separates the word processor from a typewriter. They allow you to get the most out of your printer.

So, now you've written your document, and inserted all the format commands, now how do you print it out? First, save the document and exit the Text Editor. At the title menu, select Text formatter, (make sure the program disk is in the drive) and the screen will blank with the prompt "ENTER INPUT FILENAME". Enter the name of the file you just saved, (ex. DSK1.MYFILE) and hit enter.

Next, the prompt "ENTER PRINT DEVICENAME" appears after the file is loaded. If you use a serial printer, the device name would be RS232.BA=xxx with xxx being the baud rate. If you're using a parallel printer, the device name is PIO. Also, you must add either .CR or .LF to the end of the device name. This tells TI-Writer whether your printer will handle the carriage return or the line feed. Check your printer manual and the TI-Writer manual in detail to find out which you use.

The next prompt is "USE MAILING LIST". If you aren't printing "form letters" just hit enter to accept the default of N (NO).

Next is "WHAT PAGE(S)? <ALL>. If you want to print the whole document, accept the default for all pages. Otherwise, you can print any of the pages or groups of pages.

The prompt "NUMBER OF COPIES: 1" tells how many copies of each page are to be printed.

The last prompt is "PAUSE AT END OF PAGE? N". The main purpose of this function is if you are using separate sheets of paper it will stop and wait for you to align the next sheet.

Now, about the Mailing List Option. Let's say you've written a form letter to send out to various individuals, maybe a resume'. You write the letter like normal, but when you come to a name or address or something that will change with each letter, you put in its place a variable in the form of *n*, where n is a number to identify the order. So instead of starting off with:"Dear Mr. Smith" you would have "Dear Mr. ^1*" and so on. when you're all through with your letter, save it and purge the memory. Now you must create what is called a Value File, which is your mailing list where TI-Writer will draw the variables from. A value file consists of a list values to be inserted into the letter, listed one to a line, preceded by the number of the variable and ending with a carriage return symbol. Groups of values must be separated by a line with just an asterisk and a carriage return. For example:

```
1 John Smith
2 123 STREET
3 Seattle, WA
  *
1 Jane Doe
2 456 STREET
3 Seattle, WA
```

At the top of your letter you insert the .ML f command where f equals the filename of your value file. After selecting the mailing list option the computer will use this command to fill in the variables. If there is no .ML command in the letter then when you are prompted for "MAILING LIST NAME:" you supply the filename. This allows you to call on a number of files for different groups.

Text Dimension commands, as the name implies, move or shape the words in the document (margins, linespacing, right justify, etc.)

```
 .FI   : FILL      : PUTS AS MANY WORDS ON A LINE AS WILL FIT.
 .NF   : NO FILL   : CANCELS FILL.
 .AD   : ADJUST    : ALIGNS THE TEXT TO THE LEFT AND RIGHT MARGINS. (RT. JUSTIFY)
 .NA   : NO ADJUST: CANCELS ADJUST.
 .LM n : LF MARGIN: SETS LEFT MARGIN TO "n".
 .RM n : RT MARGIN: SETS RIGHT MARGIN TO "n".
 .IN n : INDENT    : CREATES AN AUTO-INDENT FROM LEFT MARGIN.
 .LS n : LINE SP   : SETS LINE SPACING TO "n" LINES.
 .PL n : PG LENGTH: DEFINES NUMBER OF LINES TO A PAGE.
 .BP   : BEGIN PG : DEFINES FIRST LINE OF NEW PAGE.
```

Internal Format commands control the spacing of characters on a line.

```
 .SP n : SPACE     : SIMILAR TO THE TAB FUNCTION.
 .CE n : CENTER    : CENTERS NEXT "n" LINES BETWEEN MARGINS.
```

Highlighting commands control functions such as underline or bold and allow you to redefine characters to use them to send CTRL codes to the printer.

```
  ~     : REQUIRED : JOINS WORDS TOGETHER WHEN REQUIRED TO PREVENT SPLITTING IN
        :   SPACE  :      REFORMATING, UNDERLINE, ETC.
  &     : UNDERLINE: (UNDERSCORE) UNDERLINES ALL TEXT FOLLOWING UNTIL NEXT PACE.
  @     : BOLD     : (OVERSTRIKE) RETYPES FOLLOWING TEXT FOUR TIMES.
 .TL xx: TRANS-    : ALLOWS REASSIGNMENT OF ONE CHARACTER TO REPRESENT A NUMBER.
        : LITERATE : OF CHARACTER VALUES TO SEND CODES TO THE PRINTER.
 .CO t : COMMENT   : SIMILAR TO REM IN BASIC--ALLOWS NOTES THAT DONT PRINT.
```

Page identification commands print notes in the upper or lower corner of each page, either headers or footers.

```
 .HE t : HEADER    : PRINTS TEXT (t) AND PAGE NUMBER AT TOP OF EACH PAGE.
 .FO t : FOOTER    : PRINTS TEXT (t) AND PAGE NUMBER AT BOTTOM OF EACH PAGE.
 .PA   : PAGE #    : RESETS PAGE NUMBER IN .HE AND .FO
```

File management commands

```
 .IF f : INCLUDE   : MERGES A FILE TO PRINT A DOCUMENT TOO LARGE FOR ONE FILE.
       :  FILE     :
```

Mail Merge option commands are used to supply values to the variables in a letter that has been set up for the mail merge option

```
 .ML f  :MAIL LIST: IDENTIFIES VALUE FILE (f) FOR MAIL LIST.
 *n*    :VARIABLE : INSERTED IN TEXT AS VARIABLE FOR ASSIGNMENT FROM VALUE FILE.
 .DP n:t:DISPLAY  : PROMPTS YOU USING TEXT "t" TO ASSIGN TO VARIABLE (*n*).
       :  PROMPT  :
```

****

Another way to insert values is to use the Define Prompt command. With this command you do not insert a .ML comand calling a value file and instead you insert lines containing the format: .DP n:t - where n is the number of the variable and t is the prompt text. Now, when you come to the prompt "USE MAILING LIST?" you select "N" for NO and as the document is printed when a variable is encountered the printing stops and the text you chose appears on the screen asking you for the appropriate value. If you don't include a ".DP n:t" command in your text, the computer responds with "ENTER DATA FOR VARIABLE *n*" and it can get confusing trying to remember which item you're on. This method is handy for letters which you only want to print one copy at different times to different people.

Let me tell you, this is why I bought a computer. I'm sure we all went through that period of time before buying a computer when we would ask: "what am I going to use a computer for, anyway?". Well I decided there were two things I wanted to do: 1) Store files of data (recipes, albums, etc.) and 2) Use my computer as a typewriter. I didn't know about TI-WRITER when I bought the 99/4A, but now I know that I made the best choice possible. I hope you will all find TI-WRITER as easy to use and as powerful as I have.

MORE TI-WRITER TIPS

By Barb Berg

Since I use TI-WRITER all the time, I have learned how to use it extensively and would like to pass on a few tips from my experience. Some of these apply only to the Prowriter while others are useful for any printer. Please note that the exclama points at the beginning of example command lines were only put there so they would be printed instead of used as commands. Command lines that begin with a period should not have any other character before the period.

Want a header or footer in your document? You can do some interesting things with these commands. If you want to center your header or footer you have to use those carat signs for required spaces to put it in the center of your page. It's best to use the counting method here, that is, if your header or footer uses twenty characters and spaces, subtract twenty from eighty and divide the result by two. Now you know that you need thirty required spaces to center your header or footer.

Another thing I've done with a header is to put it in expanded print. I do this by including my expanded command right in the line with the header data. On the Prowriter, the command to do this is shift out, CHR$(14), and to cancel expanded is shift in, CHR$(15). To get shift out, I type CTRL U, SHIFT N, CTRL U, and to shift in is CTRL U, SHIFT O, CTRL U. Each command leaves a special character in the line which is recognized as a formatter command and should not be thought of as another space in the printed line. If you are centering an expanded header, type in the required spaces THEN add the special commands. Remember that the expanded print characters take up about twice as much space as regular sized characters. If you wish to do this, first count the number of characters in the header, then multiply that number by two. If the header has twenty characters, it will take up about forty columns in expanded print. Subtract forty from eighty and divide the result by two. You will need twenty required spaces to center a twenty character expanded print header or footer. Type in the header command with the required spaces, then insert your expanded print on command after the required spaces and before the characters of your header. Type the command to turn off expanded right after the last character in the header.

Now, if you want your header in expanded print and your document will be in condensed, like I've done with this article, you need to add another command in your header. The command for condensed print on the Prowriter is ESC Q and for pica, the default, it's ESC N. To make sure that my header is going to be in pica on each page and will be centered right with my required spaces, I need to add the pica command to the beginning of my header line and the condensed command at the end. So I inserted a CTRL U, FCTN R, CTRL U N before the carat signs and added a CTRL U, FCTN R, CTRL U Q at the end. The header for this article wound up looking like this:

```
! .HE ↑NPage % of^^^^^^^^^^^^↓MORE TI-WRITER TIPS↑^^^^^^^By Barb Berg↑Q
      ↑                      ↑                  ↑                  ↑
      Pica                expanded on        expanded off       condensed
```

Because my margins were set up for 132 characters per line, I also had to make sure my header line was before the line for my left and right margins. My document formatting commands are all on one line, like this:

```
! .PL 65;LM 0;RM 131;IN +2;FI;AD
```

UNDERLINING

This is what happens when you use the & sign to underline. Once the line is printed, this command sends the print head back to the characters to be underlined and prints the underline, character 95. This is a little difficult to read since the underline uses the same dots as the bottoms of the characters. The Prowriter has a graphics character that is an underline one step lower, character 128. To make your underlining a little easier to read, transliterate character 95 to the lower underline character 128, like this: .TL 95:128. Now your underlined characters are easier to read. You will need to use the required space character to underline a phrase like this one:

BETTER UNDERLINING

Now for a little trick I found when printing the article on TI-WRITER bit-dot graphics with the Prowriter. This should apply to all printers. Since the graphics file has to have a line feed at the end of each printed line when run through the formatter, printing a file in anything but pica, the default size of type, appears at the very least to be a lot of work. Since the screen in the editor allows only 80 characters per line, you would think that to get a line feed at the end of a line of condensed print you would have to count 132 characters and then put a line feed. There is an easier way to accomplish this.

First, set up your left and right margins in the document by typing the line .RM 0;LM 131;FI . You will need the fill command in order to get 132 characters on the line; without it the line will be printed as it appears in the document. Now, from the command line, set your editor margins using the TAB command. The left margin will be set to zero. To figure the right margin, take ha 132, or 66, and set your right margin to this number minus one, or 65, on the TAB setting line. You need to subtract one since the TAB line starts at 0 instead of 1. When you type in your document, you can use word wrap mode, as usual, and add your line feeds after you're finished. When you add the line feeds, go to fixed mode (CTRL 0) and put your line feed character at the end of every

other line, starting with the second line. Now you have your line feeds exactly where you want them, at the end of 132 characters! Be sure to put a line feed at the end of each paragraph. The Prowriter will allow you to have both a line feed character and carriage return character and will ignore the carriage return when the document is printed through the formatter with the device name of PIO.CR. If you leave the carriage return symbols in your document, you can reformat and only affect the desired paragraph.

You may find that when the document is printed, occasionally a word at the beginning of a printed line is indented by one space. I find this undesirable, since I like a nice, clean left margin, except for indenting at the beginning of a paragraph. This, too, can be removed. Once you have all the line feeds in their correct positions (you may want to print the file first and make sure they are correct), the editor's right margin can be set back to it's default setting of 79. If you have left in the carriage returns, you can now reformat each paragraph! This does not affect your line feeds; they are still exactly where you want them. But you can now check your printed document to find the places where a space was printed as the first character of a line and delete the space. When you print the document, your left margin looks like it should.

Also be aware that when using bit-dot graphics and custom line feed that you may need to adjust your page length with the .PL command. A full page of graphics will require a page length of 99. If you are only using one or two lines of custom line feed, then you may only need to add one or two lines to the default page length of 66. And you may find that when the formatter sends the top of form command at the end of that page the paper isn't positioned correctly. If this happens, and you can't seem to find the right page length to position it correctly, you will have to answer Y to the pause at end of page prompt on the formatter screen and position the paper at the end of that page before continuing.

Now for a Prowriter tip for using the graphics characters available on this printer. There are a nice variety of graphics in the range of characters 128-247 by using the ESC commands for graphics or Greek characters. These are accessed by the printer commands ESC # (for graphics) and ESC & (Greek). Even though some of you may have had difficulty accessing these characters in TI-WRITER files, it really isn't as hard as it seems.

Getting the graphics range is pretty easy, all you need to type in is CTRL U, FCTN R, CTRL U # at the beginning of your document. Then use transliterates to print the characters desired. Getting the Greek characters is not much more difficult, but it is tricky if you haven't read your TI-WRITER manual thouroughly.

Since the & sign is recognized by TI-WRITER as the formatter command to underline characters, the top of page 59 in the manual tells us that in order to print an &, you need to type it TWICE in order to print one of them. In order for TI-WRITER to recognize the & sign as an ampersand and not the underline command, two of them have to be typed in the document. So, to get the command ESC & to work you need to type two ampersands in the command also. In the document, you simply type CTRL U, FCTN R, CTRL U &&. And in case you're wondering how I got two ampersands to print in this document, I had to type four of them in a row. Therefore, for every & sign you want to print, type two and you've got it made. This applies to any printer.

The same is true if you want to type the at sign, @. You have to type two of them in order to get one printed. If you need the carat sign, ^, transliterate it to another character. For this document, I used the transliterate command .TL 92:94 to make the reverse slash into the carat sign and typed a reverse slash where I wanted the ^ printed.

Suppose you want to take two or more of the Prowriter's graphics characters and combine them to make one character. You have to set up your transliterate like this:

```
! .TL 36:128,8,136,8,148,8,151
```

This particular character consists of four separate characters, 128, _, 136, |, 148, ¯, and 151, ⌊. Character number 8 is the backspace command that allows all four characters to be printed in the same space. This transliterate is telling the printer that when character number 36, the dollar sign $, appears, then print character 128, backspace, print character 136, backspace, print character 148, backspace, then print character 151. But this is what happens when this transliterate is used by itself when you use PIO.CR as the device name in the formatter:

```
⌋¯|
```

If this happens, you need to add another command. In order to get the backspace command to work properly the printer has to be in incremental print mode. The default is logic seek mode. The command to set up incremental mode is ESC [, and to return to logic seek mode is ESC]. Simply add these commands right in the line itself before and after your special character.

```
¡[$¡]
```

You will also find that when you use PIO.CR as your device name that the underline and overstrike commands won't function properly. So if you are using these commands in a text file that will be printed up with this device name, you will need to have the incremental print mode in effect for them to work properly. These functions all work just fine if the device name is PIO.LF.

Now, suppose your document is in condensed mode, like this one, and you want to print a border or characters that will run the

complete width of the page without a gap somewhere in the line. If you are indenting, like I am, and want the same number of spaces at the end of the line so the border appears centered, then you will want a continuous border of 128 characters. My indent is 2 spaces so 128 characters gives me 2+128+2=132, the total line length in condensed.

The Prowriter has a command called repeat character. If you own another printer you will have to check your printer manual to see if it has a similar command. In TI-WRITER you access this command in this format: ESC Rnnnc where nnn is a three digit number representing the number of times the character is to be printed and c is the character to print. To print my border line of 128 hearts, I first transliterate the heart character, 233 to the plus sign, 43:

```
!  .TL 43:233
```

Then I use the repeat character command to print the border, like this:

```
↑R128+
```

Remember, the ESC character ↑ is acquired by typing this sequence: CTRL U, FCTN R, CTRL U. And the result looks like this:

♥♥♥

DATE? WHAT DATE?
by Glenn Davis

This may be a little trick you didn't know about before: Getting TI-WRITER to ask you what the date is on your letter. I keep a disk with letters that I send to people. For two reasons, really. So I remember what I said, and so I don't have to reenter the address, format codes, and other junk necessary to print the letter through the formatter.

One problem arose with this method, However. The date had to be changed when I created a new body or revised a letter to be sent later. So I began using TI-WRITER's "DEFINE PROMPT" formatter command to ask me the date when I print the letter out.

If you want to enter the date all on one line do this:

```
.CO * is redefined to print itself
.TL 94:42
STREET
CITY, ST**ZIIPP
~1~
```

On the other hand, you could use:

```
.DP 1 ENTER THE DAY
.DP 2 ENTER THE MONTH
.DP 3 ENTER THE YEAR
```

In your text header, write:

```
STREET
CITY, ST**ZIIPP
~2~ ~1~, ~3~
```

PRESTO! Date is on the letter. Give it a shot. DEFINE PROMPT is quite versatile, so try some various things until you get something that pleases you.

TI WRITER GRAPHICS

PRODUCING ART
WITH THE WORD PROCESSOR
PART ONE

BY ANNE DHEIN

How many times have you wished you could take a design you've drawn and add it to your text using the TI-Writer? You can, if you have a printer that is capable of producing dot graphics. How well you can do it depends on the combining of two factors that are not always well understood - the printer controls for your particular printer and the transliterate command in TI Writer.

The transliterate command has to do with the ASCII character codes listed on page 145 of your TI Writer manual. Any character codes can be changed, or "transliterated" to represent any other characters. This is a powerful feature of the TI Writer but it is almost ignored in the manual - perhaps because the various brands of printers interact differently with the transliterate command.

The information here comes from experimenting with the TI Epson printer. It should work without much change for Epson compatible printers such as the Gemini 10X. Although the codes may be somewhat different for other printers, the principle is the same.

Printer graphics consist of one or more columns of dots. For the TI printer there is a total of 480 such columns across a line. Each column is 8 positions high and a dot can appear in any one of the 8 positions. Each positibn has a number associated with it:

```
 _____
¦128¦
¦___¦
¦ 64¦
¦___¦
¦ 32¦
¦___¦
¦ 8 ¦
¦___¦
¦ 4 ¦
¦___¦
¦ 2 ¦
¦___¦
¦ 1 ¦
¦___¦
```

If the value of each dot were added together you would come up with the sum of 255. This is the highest number you will use and it would mean that every single position was occupied with a dot. Suppose dots 128, 16 and 2 were to be used. The sum would be 146. Any combination of dots you can think of will add up to a unique number between 1 and 255. In a column where no dots were used, a zero would be the value.

To start with, let's draw a single character that matches the text characters in size. A normal printer character is 6 columns wide including the right hand column, which is left vacant so that characters will not run together. Except for lower case decenders, the bottom positions are not used either. Designing standard size characters will allow you to use them quite freely within your text, even with such commands as adjust and center. The easiest way to design something is by using graph paper:

```
     ___________
128 ¦_¦_¦_¦_¦_¦_¦
 64 ¦_¦_¦*¦_¦_¦_¦
 32 ¦_¦_¦*¦_¦_¦_¦
 16 ¦*¦_¦*¦_¦*¦_¦
  8 ¦*¦_¦*¦_¦*¦_¦
  4 ¦_¦*¦*¦*¦_¦_¦
  2 ¦_¦_¦*¦_¦_¦_¦
  1 ¦_¦_¦_¦_¦_¦_¦
```

The sum of the first column is 24, and for the second, 4. The used positions in the third column all add up to 126, and the next two columns are 4 and 24. The last column is 0. To send the data the printer must be switched from text mode to graphics. The normal densit, density graphics mode is entered with the ASCII codes 27 and 75. The 75 must be followed by

two numbers which tell the printer how many columns of graphics to print on a line. Unless you are going to send more than 255 columns of data values (which is unlikely), the first number must be the EXACT number of columns you want to print and the second number zero; for our example, 6. The graphic data immediately follows the second number. Our string of numbers now looks like this: 27,75,6,0,24,4,126,4,24,0.

The transliterate code is now typed into the editor part of the TI Writer. We will take any keyboard symbol, such as the exclamation point which has an ASCII value of 33, and change it to represent our graphics. The transliteration code is a period followed by TL so the completed string looks like this:

> .TL 33:27,75,6,0,24,4,126,4,24,0 <

It should be on a line by itself and no carriage return should follow it. Once we have this code at the head of a document we can use the special character within the document any time by simply typing in an exclamation point. When the document is run through the formatter, the anchor will appear on the printed page wherever the exclamation point has been placed: ⚓ ⚓ ⚓

The number of small characters you can create and scatter freely throughout your document is almost unlimited! You can use just a few ASCII values you don't need in the text and use them over and over. Or, you can design a whole set of characters such as a special alphabet, each with its own unique value.

Now let's try something just a little more difficult. This next design extends 9 columns instead of 6. If the transliterate code contains data for more than 6 columns of graphics, the device name for your printer will need to have a .CR after it in order to suspend the carriage return function. Since .LF is the normal default on the printer, you will need to add line feeds to each line you want printed. This means all text, graphics and spaces, but not the transliterate codes. There are several ways to add line feed characters to your text. Probably the easiest is to run the document through the formatter, using DSK1.FILENAME as the print device. Or, using special character mode; type control U, shift J, control U. A transliterate code could also be used. You will also need to remove all carriage return symbols from your text; you can do this with the Replace String command.

Another reason why working with larger images seems more complicated is because when a graphic design extends to other lines the spacing is wrong for it. Standard spacing is 6 lines per inch, that is, 1/6 inch per line. But spacing can be set for as little as 1/72 of an inch to as much as 1 13/72 inches. The printer control codes for this are 27,65,n; where n is a number between 1 and 85. 1/6 is equavalent to 12/72 so standard line spacing would be represented by 27,65,12. The spacing we want for graphics is 8/72, or 27,65,8. I chose the = (ASCII 61) and > (ASCII 62) signs to transliterate: .TL 61:27,65,8 will give us the spacing we need for graphics and .TL 62:27,65,12 will change it back to standard spacing for text.

Does every, single transliterate code start with a pe.iod? And is there a space between the .TL and the ASCII value to be transliterated? This will be the only space in the string. Make sure there are no extra spaces and no skipped commas. Keep each .TL on a line to itself. And, contrary to what you may have heard, DO NOT put carriage returns behind any .TL codes that switch the printer to graphics mode. Don't use carriage returns at all when using .CR as part of your printer device name.

Do you have the right number of data values specified for each graphics code? For the code .TL 49:27, 75,N1,N2,1,2,3,4,5,6 the value of N1 should be 6 because there are 6 data units following. If N1 is any number up to 255 then N2 is a 0. If N1 is more than 255 it is represented by its actual number minus 255. For example, 258 -255 =3: N1 would be 3 and N2 would be 1.

Once you have transliterated your ASCII values properly, are you actually using them? .TL 33:10 changes the exclamation point to a line feed, but until you actually insert the ! into the document, nothing happens.

Do one or two of your characters show up as blanks? TI Writer reserves the use of the ampersand (shift 7), at sign (shift 2) and circumflex (shift 6) for its own purposes. It is best to stay away from these characters.

Are you printing your document through the formatter and are you using .CR at the end of your device name? If all else fails, check your values once more. Sometimes the data values 8, 12 and 13 will cause printer gliches. You may have to redesign your graphics slightly to get rid of the offending values.

Again, the suggestions in this article are just that - ideas for you to use in your own experimentation. Many printers also have double density graphics and some even go beyond that to very high resolution graphics. You may also want to consider using condensed, enlarged and enhanced print, and whatever other capabilities your printer may have. Letterheads, logos, monograms, emblems, maps, borders - there doesn't seem to be anything that the TI Writer can't do. Taking everything into consideration, there is still a lot to learn about using the transliterate codes - especially the ones concerning graphics. If you have some ideas of your own, or if you have gotten good results with another kind of printer, share it with us! Thanks go to Barb Berg of Trio+ Software for the use of her Flower Character Set (from TI Artist Companion) which is used at the head of this article.

Following is the transliteration file for our store logo, shown below. I designed it with TI Artist(tm) from Inscebot. You can use it as a model for your own graphic experiments and in a future newsletter I'll give those of you who have the TI Artist program a gigantic shortcut to TI Writer graphics. Until then, happy designing!

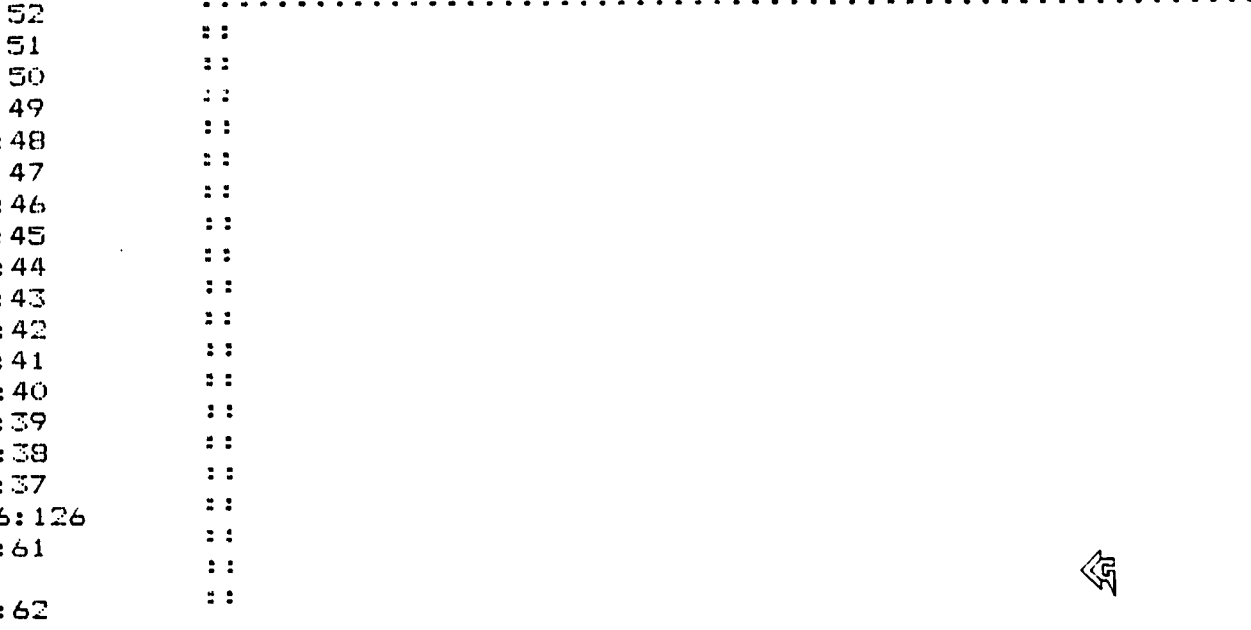
DWEBM'S
7 W AIRLINE HWY
WATERLOO IA 50703
True Value HARDWARE
[319] 236 3861

Please note: A special character was put onto the front of each line so that the transliterate codes would print through the formatter. Also; because the file is being printed through the formatter the circumflex does not show up on the printed copy. Each indented line (98765...etc.) is actually preceded by 12 circumflexes. This moves the graphics design over to the position it is wanted. The telephone number has 39 in front plus an enlarge print command. The margin command (.LM) doesn't seem to work on graphics. Also note that in order to use the ampersand (one of the TI reserved words) a double had to be used, which shows on the printed copy as a single ampersand. Line feed characters which are on the end of each line except those that hold transliterate codes do not show on the printed copy either.

```
.TL 126:10
.TL 61:27,65,8
.TL 62:27,65,12
=
.TL 57:27,75,8,0,0,0,0,0,0,31,31,31
.TL 56:27,75,8,0,31,31,28,28,30,31,15,7
.TL 55:27,75,8,0,0,31,31,31,31,31,31,1
.TL 54:27,75,8,0,31,31,31,31,0,31,31,31
.TL 53:27,75,8,0,31,31,31,29,29,29,29,29
.TL 52:27,75,8,0,0,31,31,31,31,31,31,0
.TL 51:27,75,8,0,31,31,31,31,31,31,15,7
.TL 50:27,75,8,0,31,31,31,0,0,48,57,59
.TL 49:27,75,8,0,63,63,0,0,7,15,31,31
.TL 48:27,75,8,0,31,31,29,29,29,28,28,0
.TL 47:27,75,8,0,0,0,0,0,0,0,0,0
.TL 46:27,75,8,0,0,0,0,0,0,0,0,0
.TL 45:27,75,8,0,0,0,0,0,0,0,0,0
.TL 44:27,75,8,0,0,0,0,0,0,0,0,0
.TL 43:27,75,8,0,0,0,0,0,0,0,0,0
.TL 42:27,75,8,0,0,0,0,0,0,0,0,0
.TL 41:27,75,8,0,0,0,0,0,0,0,0,0
.TL 40:27,75,8,0,0,0,0,0,0,0,0,0
.TL 39:27,75,8,0,0,0,0,0,0,0,0,0
.TL 38:27,75,8,0,0,0,0,0,0,0,0,0
.TL 37:27,75,8,0,0,0,0,0,0,0,0,0
                9876543210/.-,+*)('&%~
.TL 57:27,75,8,0,0,0,0,0,0,252,252,252
.TL 56:27,75,8,0,252,252,28,28,60,252,248,240
.TL 55:27,75,8,0,0,252,252,252,252,252,252,192
.TL 54:27,75,8,0,252,252,252,252,0,252,252,252
.TL 53:27,75,8,0,252,252,252,220,220,220,220,220
.TL 52:27,75,8,0,0,252,252,252,252,252,252,0
.TL 51:27,75,8,0,252,252,252,252,252,252,240,248
.TL 50:27,75,8,0,252,252,252,0,0,128,128,128
.TL 49:27,75,8,0,0,0,0,0,28,156,220,220
.TL 48:27,75,8,0,221,253,255,255,255,255,119,7
.TL 47:27,75,8,0,15,15,15,15,15,31,30,30
.TL 46:27,75,8,0,30,30,30,60,60,60,60,60
```

```
!   .TL 45:27,75,8,0,60,60,60,124,120,120,120,120
!   .TL 44:27,75,8,0,120,120,120,120,120,120,120,120
!   .TL 43:27,75,8,0,120,120,120,120,120,120,120,120
!   .TL 42:27,75,8,0,120,120,120,120,120,124,60,60
!   .TL 41:27,75,8,0,60,60,60,60,60,60,30,30
!   .TL 40:27,75,8,0,30,30,30,31,15,15,15,15
!   .TL 39:27,75,8,0,15,7,7,7,3,3,3,1
!   .TL 38:27,75,8,0,1,0,0,0,0,0,0,0
!   .TL 37:27,75,8,0,0,0,0,0,0,0,0,0
!                 9876543210/.-,+*)('&%~
!   .TL 57:27,75,8,0,0,0,0,0,0,67,76,112
!   .TL 56:27,75,8,0,0,0,0,0,0,127,6,127
!   .TL 55:27,75,8,0,0,0,0,0,0,63,72,63
!   .TL 54:27,75,8,0,0,17,31,17,0,31,20,11
!   .TL 53:27,75,8,0,0,31,1,1,0,17,31,17
!   .TL 52:27,75,8,0,0,31,14,31,0,31,21,17
!   .TL 51:27,75,8,0,0,0,0,0,0,127,8,127
!   .TL 50:27,75,8,0,0,31,6,31,0,28,7,28
!   .TL 49:27,75,8,0,3,7,31,63,127,127,252,248
!   .TL 48:27,75,8,0,240,240,224,224,192,192,128,128
!   .TL 47:27,75,8,0,128,128,0,0,0,0,0,96
!   .TL 46:27,75,8,0,64,64,71,120,65,71,68,68
!   .TL 45:27,75,8,0,7,0,1,6,0,0,1,3
!   .TL 44:27,75,8,0,4,0,1,34,68,67,64,64
!   .TL 43:27,75,8,0,96,62,1,0,0,3,28,96
!   .TL 42:27,75,8,0,3,4,4,3,24,38,65,34
!   .TL 41:27,75,8,0,28,0,0,1,6,0,0,1
!   .TL 40:27,75,8,0,7,0,0,0,3,4,3,128
!   .TL 39:27,75,8,0,128,128,128,192,192,224,224,240
!   .TL 38:27,75,8,0,240,248,252,127,127,63,31,15
!   .TL 37:27,75,8,0,7,0,0,0,0,0,0,0
!                 9876543210/.-,+*)('&%
!   .TL 49.49
!   .TL 50.50
!   .TL 51.51
!   .TL 54.54
!   .TL 56.56
!   .TL 57.57
!                                          [319] 236 3861
!   .TL 57:27,75,8,0,0,0,0,31,1,31,0,3
!   .TL 56:27,75,8,0,5,3,0,4,7,4,0,7
!   .TL 55:27,75,8,0,5,4,0,7,5,2,0,7
!   .TL 54:27,75,8,0,0,0,0,7,4,7,0,7
!   .TL 53:27,75,8,0,4,7,0,0,0,16,31,16
!   .TL 52:27,75,8,0,0,15,18,15,0,0,0,28
!   .TL 51:27,75,8,0,20,19,0,15,16,15,0,16
!   .TL 50:27,75,8,0,19,28,0,15,16,15,0,18
!   .TL 49:27,75,8,0,210,237,240,248,252,252,126,62
!   .TL 48:27,75,8,0,31,31,15,15,7,7,3,3
!   .TL 47:27,75,8,0,3,3,1,1,1,1,48,16
!   .TL 46:27,75,8,0,16,224,0,32,192,0,0,192
!   .TL 45:27,75,8,0,32,96,128,64,47,194,15,192
!   .TL 44:27,75,8,0,39,42,71,192,175,42,69,128
```

```
!   .TL 43:27,75,8,0,15,8,231,32,79,131,15,192
!   .TL 42:27,75,8,0,39,42,199,32,47,202,133,64
!   .TL 41:27,75,8,0,47,42,104,160,64,32,224,0
!   .TL 40:27,75,8,0,192,32,32,33,193,161,33,67
!   .TL 39:27,75,8,0,131,3,3,7,7,15,15,31
!   .TL 38:27,75,8,0,31,62,126,252,252,248,240,224
!   .TL 37:27,75,8,0,192,0,0,0,0,0,0,0
!                 9876543210/.-,+*)('&%~
!   .TL 57:27,75,8,0,0,0,0,192,128,192,0,192
!   .TL 56:27,75,8,0,0,192,0,0,192,0,0,192
!   .TL 55:27,75,8,0,64,64,0,192,0,192,0,192
!   .TL 54:27,75,8,0,64,64,0,192,64,192,0,192
!   .TL 53:27,75,8,0,64,192,0,0,0,64,192,64
!   .TL 52:27,75,8,0,0,192,0,192,0,0,0,64
!   .TL 51:27,75,8,0,64,128,0,128,64,128,0,192
!   .TL 50:27,75,8,0,0,0,0,128,64,128,0,64
!   .TL 49:27,75,8,0,64,128,0,0,0,0,0,0
!   .TL 48:27,75,8,0,0,0,128,128,128,192,192,192
!   .TL 47:27,75,8,0,224,224,224,224,224,240,240,240
!   .TL 46:27,75,8,0,240,240,240,120,120,120,120,120
!   .TL 45:27,75,8,0,120,120,120,124,188,60,188,60
!   .TL 44:27,75,8,0,188,60,188,60,188,60,188,60
!   .TL 43:27,75,8,0,188,188,60,60,188,60,188,60
!   .TL 42:27,75,8,0,188,60,188,60,188,60,184,56
!   .TL 41:27,75,8,0,184,248,248,120,120,120,240,240
!   .TL 40:27,75,8,0,240,240,240,240,224,224,224,224
!   .TL 39:27,75,8,0,224,192,192,192,128,128,128,0
!   .TL 38:27,75,8,0,0,0,0,0,0,0,0,0
!   .TL 37:27,75,8,0,0,0,0,0,0,0,0,0
!                 9876543210/.-,+*)('&%~
!   .TL 57:57
!   .TL 56:56
!   .TL 55:55
!   .TL 54:54
!   .TL 53:53
!   .TL 52:52
!   .TL 51:51
!   .TL 50:50
!   .TL 49:49
!   .TL 48:48
!   .TL 47:47
!   .TL 46:46
!   .TL 45:45
!   .TL 44:44
!   .TL 43:43
!   .TL 42:42
!   .TL 41:41
!   .TL 40:40
!   .TL 39:39
!   .TL 38:38
!   .TL 37:37
!   .TL 126:126
!   .TL 61:61
!   >
!   .TL 62:62
```



Now, using the TI Artist drawing package by Chris Faherty and the conversion program listed here, you can easily add your own graphics to any TI Writer fil

TI WRITER GRAPHICS

PRODUCING ART WITH THE WORD PROCESSOR PART TWO

BY ANNE DHEIN

Part One of this article showed you how to use transliterates to produce graphics with the TI Writer. Theoretically the idea works fine, but when you actually start designing a picture on graph paper you quickly find yourself bogged down by tremendous amounts of graphic data - the transliterates themselves and then a number for each and every dot you wish to have printed on the paper. And to top it off, the numbers have to be in an exact order. At best, if you make a mistake your picture will look funny; at worst you can crash the system by having innocent-looking numbers in wrong places.

So, the idea of TI Writer graphics, while technically feasible becomes impossibly hard as you get into larger areas of graphics. Do you recall the True Value Logo in Part One? It took 146 lines of transliterates, each with about a dozen numbers, to make a graphic reproduction on the page that was 1/2 inch high by 5 1/2 inches long (actually only 2 3/4 inches long if you don't count the phone number which is done with enlarged print, not graphics).

The True Value Logo was easier to produce than you might think, thou Instead of drawing it on graph paper and laborously figuring out the many data values as I could have done, I used TI Artist to design the picture and an Extended Basic program to convert the picture to a transliterate file that can be used to dump the graphics to a printer from the TI Writer formatter.

TI Artist is a generalized drawing program using bit map graphics to produce a high quality picture with very little training. For its small price ($19.95 at Dhein's), it has many excellent features, but the one that is of interest here is the program's ability to save a picture on disk as a DIS/VAR 80 file. The graphic data for this picture, which is called an INSTANCE, can be looked at with the TI Writer. The Instance files were designed in such a way that they would be easy for someone to use in his own Extended Basic programs. Instead of using them in programs, we want to use them as the basis of a TI Writer transliteration file that will reproduce the same picture from the TI Writer formatter.

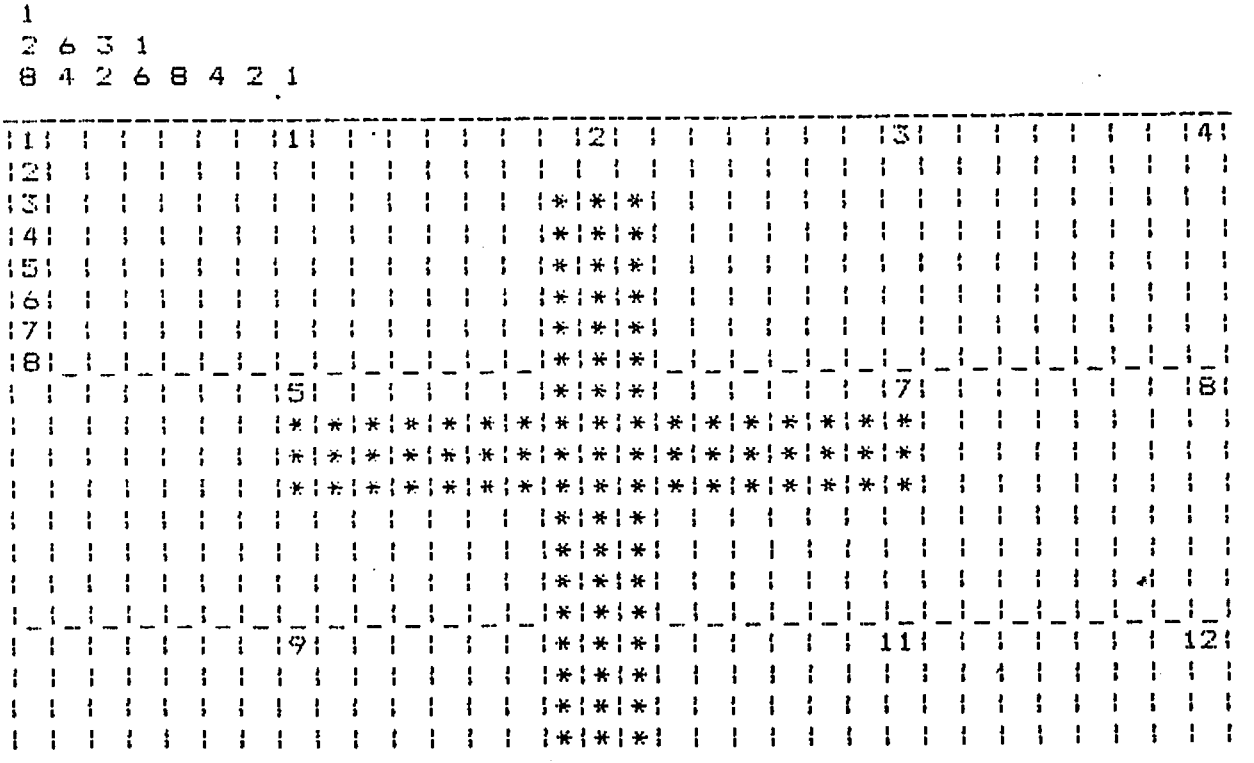
When you draw a picture using TI Artist, imagine that the screen is composed of a grid of blocks - 32 blocks wide and 24 blocks high; 768 blocks in all. Each block is composed of 64 dots, each of which is capable of being turned on or off independently. (768 X 64 is 49,152 dots over which you have control. That's a lot of dots!) When the dots are on, you can see them on the screen. When the screen is dumped to a printer, each dot that is "on" on the screen is reproduced by the printer.

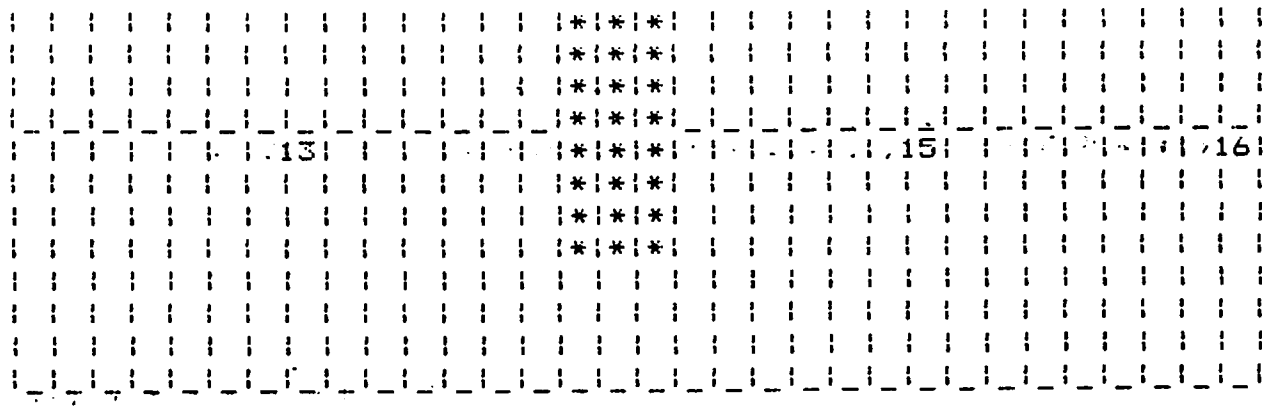
For now we want to save all or part of the picture onto a disk usin the Instance option. The area of the picture you designate for saving is

saved on character boundaries - that is, along the lines of some of those 7x8 blocks mentioned earlier. Each dot in each block that has been selected to be saved is assigned a number. The test drawing I made consisted of two lines - one horizontal, one vertical - which formed a cross. Here is the DIS/VAR 80 file I got when I saved the drawing as an Instance:

```
4,4
0,0,0,0,0,0,0,0
0,0,3,3,3,3,3,3
0,0,128,128,128,128,128,128
0,0,0,0,0,0,0,0
0,1,1,1,0,0,0,0
3,255,255,255,3,3,3,3
128,255,255,255,128,128,128,128
0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0
3,3,3,3,3,3,3,3
128,128,128,128,128,128,128,128
0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0
3,3,3,3,0,0,0,0
128,128,128,128,0,0,0,0
0,0,0,0,0,0,0,0
```

The Instance files are set up so that the top line (record 0) consists of two numbers - The number of blocks (or character positions) that the picture takes up across the screen, and the number of blocks high it is. The remaining 16 lines in this file consist of one record for each block, with the record containing the information as to which dots are turned on in that particular block. Here is a graphic illustration of the blocks with the appropriate dots "on":





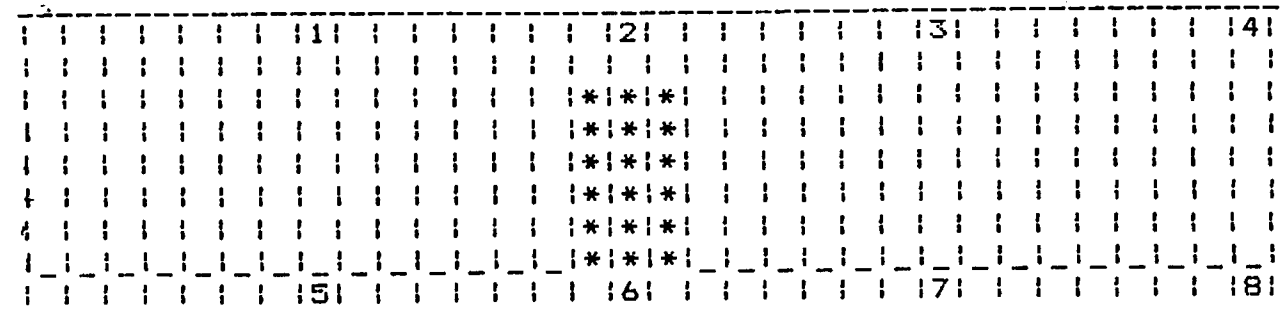
The test drawing takes up four blocks across the screen and is also 4 blocks high. Notice that each block has been divided into 64 parts to represent the 64 dots that can be turned on or off individually on the screen. The blocks have been numbered in the upper right corner so that you can see them easier. In the file, block width comes first so the first four lines correspond to blocks 1-4; the next four lines are for blocks 5-8; and so on, down to the last line of the file which is block 16.

Now think of a block as eight rows of eight dots each. Each dot in the row has been assigned a number as shown:

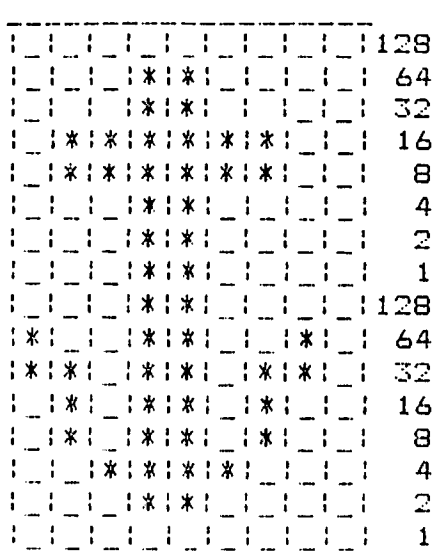
*	*	*	*	*	*	*	*
***	***	***	***	***	***	***	***
*	*	*	*	*	*	*	*
128	64	32	16	8	4	2	1

If the values of each position in the column were added together you would come up with the sum of 255. This is the highest number you will see in an Instance file. It means that every single dot in that particular block has been turned on. Suppose dots 128, 16 and 2 were on. THe sum would be 146. Any combination of dots you can think of will add up to a unique number between 1 and 255. If no dots are turned on you would have a zero value for that row.

Now match the first record of your file to block number one. There are eight zeros - one zero for each row. Since they are all zeros, no dots have been turned on in this block. Going on to the next record, we see that the first two numbers here are zero, and the remainder are threes. Rows one and two in block two have no dots. The third number, three, has been obtained by adding dots one and two together. Dots one and two are on in the remaining rows of block two. Block three has zeros for the first two rows again, and the remaining numbers are all 128. 128 is the exact number of the left-most dot; this dot is turned on. Block four is again all zeros, so our first four blocks look like this:



Now we'll design the character:



This time two transliteration codes are needed - one for each line of columns. From the left, the top 9 columns will have these data values: 0,24,24,127,127,24,24,0,0. If a "1" is used for the first character, the transliteration code will look like this: .TL 49:27,75,9,0,0,24,24,127,127,24,24,0,0. The ASCII value for one is 49; 27 and 75 are the codes needed to switch to graphics mode; the 9 and 0 are the units which tell the printer how many columns of graphics will follow. For the second transliteration we'll use "2", which has an ASCII value of 50; .TL 50:27,75,9,0,96,56,4,254,254,4,56,96,0.

Now, using the TI Writer editor, prepare a transliteration file with the codes. Save it under the file name TEST. The file should contain these lines:

```
.TL 61:27,65,8
.TL 62:27,65,12
.TL 33:10
.TL 49:27,75,9,0,0,24,24,127,127,24,24,0,0
.TL 50:27,75,9,0,96,56,4,254,254,4,56,96,0
=
1 ANCHORS!
2 AWAY!
>12!
.TL 61:61
.TL 62:62
.TL 49:49
.TL 50:50
.TL 33:33
1 2 !
```

Print the file through the Formatter using the device name you normally use, except delete the .LF and add CR. If it doesn't work, you may need to experiment to find what's right for you. Notice that the transliteration codes do not appear on the printed page at all, nor have the lines they were occupying been saved. The "1" shows up as the top part of the anchor and the "2" represents the bottom part. The equal sign narrowed the line space (look how close the two words are) but the greater than sign restored standard spacing.

As you make bigger and fancier designs, you will find that sometimes the transliterate commands just don't seem to work the way you think they should. Then you'll have to spend some time debugging. Here are some things to keep in mind:

Transliterate codes are limited to one 80 column line but you may use as many transliterates as you like for one line of graphics; you can cram as many data units as will fit onto the line, or as few as you please. Since we are using a file as our base that has eight graphics units per record (one character), we will use this same number for each transliterate code.

The program listing following this article was written by David Dhein. It was enhanced by Paul Berg of Trio+ Software so that graphics could be centered, and so it could be used with a ProWriter as well as the Epson printer. Type it into Extended Basic and save the file on disk with the name CONVERT. Or, if you'd rather not type in the program, it is available through the club library.

To use, you must first prepare your picture with the TI Artist Program, and save it as an Instance. Load CONVERT into Extended Basic and RUN. When asked for the Instance file name, type in the name of the Instance, but omit the "_I" from the end of it. This is the name that the transliterate file will be called. You will next be asked for the type of printer you have. The program will work with Epson and Gemini 10X and compatible printers, and with the ProWriter and compatibles. If you wish, the program can center your graphics on the page; otherwise, the design will start at the left margin.

Your transliterate file is now ready to be prepared. Depending on how large it is, you may have to wait quite awhile for it to be converted. When it is complete, you will have a file that you can run through the Formatter of the TI Writer. It will produce the same graphics that you prepared in TI Artist, although perhaps smaller than you had expected - one screen-width will cover about one half of a page width. When you use the formatter, use the same file name that you typed into the conversion program. When asked for the printer device name, use .CR on the end instead of .LF which is the normal default. This works on our TI Impact printer, on the Gemini 10 and 15X's and on ProWriters run on a serial interface, but if you are using a parallel interface (PIO) you may have to experiment some.

If you are a long-time user of the TI Writer, you may have already realized that when .CR is used as part of the formatter's printer parameters, a text file prepared in the editor would no longer print properly. With the carriage return suppressed, each suceeding line is typed right over the top of the first. Since one of the beauties of CONVERT was to be the fact that graphics could be printed right along with text without having to run a page through the printer twice, we had to think of a way around this problem. The solution was to add a line feed character to each text line. This does have limitations; graphics line spacing and normal line spacing do not easily mix in long files, and no good way has yet been found to use a line space of 8 lines per inch in a longer file.

If your text is not too long you can add line feed characters right from the editor. Press control U which will give you a flashing underline for the cursor. Now type a J after each line of text (make sure the J is upper case). Press control U again to get the regular cursor back. Don't add line feed characters to the transliterate file - it is complete as it is.

If you have a long file which is to be printed 6 lines to the inch, prepare you text file on the editor as usual. You can experiment with format commands - some work perfectly with the .CR parameter, others are tricky. When the document is ready (minus the drawing), run it through the formatter, only instead of printing it on paper, print it on a disk. Use, for example, DSK1.MYFILE as the device name. Be sure to use a different file name from the one you already have if don't want to lose the original file. The new file will print fine through the formatter when .CR is used as part of the printer device name.

Now you can insert your drawing into your text document at any point, using the merge technique described on page 73 of the TI Writer manual. If your files are too long to be merged into one file, you can break the text file into two or more parts (be sure to a different file name for each section). Use the Include File command (page 109 in the TI Writer manual) to print all the parts, one after the other.

The work that members of our group have done with graphics through the TI Writer has just scratched the surface of what could be done. We worked with Epson and ProWriter but there are other printers out there too. Using the Special Character mode you could undoubtedly produce graphics from the editor as well...it has been done on a limited basis. Since so little is written about the transliterate command or the Special Character mode in the TI Writer manual, one can only wonder what TI would have done with them had they stayed on the 99/4A computer business. If anyone out there can add to the program - for instance, making it work with a different type of printer - or can shed more light on the mysteries of TI Writer graphics, our club would like to hear from you. If you're interested in TI Writer graphics too, let us know and we'll share new discoveries.

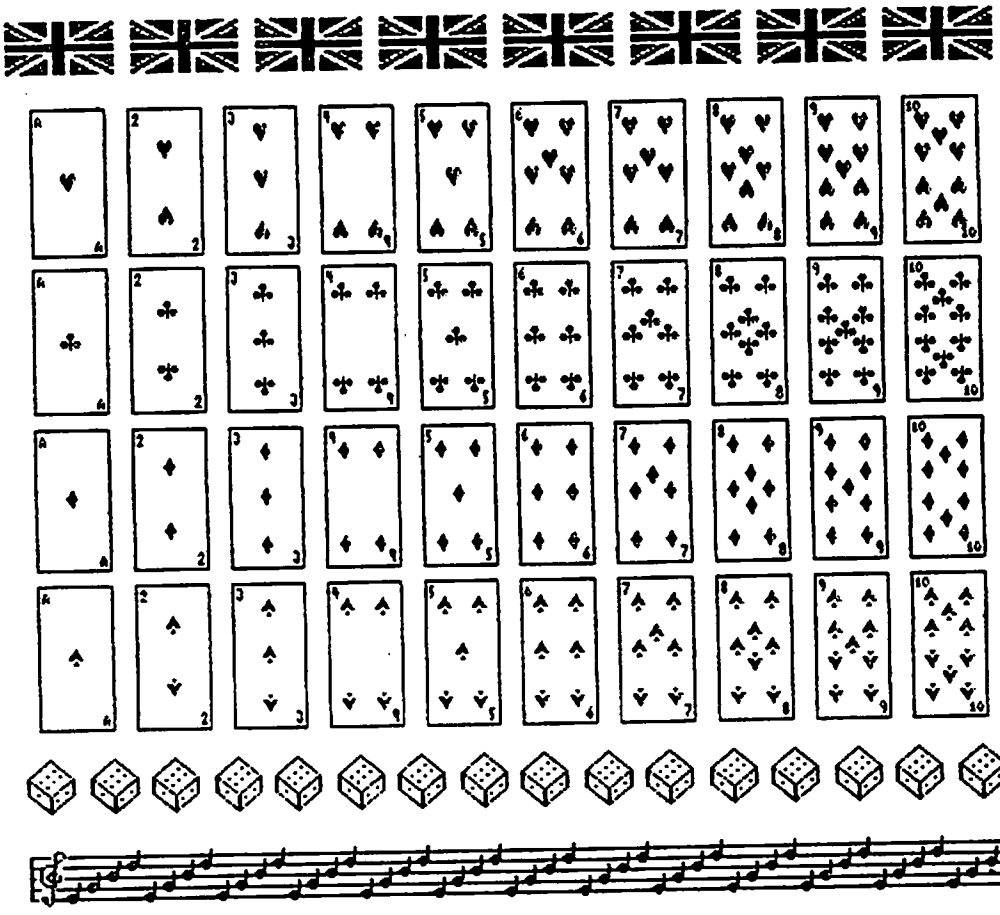
Trio+ Software will be coming out with a vastly expanded version of this program which will allow you to place graphics anywhere on the paper you wish, and possibly even to merge files. It will also contain a disk cataloging feature that will help you keep track of your transliterate files. This new program will let you design single character fonts (or use existing ones) which can then be converted to transliterate files. This means that you can take any text file and have it print out in a different font style. I saw results using a script font from Trio+ and it was beautiful! It looked like someone's small handwriting.

Thanks go to Barb Berg of Trio+ Software for the use of the Flower character font used at the beginning of the article. It comes from the TI Artist Companion package put out by Trio+. They also have a second companion package which contains the Script font mentioned above. Both sets are excellent, and you should have them if you use TI Artist very often. For more information contact Trio+ Software at P.O.Box 115, Liscomb, IA 50148; or talk to Paul Berg at any meeting.

```
100 DISPLAY ERASE ALL AT(1,4):"TI-ARTIST TO TI-WRITER": :"     CONVERSION PROGRA
M"
110 DISPLAY AT(5,1):"INSTANCE file name:"
120 ACCEPT AT(5,21)SIZE(8):NAME$
130 DISPLAY AT(7,3):"The file is on drive 1"
140 ACCEPT AT(7,24)SIZE(-1)VALIDATE(DIGIT):FD
150 DISPLAY AT(8,1):"Which drive for new file? 1"
160 ACCEPT AT(8,27)SIZE(-1)VALIDATE(DIGIT):SD
170 DISPLAY AT(10,1):"Select printer: 1": :"  1 Epson": :"  2 Prowriter"
180 ACCEPT AT(10,17)SIZE(-1)VALIDATE(DIGIT):P
190 A$="DSK"&STR$(SD)&"."&NAME$
200 NAME$="DSK"&STR$(FD)&"."&NAME$&"_I"
210 DISPLAY AT(18,8):"...Working."
220 OPEN #1:NAME$,INPUT
230 OPEN #2:A$,OUTPUT
240 INPUT #1:X,Y
241 DISPLAY ERASE ALL AT(1,1):"OUTPUT CENTERED ? Y" ::ACCEPT VALIDATE("YN")SIZE
(-1)AT(1,19):C$
250 IF X*Y>25 THEN DISPLAY AT(20,4):"This may take awhile.":::DISPLAY AT(21,4):
"Please be patient..."
260 PRINT #2:".TL 92:10" ! \=CHR$(10)=LINE FEED
270 IF P=1 THEN PRINT #2:".TL 61:27,65,8" :: PRINT #2:".TL 62:27,65,12" ::PRINT
 #2:"=" :: GOTO 290 ! EPSON COMMANDS
275 ! = IS 8/72  LINE SPACE        > IS 12/72 LINE SPACE
280 PRINT #2:".TL 62:27,65" :: PRINT #2:CHR$(27)&"T16":: PRINT #2:"\" ! PROWRI
TER COMMANDS
285 ! > IS 6 LINES TO INCH         2nd LINE IS CUSTOM LINE SPACE AT 16/144
290 FOR K=1 TO Y
300 FOR L=1 TO X
310 IF P=1 THEN INPUT #1:C(7),C(6),C(5),C(4),C(3),C(2),C(1),C(0):: GOTO 330
320 INPUT #1:C(0),C(1),C(2),C(3),C(4),C(5),C(6),C(7)
330 FOR I=7 TO 0 STEP -1
340 A=C(I)
350 FOR J=7 TO 0 STEP -1
360 IF 2 J>A THEN 390
370 A=A-2^J
380 B(J)=B(J)+2^I
390 NEXT J
400 NEXT I
410 A$=STR$(B(0))
420 B(0)=0
430 FOR I=1 TO 7
440 A$=STR$(B(I))&","&A$
450 B(I)=0
460 NEXT I
470 IF P=1 THEN PRINT #2:".TL "&SEG$(STR$(127-L),1,3)&":27,75,8,0,"&A$ ::GOTO
90
480 PRINT #2:".TL "&SEG$(STR$(127-L),1,3)&":27,83,48,48,48,56,"&A$
490 NEXT L
491 IF C$="Y" THEN PRINT #2:".CE"
500 N$="" :: FOR N=1 TO X :: N$=N$&CHR$(127-N):: NEXT N :: PRINT #2:N$&"\"
510 NEXT K
520 FOR N=1 TO X ::N$=".TL ":: N1$=SEG$(STR$(127-N),1,3):: N$=N$&N1$&":"&SN1
:: PRINT #2:N$ :: NEXT N
7
```

```
530 PRINT #2:".TL 92.92"
540 IF P=1 THEN PRINT #2:".TL 61.61"
550 PRINT #2:">"
560 PRINT #2:".TL 62.62"
570 CLOSE #1
580 CLOSE #2
590 END
```

TI WRITER GRAPHICS



BY JAMES STRINGFELLOW

THESE GRAPHICS WERE PRINTED DIRECTLY FROM THE KEYBOARD
USING TI WRITER AND EPSON MX80 PRINTER

Here is a program which should be useful for people using graph paper to design graphics for TI-WRITER.

It is for use with the EPSON 80 printer and should be easy to convert. I have included at the end of the program some changes to be made if your printer head has the top wire numbered 64 instead of 1 as with the EPSON 80.

The screen is 14 dots high and 16 dot rows long which should make quite large characters.

You can inverse your graph by pressing "M" for mirror.

After saving to a disk you can load into your TI-WRITER for printing with your letter.

TI WRITER GRAPHICS

Graphics can be included in your letters printed by TI Writer, directly from the keyboard. This can be useful to personalize your letterheads. The printer manual explains how to do graphics.
A Basic program to print this ■ would be as follows:

```
10 OPEN #1:"RS232.BA=4800"20 PRINT #1:CHR$(27);CHR$(75);CHR$(6);CHR$(0)
;CHR$(63);CHR$(63);CHR$(63);(63);CHR$(63);CHR$(63);CHR$(63) :: CLOSE #1
```

A dot matrix printer prints 7 dots vertically each time it receives a ASCII code from your program. See ASCII character code chart.

With TI Writer, it is only for numbers 0 to 31 that you will use CTRL U. I have enclosed a chart giving the equivalent numbers for each character that you will see on the screen; also a grid that will help you design your graphics.

To print this ■ you would see on the screen

'bK66??????

broken down as follows for a better understanding:

'bK66
↑ This is CHR$(27) followed by K to turn on graphic mode.

'bK66
↑ This position indicates the amount of graphic lines following, up to 127, in this case "six".

'bK66
↑ This position can be from 0 to 4, each unit equals 256 vertical dot lines i.e 1=256, 2=512, 3=768, 4=1024. In this case "zero".

?????? are the number of vertical dots to turn on, six times 63.

Resume:

- The 'bK turns on graphic mode.
- The small 6 indicates there will be 6 data following.
- The small 6 in the position of bigger graphic data.

```
Dot 7 = 64
Dot 6 = 32        Examples:
Dot 5 = 16
Dot 3 =  8        Dots 64+1      = 65 or "A" would print :
Dot 3 =  4        Dots 64+8+1    = 73 or "I" would print :
Dot 1 =  2        Dots 1+2+8+16  = 27 or "b" would print :
Dot 1 =  1
       ---
Total  127 .... giving you this graphic  |
       ===
```

In order to print a data, add the value of the dot positions together.

127 can be used by FCTN & V but you will see nothing on the screen.

To draw a straight line like this

type in the following lines using CTRL U to see this on the screen

```
'bK:1
1111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
1111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
1111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
```

'bK This turns on graphic mode.
:1
These two numbers give the amount of graphic dot lines to print.
The first number can be from 1 to 127.
'bK~:
↑ The ~ here indicates 126 vertical dot lines.

'bK:1
↑ The 1 here indicates 256 vertical dot lines.
This number can be multiplied by 256, i.e. 1 in this position = 256.
When using more than 80 graphic lines you must add a carriage return
to your printer, i.e. "RS232BA=4800.CR".

To print a ♡ this is what you would see on the screen.

```
'bL:0HD"1"DHO
```

To print a ♠ this is what you would see on the screen.

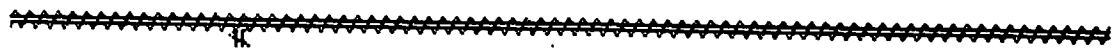
```
'bL:0A'C
      89s98tA
```

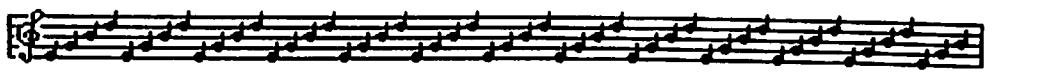
The letter K turns on double graphic printing.
The small : is the amount of graphics to print.
The 0 is not used because the amount of lines is under 128.
The nine following codes are the vertical graphic dot lines.

I hope that these examples will enable you to make your own graphics
with TI Writer.
I will be happy to answer any questions that you may have.

JAMES STRINGFELLOW

James Stringfellow
23 rue Pasteur
78700 CONFLANS
FRANCE





PROWRITER GRAPHICS

TI-WRITER GRAPHICS WITH THE PROWRITER

By Barb Berg

Those of us with the Prowriter know this printer can do a lot of neat things, including graphics, but don't know how to use these capabilities with such software as TI-WRITER. Bit-dot graphics, especially, seem elusive even to the advanced programmer. This article will show how the Prowriter does bit-dot graphics in TI-WRITER.

The graphics are produced by using the transliterate command, which lets you make any printable ASCII character into another character. You may have used this method to print the Prowriter graphics characters in the range of 128-247. But one thing that always seemed to elude me was how to create my own bit-dot graphics characters. Finally, through much trial and error, I stumbled onto the codes needed to do this and I'll try to explain how it is done.

If you have printed the graphics characters using transliterates, you already know how to set up a regular transliterate code. For example, if you want to print the Prowriter's graphic character of a solid heart shape, character 233, you would use a printable character and transliterate it to that character. If you use the tilde, or character 126, as the character to transliterate, your command would look like this:

```
! .TL 126:233
```

But if you want to design your own graphics character or even redefine the character set, you need a series of codes to define the new character(s). First, let's see how a character is printed on this dot-matrix printer.

Each character on the Prowriter consists of a series of columns with certain dots printed in each column. We can have one of two formats on the Pro: 7Hx9V or 8Hx8V, where the H stands for horizontal (rows) and the V for vertical (columns). In order to do bit-dot graphics, we need the 8x8 format. There is a DIP switch inside the front of your printer that determines which format the printer will operate in, designated as SW2-6. If the switch is set open, you have the 8x8 format, if closed, you have the 7x9 format, which is the setting the printer came with. The switch is in one of the two sets of switches located inside the front of the printer. Chances are, you will have reset one of these DIP switches at some time since you bought your printer and will know where to locate them. If not, check your manual.

Now, with the printer set to receive 8-bit data, each letter will be printed in an 8x8 matrix, similar to the way the TI prints its screen characters. The difference is that the printer will print the character starting with the left-most column and print columns, instead of the screen method of top row to bottom row. Now let's see how each column is set up.

Each column consists of eight dots which are printed by using corresponding pins on the print head. These pins are addressed through a number code, using the numbers 1-255. Each number addresses a different pin or combination of pins, depending on the number.

Each pin has its own identifying number, as shown. Beginning at the top, the pins are numbered 1,2,4,8,16,32,64 and 128. If a code of 1 is sent, the top pin hits the paper and a dot is left in that position, a 2 prints in the second position down, a 4 prints in the third position, etc. To print in the top two positions, you simply add the numbers of the two pins together. A 3, then, would cause the top two pins to strike the paper. You should be able to see how each combination has its own code number.

Programmers who design screen graphics usually have some 8x8 grid graph paper on hand, and this works great for printer graphics as well. Let's start by making a character that will be useful as a border.

```
1   00000000   To produce this character, we need to add up the pin numbers for each column from left to right. Column one uses
2   00000000   the pins numbered 8 and 64, so the code for column one is 8+64=72. The code for column two is 36, column three is
4   00000000   18, column four is 9, column five is 13, column six is 30, column seven is 60 and column eight is 88. Once we
8   00000000   have these numbers, we need to set up the transliterate code for our new character.
16  00000000    In order to use these eight bit-dot numbers as one character, we need to tell the printer that we are using
32  00000000   bit-dot mode. In a program, we would address it this way:
64  00000000
128 00000000   PRINT #1:CHR$(27);"S0008";CHR$(72);CHR$(36);CHR$(18);CHR$(9); CHR$(13);CHR$(30);CHR$(60);CHR$(88)
```

Getting this to work in TI-WRITER transliterate form took some time to figure out, so be sure you understand this part! In the command above, the "S0008" is interpreted this way: the "S" command signifies bit-dot mode. The "0008" is the four digit number which tells the printer how many characters are to be interpreted as bit-dot graphics. Since we are sending eight codes, we need to send the printer the number 8 in the format it expects. The rest of the command consists of the eight bit-dot character codes.

To set this up in TI-WRITER, as each character is transliterated we need to include this "S0008" code. The ONLY way TI-WRITER will accept this code is in decimal equivalent ASCII numbered format. The "S0008" command is in decimal format. NOTE DIFFERENCE!

What we do, then, is figure out the decimal ASCII numbers for each character in the code. ESCape, or CHR$(27) stays the same, 27. "S" is ASCII character 83. The number "0" is ASCII 48, and "8" is ASCII 56. So the numbers we need to set up a bit-dot character with eight codes are these:

27,83,48,48,48,56
ESC S 0 0 0 8

See how it works? To put our character into transliterate form, type in the following line (without the exclamation point):

! .TL 126:27,83,48,48,48,56,72,36,18,9,13,30,60,88

Now print your border by typing a line of tildes, like this:
~~~~~~

One more thing: when using bit-dot graphics, you will need to put a line feed at the end of each printed line, whether it has bit-dot graphics or text. This can be accomplished by transliterating a little-used character to the character code for a line feed, CHR$(10):

! .TL 124:10

To print the file through the formatter, change the printer device name from PIO.LF to PIO.CR. This will suppress the carriage return instead of the line feed.

Suppose you want to make a design that would use two lines of print and is two characters high. Since the default line spacing would leave a nice little gap in the middle of the design, you will have to set up the printer for custom line feed. This is done simply by sending a code in this format:

ESC T N1 N0

"T" sets up for custom line feed. N1 and N0 represent a two digit number, N, which must be between 01 and 99. N represents the line feed pitch of N/144 inch. The best graphics result when N is set to 16 by using the code ESC T16. In a program, it's done like this:

PRINT #1:CHR$(27);"T16"

You can use a transliterate code to set up custom line feed in TI-WRITER or you can simply send a special character control code sequence, using the CTRL U command shown on page 146 of the TI-WRITER manual.

If you use the first method, anytime you type character 91, or open bracket, your document will begin printing in custom line feed for graphics. You would also want to define another character, such as the close bracket, as normal line feed and type it in the document after your graphics end.

CUSTOM LINE FEED	DEFAULT LINE FEED (6 per inch)
! .TL 91:27,84,49,54	.TL 93:27,65
OR	
CTRL U / FCTN R / CTRL U / T16	CTRL U / FCTN R / CTRL U / A

The second method is the one I use. This will show on the screen as \T16 and \A. Use the method that suits you best; I've done it this way long enough that it has become second nature for me, but you may like the other method better.

Now let's define a two-character high design. For the heck of it, we'll do a little bottle and transliterate it to the two brace characters, 123 and 125:

```
  1   00000000
  2   00000000
  4   00000000
  8   00000000  Character 1:
 16   00000000  .TL 123:27,83,48,48,48,56,123,64,62,1,1,62,64,123
 32   00000000
 64   00000000
128   00000000
  1   ********
  2   *000000*
  4   *000000*
  8   *000000*  Character 2:
 16   *000000*  .TL 125:27,83,48,48,48,56,127,161,161,161,161,161,161,127
 32   ********
 64   *000000*
128   *******0
```
~~~~~~

Now set up a file with text like this: (omit the exclamations)

```
! .TL 124:10
! .TL 123:27,83,48,48,48,56,128,64,62,1,1,62,64,128
! .TL 125:27,83,48,48,48,56,127,161,161,161,161,161,161,127
! \T16
! ( BOTTOMS UP!!
! )!
! \A
! .TL 123:123
! .TL 125:125
```

Try it! This is what it will produce:

```
( BOTTOMS UP!
```

Please note that the exclamation points at the beginning of the transliterate code lines were only put there so they would print in this document and not be used as transliterates. When you put them in your document, the .TL should not have any other characters on the line before it.

You may want to try to keep your characters no more than 8 columns wide, when possible. If a character is more than 8 columns wide, the printer still thinks that character is within that 8 column limit and has a tendency to lose track of the number of columns in a line. I recently redefined the entire character set to get a script font, and some of the characters had to be over 8 columns wide in order to look right. When the file was printed, if the line came close to the margin limits, instead of the character I got the code numbers that were left in the character after getting to the last printable column. Using the right margin command .RM within the document didn't help regain the lost part of the transliterated characters.

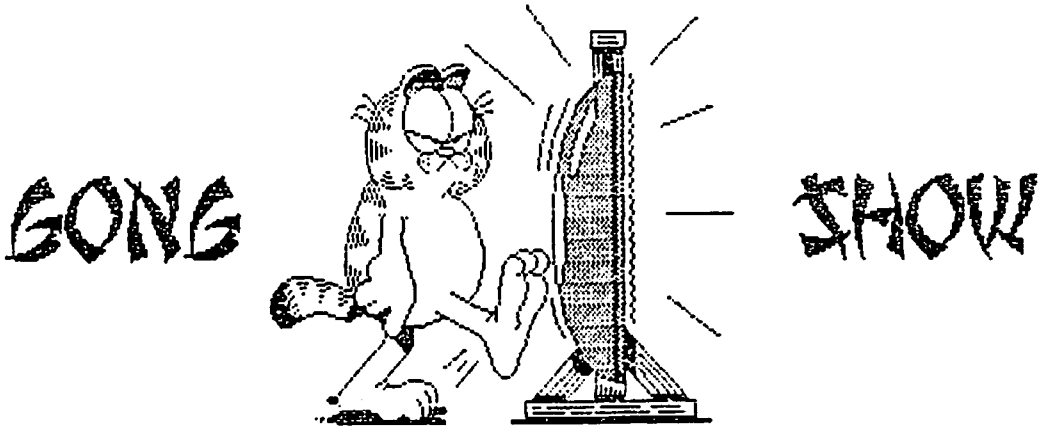
To overcome this, when I create the document I use the Tab command and set my right margin to about 10 characters less than what I want. After the document is complete, I then reset my right margin to the desired position. If you find you have to do this, REMEMBER: DON'T REFORMAT THE TEXT WITH THE NEW MARGIN SETTING IN WORD WRAP MODE! You have to reset the right margin to your first setting to preserve your file if you have any reformatting to do. You may wish to type the document in fixed mode instead, since reformatting only affects the line where the cursor is positioned. This also gives you the opportunity to type your line feed character at the end of each line as you go.

If you do redefine a character that uses more than 8 columns, you need to change the numbers in the bit-dot command. For example, if the new character will be 10 columns wide, your transliterate code would be in this format:

```
27,83,48,48,49,48,n1,n2,n3,n4,n5,n6,n7,n8,n9,n10
ESC S  0  0  1  0 ....
```

You can make your graphics as many characters wide as you can get to fit on a line and still print correctly. The maximum characters per line that I have been able to use is 72, but I can also use required spaces in the line to get the graphics positioned where I want them.

Triot Software is currently working on a program that will allow you to use TI-ARTIST instances and fonts in TI-WRITER by converting them to the latter's format. This makes it even easier to produce quality graphics in with your documents. Just to give you an idea, here is one of the graphics I produced recently for a local celebration. The Garfield and gong graphic was done on TI-ARTIST and saved as an instance, as were the words GONG and SHOW. They were then converted to TI-WRITER format with an early version of the convert program. The resulting files were merged together in the desired places to produce the final file.



PRINTER COMMANDS

(energizes or turns on)

	10X	SG-10	MX-80	FX-80	KX-P1091	OKIDATA
ITALICS	27 52	27 52	##########	27 52	27 52	##########
ELITE	27 66 2	27 66 2	##########	27 77	27 77	28
CONDENSED	27 15	27 15	27 15	27 15	27 15	29
PICA	27 66 1	27 66 1	##########	##########	27 80	30
EXPANDED	27 87 1	27 87 1	27 14	27 87 1	27 87 1	31
SUPERSCRIPT	27 83 0	27 83 0	########## 27 83 0	27 83 0	27 74	
SUBSCRIPT	27 83 1	27 83 1	##########	27 83 1	27 83 1	27 76
NEAR LETTER	##########	27 66 4	##########	27 120 1	27 110	27 49
EMPHASIZED	27 69	27 69	##########	27 69	27 69	27 84
UNDERLINE	27 45 1	27 45 1	##########	27 45 1	27 45 1	27 67
DOUBLE STRIKE	27 71	27 71	27 71	27 71	27 71	27 72
SLASHED ZERO	##########	27 92 1	##########	##########	##########	##########
1/8 LINE SP.	27 48	27 48	27 48	27 48	27 48	27 56
1/6 LINE SP.	27 50	27 50	27 50	27 50	27 50	27 54
7/72 LINE SP.	27 49	27 49	27 49	27 49	27 49	##########
n/72 LINE SP.	27 65 n	27 65 n	27 65 n	27 65 n	##########	##########
n/144 LINE SP.	27 51 n	27 51 n	##########	##########	##########	27 37 57 n
n/216 LINE SP.	##########	##########	##########	27 51 n	##########	##########
TOP MARGIN	27 82 n	27 82 n	##########	##########	##########	##########
BOTTOM MARGIN	27 78 n	27 78 n	27 78 n	27 78 n	##########	##########
LEFT MARGIN	27 77 n	27 77 n	##########	27 108 n	##########	##########
RIGHT MARGIN	27 81 n	27 81 n	##########	27 81 n	##########	##########
COLUMN WIDTH	##########	##########	27 81 n	##########	##########	##########
PAGE LTH. LINES	27 67 n	27 67 n	27 67 n	27 67 n	##########	##########
PAGE LTH. INCHES	27 67 0 n	27 67 0 n	##########	27 67 0 n	##########	##########
PAPER OUT "OFF"	27 56	27 56	27 56	27 56	27 56	##########
PROPORTIONAL	##########	27 112	##########	27 112	27 111	##########
RESET PRINTER	27 64	27 64	##########	27 64	##########	24

THE ABOVE ARTICLE COULD NOT HAVE COME AT A BETTER TIME. I HAVE RECEIVED MANY QUERIES ON HOW TO CONTROL A PARTICULAR PRINTER, WELL WHAT DO I KNOW ABOUT EVERY PRINTER UNDER THE SUN, SO WHEN THIS ARTICLE ARRIVED IN THE " BRAZOS VALLEY 99'ERS ", I WAS ELATED ! SO YOU HAVE ALONG WITH RICH SPRECHER'S ARTICLE QUITE A BIT OF NEW PRINTER INFO TO CHEW ON.

PRINTING MAILING LABELS WITH TI-WRITER

by Jerry Keisler

```
0001  CTRL/U FCTN/R  CTRL/U SHIFT/C CTRL/U SHIFT/I CTRL/U  cr
0002  MAILING LIST cr
0003  PARIS 99/4A USERS GROUP cr
0004  I=INDIVIDUAL  F=FAMILY cr
0005  R2=#SENT TO VISITOR cr
0006  CTRL/9
0007  JERRY KEISLER   I 5-87 cr
0008  2221 COLLEGE DR cr
0009  PARIS, TX 75460 cr
0010  CTRL/9
0011  STEPHEN BARACKMAN I 5-87 cr
0012  1330 FAIRFAX cr
0013  PARIS, TX 75460 cr
0014  CTRL/9
0015  CTRL/U FCTN/R CTRL/U @ cr
```

The easiest way I found to print mailing labels is to use TI-WRITER.

TI-WRITER starts in the word wrap ON mode indicated by a solid rectangle cursor. This is the mode you want. The other writing mode is word wrap OFF and is indicated by a hollow rectangle cursor. These modes are toggled between by pressing CTRL/0 (zero).

Any time you see CRTL/U, SHIFT/I, FCTN/R etc. The first key is held while you press the second key.

CTRL/U puts you in a special character mode in which the ASCII value of each character is reduced by 64. This mode is used for sending special operating instructions, that can not be addressed by the normal keyboard, to the printer. A second CTRL/U will turn this mode off. This mode is repesented by an underscore cursor. The table on page 146 of the TI-WRITER manual shows the ASCII number on the left, the keys to press and the symbol produced on the right.

CTRL/9 produces a funny looking "ph cr" which is the page and carriage return. The other "cr"'s are produced with the ENTER key while in the solid cursor mode.

A printer normally prints 6 lines to the inch. My labels are 1.5 inches from top to top or 9 lines. TI-WRITER seems to require 2 of the 9 lines for the page command leaving me 7 lines to use on the labels. If you are using 1 inch labels you will have 6 lines or 4 lines to print on.

Line 0001 tells my printer a page has 9 lines.The coding for my printer is:

1. (CTRL/U FCTN/R CTRL/U) or escape is ASCII 27 and tells the printer that a printer command follows and is not to be printed. This is true of most printers.
2. (SHIFT/C) or ASCII 67 which tells my printer the number of lines per page follows in ASCII. This may be different for your printer.
3. (CTRL/U SHIFT/I CTRL/U) or ASCII 9 tells my printer there are 9 lines per page.

Check your printer's instruction manual for its coding and page 146 of the TI-WRITER manual for the CRTL/U coding.

Lines 0002 thru 0005 is the first label and explains what the labels are for and any special coding on the labels. Line 0006 is a page command. Lines 0007 thru 0009 is the first address label. Line 0010 is another page command. Line 0011 thru 0013 is the second label.

Line 0014 tells my printer (escape)@ or reset all settings.

You may insert labels by pressing CRTL/8 4 or 5 times to creat "cr"'s. then going back to the first "cr" you just made and entering the label and a CRTL/9. The "cr"'s will move to the right as you type. Now delete any lines with only "cr"'s by putting your cursor on that line and pressing FCTN/3.

By using Find String or FCTN/9 FS ENTER and following the instructions you can find any label. NOTE: Your cursor should be at home or at a line number lower then the word you want to find. To print that label enter FCTN/9 PF ENTER F L PIO ENTER where F is the first line number and L is the last line number of the desired label. PIO is the printer description.

Before you print single labels or multiple labels line 0001 must be printed at least one time to set the printer to the required lines per page. When printing the whole file FCTN/9 PF ENTER PIO ENTER, this is done automatically.

MAKING ADDRESS LABELS WITH TI WRITER
BY Karl Barna

Many of you are using TI Writer or Funnelweb and when you try to make address labels you skip one label for each label printed. Well here's the answer if you have a Gemini printer. What it does is change:

a) the line feed to 9/144ths of an inches
b) page length to 2 inches
c) double strike
d) emphasized

Then it set the left and right margin. The page length is now set to 16 lines. Then it loads your mailing list file. It prints ine

It prints line 1 and feed 3 lines, prints line 2 and feeds three lines etc. up to 4 lines. When it is done printing the label you get a new page command that puts you to the next label.

What makes this program work is setting the line feed to 9/144ths of an inch because 16 of them equal one label. You could just set page length to 16 if TI Writer would let you, but that would be to easy. One thing I did that make this label just a little better looking is change the spacing between the lines try it and see what you think.

```
.TL 33:17;27;67;2;27;51;9;27;69;27;71
!
.LM 0
.RM 30
.PL 16
.ML DSK1.ADDRESS
.NF
*1*


*2*


*3*
```

XB PROGRAM EDITING WITH TI-WRITER

by Jonathan Black

How would you like to have an editor in XB in the command mode in which you could do a global search and replace, delete blocks of lines, etc?

I understand that such things are available, but until such a time as I can afford one here is the next best thing: Using TI-Writer to edit your XB programs.

To do this requres that you first list your program to disk. You do this by first loading your program into XB and then typing the following:

LIST "DSKn.FILENAME"
Where n = drive number and
FILENAME = a valid filename.

WARNING: THIS FILENAME SHOULD NOT BE THE SAME AS YOUR ORIGIONAL PROGRAM NAME. IF IT IS, YOUR PROGRAM WILL BE ERASED. I would suggest that you name it something like "XXX" or "ZZZ". After you press enter, the program will be listed to the disk.

Then, load TI-Writer or one of the other XB TI-Writer loaders. Go to the command line and type "LF" and press enter. You will then get a prompt for a file name. Enter the name of your listed program and press enter. It will then load. Now, before you do anything else press <ctrl><0> to turn off word wrap mode. Otherwise, if you accidently hit <ctrl><2> for reformat, you will end up with all of your program scrunched up on a few lines (depending of course on the size of your program). Now you can use almost all of TI-Writer's editing functions on your program.

Please note that to use the Replace String function, YOU MUST FIRST PUT A C/R ON THE END OF EVERY PROGRAM LINE AND THEN SWITCH TO WORD WRAP MODE. If this is not done, you might lose some necessary characters from your program. The reason for this is similar to the insert mode in Basic. If you go into the insert mode while editing a line, it will force characters off of the "edge of the screen" when your line reaches 4 or 5 lines in length.

You might say, "O.K. I did that. Now what do I do with it?" Well, you should now save it to disk. But, before you save it you must go through your program and make sure that on each TI-Writer line (not program line), the only number that is first on the line is the actual line number. For example:

```
100 CALL CLEAR :: FOR I=1000 TO
9999 :: NEXT I
```

Must be changed to:

```
100 CALL CLEAR :: FOR I=1000 TO
 9999 :: NEXT I
```

In this instance, 9999 would have been interpreted by the converter program to be a line number.

You must then save it by going to the command line and typing "PF" (not "SF"). You will then get a prompt for a file name. You should type the following:

C DSKn.LISTNAME

The C is there to strip out any control characters like C/R, or P/A that might have been entered either accidently or intentionally.

n = drive number
LISTNAME = a valid file name

It is okay this time to use the same filename as you used for the listing in the first place.

"Yes, but it will save it in D/V 80 format, and I can't load it back into XB?!", you say. Well that's why someone in the Atlanta 99'ers CUG wrote a program to convert a D/V 80 program listing into a D/V 163 file that you load using MERGE "DSKn.filename". Following this article is a listin of their program slightly trimmed down.

After you have saved your program listing to disk, go to XB and load this program. When you run

it you will be prompted for an input filename. Enter the name you gave your listing when you saved it. You will then be prompted for an output filename. Use a different name for this. It will then proceed to convert your listing to a merge file. after it is finished, type 'N' for no more files to do, and when it says ready type "NEW" and press enter. Then type:

MERGE "DSKn.FILENAME".

Where n = drive number and filename = the name you gave it as an output file. The merge format file will then be loaded. When it is finished, you must go through and delete the space that was inserted in front of each line. Then, when finished and before running it, save your program under a different name. Now run it and see if it worked!!

This technique for editing programs will only be really useful in long programs. For example, I have used it to customize a program to work with a 40-column assembly language sub program. One of the Replace Strings was like this: "RS" <enter> /DISPLAY AT/CALL LINK("DSPLY",/ <enter>

Also, you can write Basic or X-Basic programs using TI-Writer. Just save them the way I described and go from there. If anyone has any trouble with it, just call me. My number is on the front of the newsletter under "Librarian-Limited".

Jonathan

```
10 !**************************
20 !*D/V80 to D/V163 CONVRT*
30 !*  ATLANTA 99'er CUG   *
40 !*    modified by       *
50 !*   Johnathan Black    *
60 !* CC 99'er       12/86 *
70 !**************************
100 @=1 :: _=2 :: ON WARNING
NEXT :: DEF A(A$)=POS(A$," "
,@)-@ :: DISPLAY AT(_,@ )ERA
SE ALL:"ASCII FILE CONVERTER
" :RPT$("-",20): : : : :
110 DISPLAY AT(8,@): : : : :
"INPUT FILE? DSK1." :: ACCEP
T AT(12,18)SIZE(-12)BEEP :B$
:: DISPLAY AT(14,@):"OUTPUT
FILE? DSK1." :: ACCEPT AT(14
,19)SIZE(-12)BEEP:C$
120 OPEN #@:"DSK1."&B$,INPUT
:: OPEN #_:"DSK1."&C$,OUTPUT
,VARIABLE 163 :: CALL CLEAR
:: PRINT "ONE MOMENT...": :
:
130 LINPUT #@:D$ :: IF D$=""
OR LEN(D$)<3 THEN 130
140 IF EOF(@)=[ THEN LINPUT
#@:E$ :: D=ASC(E$):: IF D<49
OR D>57 THEN D$=D$&E$ : GOTO
 140 ELSE GOSUB 170 :: D$=E$
:: GOTO 140
150 D$=E$ :: GOSUB 170 :: PR
INT #_:CHR$(255)&CHR$(255)::
    CLOSE #@ :: CLOSE #_
160 PRINT : :"THE FILE: ":SE
G$(C$,3,LEN(C$)-_):"HAS BEEN
CREATED": :"DO ANOTHER FILE?
" :: CALL HCHAR(23,20,30)::
GOSUB 180 :: CALL HCHAR(23,2
0,B):: CALL CLEAR : IF C=@ T
HEN 110 ELSE END
170 D$=SEG$(D$,@,162):: D=A(
D$):: F=INT(VAL(SEG$(D$,@,D)
)/256):: G=VAL(SEG$(D$,@ ,D)
  )-(F6):: PRINT D$ :: D$=C
HR$(F)&CHR$(G)&SEG$(D$,D+@,L
EN(D$)-D)&CHR$([):: PRINT #_
:D$ :: RETURN
180 CALL KEY(0,K,S):: IF NOT
S THEN 180 ELSE IF K=89 OR K
=121 THEN C=@ :: RETURN ELSE
C=_ :: RETURN
```

GRAPHS USING TI-WRITER
by Jack Coleman

Wouldn't it be nice to include some text along with a graph chart? This program will convert a graph made with BUSINESS GRAPHS 99 (available trough Triton) into a file which can be printed using TI-Writer.

```
100 REM GRAPH99 PROGRAM
110 REM GRAPH CONVERTER
120 REM USE WITH BG99
130 REM BY MIKE McCANN MODIFIED BY JACK COLEMAN
140 REM
150 CALL CLEAR
160 PRINT "ENTER NAME OF BG99 GRAPHFILE"
170 INPUT FN$
180 OPEN #1:FN$,INPUT,DISPLAY,VARIABLE 132
190 INPUT "ENTER NEW FILE NAME ":FN2$
200 OPEN #2:FN2$&"/GPH",DISPLAY,VARIABLE 80
210 CALL CLEAR :: CALL SCREEN(11)
220 PRINT "CONVERTING FILE TO DV80"
230 FOR X=1 TO 200
240 LINPUT #1:A$
250 PRINT #2:SEG$(A$,A,80);
260 IF LEN(A$)>80 THEN PRINT #2:SEG$(A$,81,132)
270 IF EOF(1) THEN 290
280 NEXT X
290 CLOSE #1
300 CLOSE #2 :: END
```

The procedure for converting a graph is as follows:

1. Create a graph using BG99 and print to disk using the printer function.

2. Convert the DV132 file created in step 1 to DV80 using the above program. Note: The new filename will have a /GPH extension. (When entering the new filename, you are limited to only 6 characters. The program does not check for this.)

3. Write your text using TI-Writer, then save the file as usual.

4. Use the formatter to print your text file. Instead of printing to your printer, print the file to a disk.

5. Finally, get back into the editor and load the text file created from the formatter. Using the merge option found on page 73-74 of the TI-writer manual, insert your converted graph file within the text. Then print the complete file using PrintFile from the Editor. Be sure to use a .CR extension for the printer name (ie. PIO.CR), or both the text and graph will be printed incorrectly.

```
10 !************************
12 !*                      *
14 !* TIWRITER FONT MAKER  *
16 !*                      *
18 !*  VERSION 1  1987     *
20 !*                      *
22 !* James Stringfellow   *
24 !*                      *
26 !************************
27 !
100 CALL CLEAR :: CALL SCREE
N(5):: CALL CHAR(128,"FF8181
81818181FF",122,RPT$("0",16)
,129,RPT$("F",16))
110 FOR I=5 TO 12 :: CALL CO
LOR(I,2,11):: NEXT I :: CALL
 COLOR(13,2,11,0,5,5,1,1,13,
3,2,13,4,2,13):: P$=RPT$(CHR
$(128),16)
120 X=64 :: CH=128 :: FOR R=
4 TO 17 :: DISPLAY AT(R,4):S
TR$(X);TAB(7);P$ :: X=X/2 ::
 IF X<1 THEN X=64
130 NEXT R :: CALL VCHAR(1,3
1,28,96):: Z$=RPT$(CHR$(32),
28)
140 DISPLAY AT(21,1):"zMovez
withzzWzEzRzSzDzZzXzCz":"zQz
Togglezonzoffzzz@zstopzzz":"
zKzclearzzPzprintzzMzmirrorz
":"zzzzzzzzzzzzzzzzzzzzzzzzz
zzz" :: M$,M2$=""
150 DISPLAY AT(1,7):"zTIWRIT
ERzFONTSz":Z$:Z$ :: DISPLAY
AT(18,1):Z$:Z$:Z$ :: R=4 ::
C=9 :: K=75 :: CALL SPRITE(#
1,128,16,25,65)
160 REM
170 CALL KEY(3,K,S):: Y=(K=6
9 OR K=82 OR K=87)-(K=88 OR
K=90 OR K=67):: X=(K=83 OR K
=87 OR K=90)-(K=68 OR K=82 O
R K=67):: IF K=81 THEN 190 :
: IF K=80 THEN 200 :: IF K=7
5 THEN 120
180 IF K=64 THEN 370 :: IF K
=77 THEN 330 :: R=R+Y :: C=C
+X :: R=R-(R<4)+(R>17):: C=C
-(C<9)+(C>24):: CALL HCHAR(R
,C,CH):: CALL LOCATE(#1,R*8-
7,C*8-7):: GOTO 170
190 CH=CH+1+(CH=129)*2 :: CA
LL HCHAR(R,C,CH):: FOR I=1 T
O 30 :: NEXT I :: GOTO 170
200 GOSUB 350 :: FOR C=9 TO
24 :: X=64 :: FOR R=4 TO 10
:: CALL GCHAR(R,C,G):: IF G=
129 THEN A=A+X
210 X=X/2 :: NEXT R :: FOR J
=1 TO LEN(STR$(A)):: CALL VC
HAR(J,C,ASC(SEG$(STR$(A),J,1
))):: NEXT J :: M2$=M2$&CHR$
(A):: A=0 :: NEXT C
220 FOR C=9 TO 24 :: X=64 ::
 FOR R=11 TO 17 :: CALL GCHA
R(R,C,G):: IF G=129 THEN A=A
+X
230 X=X/2 :: NEXT R :: FOR J
=1 TO LEN(STR$(A)):: CALL VC
HAR(17+J,C,ASC(SEG$(STR$(A),
J,1))):: NEXT J :: M$=M$&CHR
$(A):: A=0 :: NEXT C :: CALL
 DELSPRITE(ALL):: CALL MAGNI
FY(1)
235 CALL SPRITE(#1,128,16,18
5,233):: DISPLAY AT(24,6):"z
zzzzzzDoublezDensityzN" :: A
CCEPT AT(24,28)VALIDATE("YN"
)SIZE(-1):Q$ :: K=75 :: IF Q
$="Y" THEN K=K+1
240 OPEN #1:"RS232.BA=4800"
250 PRINT #1:CHR$(27)&"A"&CH
R$(7)
260 PRINT #1:CHR$(27)&CHR$(K
)&CHR$(16)&CHR$(0)&M2$&CHR$(
13)
270 PRINT #1:CHR$(27)&CHR$(K
)&CHR$(16)&CHR$(0)&M$ :: CLO
SE #1
280 DISPLAY AT(24,6)SIZE(23)
:"Printzagainzyeszorznoz N" :
: ACCEPT AT(24,28)VALIDATE("
YN")SIZE(-1):Q$ :: IF Q$="Y"
 THEN 235
290 DISPLAY AT(24,6):"zzzzzz
zzzzSavezyesznozN" :: ACCEPT
 AT(24,28)VALIDATE("YN")SIZE
(-1):Q$ :: IF Q$="N" THEN 14
0
300 CALL LOCATE(#1,185,145):
: DISPLAY AT(24,2):"zzzFilen
amezDSK" :: ACCEPT AT(24,17)
SIZE(-12):F$ :: IF F$="" THE
N 140 :: OPEN #2:"DSK"&F$,FI
XED 80
310 PRINT #2:CHR$(27)&"K"&CH
R$(16)&CHR$(0)&M2$
320 PRINT #2:CHR$(27)&"K"&CH
R$(16)&CHR$(0)&M$ :: CLOSE £
2 :: GOTO 140
330 FOR R=4 TO 17 :: X=64 ::
 FOR C=9 TO 24 :: CALL GCHAR
(R,C,CH):: IF CH=128 THEN CA
LL HCHAR(R,C,129,1)ELSE IF C
H=129 THEN CALL HCHAR(R,C,12
8,1)
340 NEXT C :: NEXT R :: CH=C
H+1+(CH=129)*2 :: GOTO 140
350 DATA 80,76,69,65,83,69,3
2,87,65,73,84
360 CALL DELSPRITE(#1):: CAL
L MAGNIFY(2):: FOR I=2 TO 22
 STEP 2 :: READ Y :: CALL SP
RITE(#I,Y,16,I*8-7,I+I*8-7+1
6):: NEXT I :: RESTORE :: RE
TURN
370 END
```

DIS/VAR CONVERTER PROGRAM
by James Stringfellow

```
100 !************************
110 !    TRANSLATES FROM
120 ! DIS/VAR 80 TO MERGE
130 !        FORMAT
140 !************************
150 !
160 !USE A FULL SCREEN
170 !EDITOR TO CREATE
180 !EXTENDED BASIC PROGRAM
190 !
200 !CREATE A FILE USING
210 !TI-WRITER - MAKE
220 !SURE YOU DISABLE THE
230 !WORD WRAP MODE AND
240 !LIMIT THE LENGTH
250 !TO 80 CHARACTERS
260 !
270 CALL CLEAR
280 DISPLAY AT(3,7)BEEP ERAS
E ALL:"***TRANSLATE***"
290 DISPLAY AT(7,5):"DIS/VAR
 80 FILENAME:": :"DSK1."
300 ACCEPT AT(9,4)SIZE(-12):
IN$
310 DISPLAY AT(12,15)BEEP:"M
ERGED OUTPUT FILENAME:":"DSK
1."
320 ACCEPT AT(14,4)SIZE(-12)
:OUT$
330 OPEN #1:"DSK"&IN$
340 OPEN #2:"DSK"&OUT$,VARIA
BLE 163
350 LINPUT #1:L$
360 S=POS(L$," ",1)
361 A$=SEG$(L$,S+1,80)
370 ON ERROR 490
380 N=VAL(SEG$(L$,1,S))
390 ON ERROR 440
400 !
410 PRINT L$
420 PRINT #2:CHR$(INT(N/256))
)&CHR$(N-256*INT(N/256))&CHR
$(131)&A$&CHR$(0)
430 GOTO 350
440 PRINT #2:CHR$(255);CHR$(
255)
450 CLOSE #2
460 PRINT :: "ENTER""NEW"" A
ND THEN ""MERGE"" THE TRANSL
ATED FILENAME:":";OUT$: : :
470 PRINT "REMEMBER TO REMOV
E THE      LEADING ""!"" IN
EVERY LINE.": : : :
480 END
490 ON ERROR 440
500 RETURN 350
```

CREATING BOXES WITH YOUR PRINTER

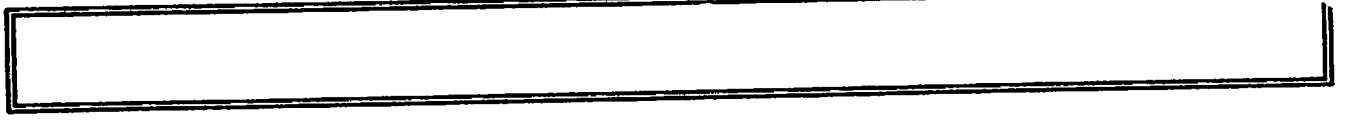
Using the transliteration feature of TI-WRITER you can create custom boxes with your IBM compatible printer. Use these boxes to dress up text or emphasize portions of a report. You can even make custom forms using the IBM character set graphics available with your printer. Using the transliterates:

```
.TL 91:201
.TL 93:187
.TL 96:205
.TL 124:186
.TL 123:200
.TL 125:188
```

WILL CHANGE THIS:

```
[----------]
|          |
|          |
{----------}
```

INTO THIS:



??? GUESS WHO ???

Put yourself in this situation... Imagine yourself watching late night TV. The movie is PSYCHO. All of a sudden, the mail slot in your front door clinks, your man eating doberman, Killer Magoo, jumps up and tears a path to the door, knocking over the maroon and green vase your mother in law gave you for Christmas, sending it crashing to the marble floor.

"WOOF! WOOF! WOOF!" Now What! You think as you rush over to check out the ruckus. There, sticking half way in the mail slot, your mind reels! You squint your eyes to see in the semidarkness of the foyer. Dumb dog! you think to yourself, as Killer again bangs into the oaken door, knocking himself silly. Your eyes see it but you cannot believe it...a floppy disk???

You're moving fast now. You quickly fling open the door, too late, you are only in time to catch a glimpse of a shiny, long, black car fading off in the darkness. RATS! It's HIM again!

YESIREE! Our mystery man has done it again, this time a real doozie of a program which turns TI-Writer into a data base manager for a name and address file and a label maker. TI-Writer always had a mail merge option which allowed you to put in a value file for use in your text. Fancier word processors like the WANG uses fancier terminology like "glossary". No big deal, they only cost $40,000 more, they can use "glossary".

Only thing, I have never seen our office WANG include sorted mail labels to go along with the form letters. HA! Our secret recluse writer, Bill, has included a program to print out mailing labels and hints for using a RAM Sort program with the value file. The thing that's really neat is that most mail label programs do not allow you to selectively print out only a portion of the name and address file. This one does.

Take it away Bill...Someday, I'm gonna find out who you REALLY are.

```
To: *1* *2*
    *3*
    *4*


Dear *1*
```

A program to help TI-Writer users when doing form letters.

TI-Writer makes it easy to put values into a memo, letter, article, essay, term paper or any other correspondence. Just enclose the text that is variable within the asterisks. The text for the value can be up to 77 characters long.

One needs to set-up a file of the values, which is used by the Formatter when using the Mail Merge option. If there is no value file, or the value for the # within the asterisk, you'll be prompted to enter the value as the document is printing. This is nice for values/text that changes often, ie. "your current status is: *5*"

The value file for this note would look like this:

```
1 Mrs. Nancy D.
2 Anderson
3 8824 S. Milltown Street
4 Chino, CA 91710
*
1 Andy
2 Jones
3 Apt #7 - West Hillsbourgh Avenue
4 Pomona, CA 91807
*
```

You can print an original letter to each

(CONTINUED)

person in the value file. You could have as a 5th value for each recipient - a special, personal, individualized message or comment. It's also nice for long words or phrases that are used several times in a memo, since you just type in the *## referring to the long (77 char. max.) value in the value file. One can print all or selected values -DATA SETS- (separated by asterisks.)

However, when sending these "form" notes to people, we usually want labels also. Since I couldn't figure out a quick way to directly use the value file for Name Address labels, I wrote a program to do it. I got tired of maintaining both a value file and a label file. It's a very short, simple, fast program and could run in BASIC or EXTENDED BASIC.

The clincher was the availablitiy of a true Mail Listing. A listing can have lots of status codes also and one can sort it on different criteria. One of the best sorters of TI-Writer files is TI-SORT by R. Romer and J. Clulow. It's a super fast assy prog that one just chooses off the TI-Writer menu as #3 UTIL1. It's a RAM sort and is limited to 300 records, so I limited my program to 300 also. TI-SORT allows you to sort on two different fields. A major and minor sort criteria. (EDITOR'S Note - This sort program is in the BUG library as FREEWARE. Get the Utility 7 disk, page 74. You will need to write your own parameters to use this sorter with this program.)

Anyway, back to my program to create Value files and Labels and get a printout of a Listing... It works off a file you create, usually in TI-Writer with a line for each DATA SET or each Name Address record. The first record/line is a Label of the file (whatever you want. ie. date, diskfile name, content desc.) To create the file is EASY. In TI-Writer just set TABS at the indicated positions and type away. You can easily customize the file and program as it's fully explained and really simple.

The program I wrote has one feature not found in most mail listers...you select the block of records (Name & Addresses) to be printed. You enter the # of the first one and the # of the last one (or just 300 if you want them all). Now we have a full data base system (TI-Writer for the great file data entry and editing and block move, copy, delete, save, combine capabilities plus my program to save a little time and TI-SORT).Again, my laziness has forced me to a better solution to a problem. (EDITOR'S Note - the program follows. See listing for TIWRITMAIL.)

-- E X P L O R E -- in Harms' Way

TIWRITMAIL PROGRAM LISTING

```
100 CALL CLEAR
110 DISPLAY "TIWRITMAIL";;;;
120 DISPLAY "   by Bill Harms";;;;
130 PRINT TAB(10);"utility for:";;;;
140 DISPLAY "Printing Labels-1up-3""x1""
"
150 PRINT TAB(15);"&"
160 DISPLAY "TI-Writer Value Files for
form letters, etc in the Mail-Merge optn
";;;;
170 PRINT TAB(15);"&"
180 DISPLAY "a Plain list of mail
file";;;;"##############################"
190 DISPLAY "It reads a D/V80 file
you've setup in TI-Writer or other";;;;
200 DISPLAY "KEEP ALPHA LOCK KEY DOWN";;
;;
210 CALL SCREEN(13)
220 FOR A=1 TO 12
230 CALL COLOR(A,16,13)
240 NEXT A
250 DISPLAY "B= Background. P= Proceed."
260 CALL KEY(0,K,S)
270 IF S<1 THEN 260
280 IF K=80 THEN 860
290 IF K=66 THEN 310
300 GOTO 260
310 RESTORE
320 DATA This prog works off a file
330 DATA you can create in TIWriter.
340 DATA Just enter the names and
350 DATA addresses and codes.
```

(CONTINUED)

```
360 DATA Then use The SaveFile
370 DATA -with C (no Control codes)
380 DATA Set tabs at 1-5-9-13-25-
390 DATA -37-60-73-76
400 DATA Enter data starting at 1:
410 DATA 1=status        5=age
420 DATA 9=size         13=last name
430 DATA 25=first name 37=street add
440 DATA 60=city        73=state code
450 DATA 76=zip
460 DATA you could change these and
470 DATA then change this program in
480 DATA lines starting at: 1020*
490 DATA 305010
500 DATA They are explained in REMs
510 DATA Max.records is 300- TI-SORT
520 DATA Use it (freeware) to 2xsort
530 DATA by Romer/Clulow in TI-Writer
540 DATA You might want to chg the
550 DATA 3 status codes or eliminate
560 DATA them (no need or want more
570 DATA char. for name-etc.)
580 DATA See REMs at PRINT stmts.
590 DATA Just chg start # and # of
600 DATA characters in the field to
610 DATA be printed.
620 DATA I did this little utility
630 DATA because TI-Writer is the
640 DATA fastest-simplist data base
650 DATA editor known to mankind --
660 DATA but it was a pain to create
670 DATA the data files for the Mail
680 DATA -Merge option and then I
690 DATA still didn't have  labels
700 DATA for the form letters!
710 FOR A=1 TO 22
720 READ TEXT$
730 DISPLAY TEXT$
740 NEXT A
750 DISPLAY "**PRESS ANY KEY TO
PROCEED**"
760 CALL KEY(0,K,S)
770 IF S<1 THEN 760
780 CALL CLEAR
790 FOR A=1 TO 17
800 READ TEXT$
810 DISPLAY TEXT$
820 NEXT A
830 DISPLAY "**PRESS ANY KEY TO
PROCEED**"
840 CALL KEY(0,K,S)
850 IF S<1 THEN 840
860 DISPLAY ;:;:"Enter disk file name
to be used or END to END PROG."
870 DISPLAY "  DSK1.filename **"
880 REM ON
890 INPUT F1$
900 IF F1$="" THEN 890
910 IF LEN(F1$)>15 THEN 890
920 IF F1$="END" THEN 1360
930 IF SEG$(F1$,1,3)<>"DSK" THEN 890
940 IF (VAL(SEG$(F1$,4,1))<1)+(VAL
(SEG$(F1$,4,1))>3)THEN 890
950 OPEN #1:F1$,INPUT ,DISPLAY ,VARIABLE
80
960 INPUT #1:LABEL$
970 CALL CLEAR
980 DISPLAY "source file: ":F1$;:;:
990 DISPLAY "This is the 1st line/record
in the file.  It describes the file
content";:;:
1000 DISPLAY LABEL$;:;:
1010 INPUT #1:ITEM$
1020 DISPLAY SEG$(ITEM$,1,12)
1030 DISPLAY SEG$(ITEM$,13,24)
1040 DISPLAY SEG$(ITEM$,37,22)
1050 DISPLAY SEG$(ITEM$,60,21);:;:
1060 DISPLAY "Press 1 to Proceed   or
Press 2 to Read a diff. file";:;:
1070 CLOSE #1
1080 CALL KEY(0,K,S)
1090 IF S<1 THEN 1080
1100 IF K=49 THEN 1130
1110 IF K=50 THEN 860
1120 GOTO 1080
1130 DISPLAY ;:;:"Enter output device
name";:
1140 DISPLAY "ie. PIO or
RS232.BA=4800.LF"
1150 DISPLAY "    or DSK1.VALUEFILE6"
1160 INPUT F2$
1170 IF F2$=F1$ THEN 1160
1180 OPEN #2:F2$,OUTPUT,DISPLAY ,
VARIABLE 80
1190 REM   IF SEG$(F2$,1,3)="DSK"
THEN 1200
1195 PRINT #2:CHR$(7)
1200 CLOSE #2
1210 CALL CLEAR
1220 DISPLAY "L for Labels":"V for
Value File":"M for Mail list view -
print"
1230 DISPLAY "R to  Rerun prog/use
different files"
1240 INPUT CH$
1250 IF CH$="" THEN 1240
1260 IF (CH$<>"M")*(CH$<>"L")*(CH$<
>"V")*(CH$<>"R")THEN 1240
1270 REM
1280 CALL CLEAR
1290 IF CH$="L" THEN 1410
1300 IF CH$="M" THEN 2270
```

(CONTINUED)

```
1310 IF CH$="R" THEN 1350
1320 IF CH$="V" THEN 1770
1330 CLOSE #1
1340 CLOSE #2
1350 GOTO 100
1360 END
1370 REM
1380 REM
1390 REM
1400 REM
1410 REM body of labels routine
1420 DISPLAY "Labels printer
routine";:;:"Load labels in printer and
Press -P- to Proceed"
1430 CALL KEY(0,K,S)
1440 IF S<1 THEN 1430
1450 IF K<>80 THEN 1430
1460 GOSUB 2850
1470 OPEN #2:F2$
1480 DISPLAY ;:;:"While labels are
printing you can:";:;:"Press -P-
Pause(on/o ff)":"Press -R- Restart print
":"Press -M- Menu"
1490 OPEN #1:F1$,INPUT ,DISPLAY ,
VARIABLE 80
1500 INPUT #1:ITEM$
1510 IF FROM=1 THEN 1560
1520 FOR A=1 TO FROM-1
1530 IF EOF(1)THEN 1740
1540 INPUT #1:ITEM$
1550 NEXT A
1560 FOR A=FROM TO TOO
1570 IF EOF(1)THEN 1740
1580 INPUT #1:ITEM$
1590 K=0
1600 S=0
1610 CALL KEY(0,K,S)
1620 IF S=1 THEN 1630
1625 GOSUB 2140
1630 REM print 1st 3 char of record
( status code)
1640 PRINT #2:"        ";SEG$(ITEM$,1,3)
1650 REM print first name (from char25
for length of 11) *  last name-(from
char 13 for 11 char. len
1660 PRINT #2:SEG$(ITEM$,25,11);" ";
SEG$(ITEM$,13,11)
1670 REM street address -from
char 37 for 22 len
1680 PRINT #2:SEG$(ITEM$,37,22)
1690 REM city, state, zip (from char 60
for 21 len)
1700 PRINT #2:SEG$(ITEM$,60,21)
1710 PRINT #2:""
1720 PRINT #2:""
1730 NEXT A
1740 CLOSE #2
1750 CLOSE #1
1760 GOTO 1210
1770 REM body of value file routine
1780 DISPLAY "Value File maker for TI-
Writer- Mail Merge option for form
letters, etc."
1790 DISPLAY :;"The value file will look
like this:"
1800 DISPLAY " 1 first name":
" 2 last name":" 3 street address":
" 4 city, state zip"
1810 DISPLAY " *";:;:
1820 DISPLAY ;:;:"While value file is
disking you can:";:;:"Press -P-
Pause(pn/off)":"Press -R- Restart print
":"Press -M- Menu"
1830 GOSUB 2850
1840 OPEN #2:F2$,OUTPUT
1850 OPEN #1:F1$,INPUT ,DISPLAY ,
VARIABLE 80
1860 INPUT #1:LABEL$
1870 IF FROM=1 THEN 1920
1880 FOR A=1 TO FROM-1
1890 IF EOF(1)THEN 2110
1900 INPUT #1:ITEM$
1910 NEXT A
1920 FOR A=FROM TO TOO
1930 IF EOF(1)THEN 2110
1940 INPUT #1:ITEM$
1950 REM print first name (from char25
for length of 11)
1960 PRINT #2:"1 ";SEG$(ITEM$,25,11)
1970 REM print LAST name (from char13
for length of 11)
1980 PRINT #2:"2 ";SEG$(ITEM$,13,11)
1990 REM street address -from char 37
for 22 len
2000 PRINT #2:"3 ";SEG$(ITEM$,37,22)
2010 REM city, state, zip (from char
60 for 21 len)
2020 PRINT #2:"4 ";SEG$(ITEM$,60,21)
2030 REM print the * (a seperator)
2040 PRINT #2:"*"
2050 S=0
2060 K=0
2070 CALL KEY(0,K,S)
2080 IF S<1 THEN 2090
2085 GOSUB 2140
2090 REM  REQUIRED REM DON'T DELETE
2100 NEXT A
2110 CLOSE #2
2120 CLOSE #1
2130 GOTO 1210
2140 REM call key for print
2150 IF (K<>80)*(K<>82)*(K<>77)THEN 2190
```

(CONTINUED)

```
2160 IF K=77 THEN 2240
2170 IF K=82 THEN 2240
2180 IF K=80 THEN 2200
2190 RETURN
2200 REM call key for above
2210 CALL KEY(0,F,T)
2220 IF T<1 THEN 2210
2230 GOTO 2190
2240 CLOSE #1
2244 IF L=83 THEN 2260
2250 CLOSE #2
2260 L=0
2261 GOTO 1210
2270 REM body of plain maillisting
2280 DISPLAY "Scan or Print of 80 column
listing":::"Press S/can or P/rint"
2290 CALL KEY(0,L,W)
2300 IF W<1 THEN 2290
2310 IF L=83 THEN 2570
2320 IF L=80 THEN 2340
2330 GOTO 2290
2340 GOSUB 2850
2350 DISPLAY ;:;:"While listing is
printing you can:";:;:"Press -P-
Pause(on/o ff)":"Press -R- Restart print
":"Press -M- Menu"
2360 OPEN #2:F2$
2370 OPEN #1:F1$,INPUT ,DISPLAY ,
VARIABLE 80
2380 INPUT #1:LABEL$
2390 IF FROM=1 THEN 2440
2400 FOR A=1 TO FROM-1
2410 IF EOF(1)THEN 2540
2420 INPUT #1:ITEM$
2430 NEXT A
2440 FOR A=FROM TO TOO
2450 IF EOF(1)THEN 2540
2460 INPUT #1:ITEM$
2470 PRINT #2:ITEM$
2480 PRINT #2:" "
2490 K=0
2500 S=0
2510 CALL KEY(0,K,S)
2520 IF S<1 THEN 2530
2522 GOSUB 2140
2530 NEXT A
2540 CLOSE #1
2550 CLOSE #2
2560 GOTO 1210
2570 CALL CLEAR
2580 DISPLAY "Scanner of a Mail file"
2590 GOSUB 2850
2592 DISPLAY ;:;:"While listing is
displaying you can:";:;:"Press -P- Pause
(on/off)":"Press -R- Restart print":
"Press -M- Menu"
2600 OPEN #1:F1$,INPUT ,DISPLAY ,
VARIABLE 80
2610 INPUT #1:LABEL$
2620 IF FROM=1 THEN 2670
2630 FOR A=1 TO FROM-1
2640 IF EOF(1)THEN 2830
2650 INPUT #1:ITEM$
2660 NEXT A
2670 FOR A=FROM TO TOO
2680 IF EOF(1)THEN 2830
2690 INPUT #1:ITEM$
2710 REM display status codes 1st
thru 11th char
2720 DISPLAY :;:;SEG$(ITEM$,1,12)
2730 REM display last name,
first name (13th for 24char)
2740 DISPLAY SEG$(ITEM$,13,24)
2750 REM display street address
(37th for 23char)
2760 DISPLAY SEG$(ITEM$,37,23)
2765 REM display CITY ,state,
zip(60th for 21char)
2770 DISPLAY SEG$(ITEM$,60,21)
2780 S=0
2790 K=0
2800 CALL KEY(0,K,S)
2810 IF S<1 THEN 2820
2812 GOSUB 2140
2820 NEXT A
2830 CLOSE #1
2832 DISPLAY ;:;:"PRESS A KEY FOR MENU"
2833 CALL KEY(0,K,S)
2834 IF S<1 THEN 2833
2840 GOTO 1210
2850 DISPLAY "We""ll process the records
inthe range you specify.";:;:
2860 DISPLAY "Enter 1st number of range.
ie 1"
2870 INPUT FROM
2880 IF FROM<1 THEN 2870
2890 IF FROM>300 THEN 2870
2900 DISPLAY "Enter last number of
range.  Enter 300 to go to the end!"
2910 INPUT TOO
2920 IF TOO<FROM THEN 2910
2930 IF TOO>300 THEN 2910
2950 RETURN
```

KAREN SAYS

Age is not important unless you're a cheese.

WORD COUNTING TI-WRITER FILES

Not everyone wants or needs to know the number of words in a text file, but there are times when this information can be useful. Knowing the number of words can help editors estimate the length of an article before it is set into type. It can help newsletter writers plan for their space requirements. It can even be helpful for those freelancers who are paid for their articles by the word.

The following program was submitted by B. Davies of Leander, Texas. Called Wordcount, it is designed to count the number or words in TI-Writer text files. The program uses a DIM statement to eliminate TI-Writer printer commands from the word count. This slows the process down, but it gives a more accurate count of the number of words that will actually appear on the printed page. Readers may increase this "exception" list by increasing the number of DIM elements. The program requires Extended BASIC and a disk. The program is easy to use. At the end it will tell you the approximate number of words that are included in the file.

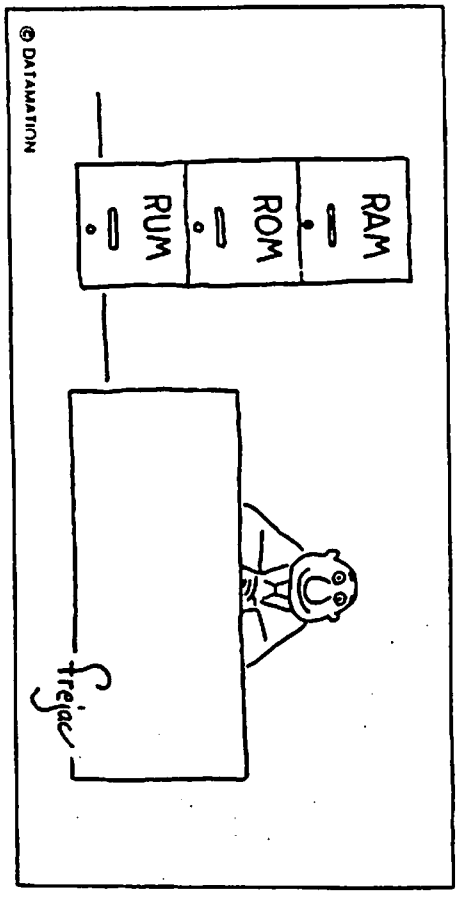
```
10 !
20 !************************
30 !     ** WORDCOUNT **
40 !      BY B. DAVIES
50 ! COUNTS WORDS IN A TEXT
60 !         FILE IN
70 ! DISPLAY/VARIABLE FORMAT
80 !     REQUIRES XBASIC
90 !************************
100 CALL CLEAR
110 DIM B$(20)
120 B$(1)=".AD"
130 B$(2)=".BP"
140 B$(3)=".CE"
150 B$(4)=".CO"
160 B$(5)=".DP"
170 B$(6)=".FI"
180 B$(7)=".FO"
190 B$(8)=".HE"
200 B$(9)=".IF"
210 B$(10)=".IN"
220 B$(11)=".LM"
230 B$(12)=".LS"
240 B$(13)=".ML"
250 B$(14)=".NA"
260 B$(15)=".NF"
270 B$(16)=".PA"
280 B$(17)=".PL"
290 B$(18)=".RM"
300 B$(19)=".SP"
310 B$(20)=".TL"
320 PRINT "ENTER TEXT FILE N
AME
330 PRINT
340 INPUT "":D$
350 PRINT
360 OPEN #1:D$,INPUT,DISPLAY
,VARIABLE 80
370 LINPUT #1:A$
380 IF EOF(1)=1 THEN 520
390 IF E=1 THEN 440
400 FOR B=1 TO 20
410 IF POS(A$,B$(B),1)=1 THE
N 370
420 NEXT B
430 E=1
440 FOR B=1 TO LEN(A$)
450 C=ASC(SEG$(A$,B,1))
460 A=((C>64)*(C<91))+((C>96
)*(C<123))+((C>47)*(C<58))
+(C=39)
470 IF A=0 THEN 480 :: D=1 :
: GOTO 490
480 IF D=0 THEN 490 :: W=W+1
 :: D=0
490 NEXT B
500 D=0 :: IF A=1 THEN 510 :
: GOTO 370
510 W=W+1 :: GOTO 370
520 PRINT "THERE ARE ABOUT";
W:"WORDS IN THE TEXT FILE"
530 END
```

Download characters to your Gemini

The following program was written by Jim Peterson of Columbus, Ohio. Many TI users know Peterson as the author of the Tips from Tigercub column that appears regularly in many user group newsletters.

The program, called DOWNCHAR, permits on-screen design of downloadable characters for Gemini printers. It is also compatible with Epson printers. The program features a direct dump to the printer for viewing the newly designed character and optional saving to disk. Peterson released the program to the public domain.

```
100 CALL CLEAR :: CALL SCREE
N(4):: CALL CHAR(128,"FF8181
81818181FF",129,RPT$("F",16)
):: CALL COLOR(13,2,16)
110 FOR R=9 TO 15 :: CALL HC
HAR(R,11,128,9):: NEXT R
120 X=1 :: FOR R=9 TO 15 ::
DISPLAY AT(R,7)SIZE(2):STR$(
X):: X=X*2 :: NEXT R :: FOR
C=9 TO 17 :: DISPLAY AT(8,C)
SIZE(1):STR$(C-8):: NEXT C
130 DISPLAY AT(2,9):"TIGERCU
B'S" :: DISPLAY AT(4,1):"GEM
INI CHARACTER DOWNLOADER" !p
rogrammed by Jim Peterson fo
r the Public Domain
140 DISPLAY AT(17,1):" Move
cursor with W,E,R,S,D,":"Z,X
 and C keys. Toggle on":"and
 off with Q key. Press":"Ent
er when finished.": : :"Pres
s any key"
150 CALL KEY(0,K,ST):: IF ST
=0 THEN 150 :: CALL HCHAR(17
,1,32,224)
160 R=9 :: C=11 :: CH=128
170 CALL HCHAR(R,C,32):: CAL
L HCHAR(R,C,CH):: FOR D=1 TO
 10 :: NEXT D :: CALL KEY(3,
K,ST):: IF ST=0 THEN 170
180 ON POS("QWERDCXZS"&CHR$(
13),CHR$(K),1)+1 GOTO 170,31
0,230,220,210,200,190,260,25
0,240,330
190 R=R+1
200 C=C+1 :: GOTO 270
210 C=C+1
220 R=R-1 :: GOTO 270
230 R=R-1
240 C=C-1 :: GOTO 270
250 C=C-1
260 R=R+1
270 R=R-(R<9)+(R>15):: C=C-(
C<11)+(C>19):: IF CH=128 THE
N 300 :: CALL GCHAR(R,C-1,GX
):: CALL GCHAR(R,C+1,GZ):: I
F (GX<>129)*(GZ<>129)THEN 30
0
280 DISPLAY AT(22,1):"You ca
n't have two in a row":"hori
zontally!" :: FOR D=1 TO 50
:: NEXT D :: DISPLAY AT(22,1
):" ":" "
290 CH=CH-1
300 CALL HCHAR(R,C,CH):: GOT
O 170
310 CH=CH+1+(CH=129)*2 :: IF
 CH=128 THEN 320 :: CALL GCH
AR(R,C-1,GX):: CALL GCHAR(R,
C+1,GZ):: IF (GX<>129)*(GZ<>
129)THEN 320 ELSE 280
320 CALL HCHAR(R,C,CH):: GOT
O 170
330 FOR C=11 TO 19 :: X=1 ::
 FOR R=9 TO 15 :: CALL GCHAR
(R,C,G)
340 IF G=129 THEN A=A+X
350 X=X*2 :: NEXT R
360 FOR J=1 TO LEN(STR$(A)):
: CALL VCHAR(15+J,C,ASC(SEG$
(STR$(A),J,1)))):: NEXT J ::
M$=M$&CHR$(A):: A=0 :: NEXT
C :: A=0
370 DISPLAY AT(20,1):"Print?
 Y/N Y" :: ACCEPT AT(20,12)V
ALIDATE("YN")SIZE(-1):Q$ ::
IF Q$="N" THEN 470
380 IF F=1 THEN 390 :: F=1 :
: DISPLAY AT(20,1):"Printer
name?" :: ACCEPT AT(20,15):P
$ :: OPEN #1:P$
390 DISPLAY AT(20,1):"ASCII
to redefine?" :: ACCEPT AT(2
0,20)VALIDATE(DIGIT)SIZE(3):
CH
400 DISPLAY AT(20,1):"Descen
der (0 or 1)? 0" :: ACCEPT A
T(20,21)VALIDATE("01")SIZE(-
1):D$ :: D=VAL(D$)
410 M$=CHR$(27)&CHR$(42)&CHR
$(1)&CHR$(CH)&CHR$(D)&M$
420 PRINT #1:M$ :: PRINT #1:
CHR$(27);CHR$(36);CHR$(1);
430 PRINT #1:RPT$(CHR$(CH),7
2):: PRINT #1:CHR$(14);RPT$(
CHR$(CH),36)
440 DISPLAY AT(20,1):"Save (
Y/N)? Y" :: ACCEPT AT(20,13)
VALIDATE("YN")SIZE(-1):Q$ ::
 IF Q$="N" THEN 470
450 IF F3=1 THEN 460 :: F3=1
 :: DISPLAY AT(20,1):"Filena
me? DSK" :: ACCEPT AT(20,14)
:F$ :: OPEN #2:"DSK"&F$
460 PRINT #2:M$
470 M$="" :: DISPLAY AT(20,1
):"Another (Y/N)? Y" :: ACCE
PT AT(20,16)VALIDATE("YN")SI
ZE(-1):Q$ :: IF Q$="Y" THEN
100
480 CLOSE #1 :: CLOSE #2 ::
END
```



TI-WRITER TEXT SORTER
by Jerry Keisler

Have you ever wanted a quick way to sort selected information, even if the information requires more room than is available in computer memory? I have, several times. I also wanted to sort just part of a report that contained tabulated information and leave the rest of the report intact.

Using the tab function in TI-WRITER with WORD WRAP OFF mode, you can type files such as a list of all titles in a series of magazines using seperate columns for language (XB, B, E/A, etc.), type of program (utility, educational, games, business, etc.), page number, issue, and article title. You can even provide an introduction and define key symbols at the top along with column headings.

This sort program will allow you to skip the heading lines, sort on primary and secondary columns and skip any lines at the bottom you don't want sorted. The file will be printed back to disk in DV80 format with your indicated file name so the old information is not overwritten.

By including only the columns required for the sort, 24,000 bytes in files larger than 24,000 bytes can be sorted. This is done in the following way:

1. The top nonsorted lines are moved from OLDFILE in drive 1 to disk drive 2 to a relative fixed 80 format file called TEMPFILE.
2. Information to be sorted is extracted from OLDFILE and put into an array along with the line (record) number. These lines are also transfered from OLDFILE to TEMPFILE intact.
3. The bottom nonsorted lines are moved to TEMPFILE.
4. The array is sorted.
5. The top nonsorted lines are moved from TEMPFILE to your NEWFILE in disk drive 1.
6. The sorted array is read from top to bottom. The record number, stored in step 2, is extracted from the first array element. That record is pulled from the TEMPFILE and placed into NEWFILE as the next line. This continues until all sorted files are read from the array.
7. All bottom nonsorted lines are moved from TEMPFILE to NEWFILE.

Some limits to this program are no more than 8,000 bytes or 300 lines can be sorted at one time. This is because of the size of string memory and the speed of an Extended BASIC sort slows down drastically with over 300 items.

It should be noted the first column in TI-WRITER is zero and the tab set function can be used to find which columns you want to sort.

This program requires: TI-WRITER files, Extended BASIC, Two disk drives, and 32K memory expansion.

```
100 DIM A$(300)
110 DISPLAY AT(1,4)ERASE ALL
:"TI-WRITER TEXT SORT":"
:: by Jerry Keisler": "PRIMARY
 AND SECONDARY SORT": :"IND
ICATE FIRST AND LAST LINE":
:"TO BE SORTED."
120 DISPLAY AT(10,1):"FIRST
COLUMN = 0": :"SEC SORT FIRS
T COL=99 TO": :"SKIP.": :"EN
D LINE = 999 FOR SORT TO": :
"END."
130 DISPLAY AT(20,1):"DISK2
FILE IS TEMPFILE": :"PRESS E
NTER" :: ACCEPT AT(22,13)BEE
P:Q$
140 DISPLAY AT(1,10)ERASE AL
L:"TEXT SORT"
150 DISPLAY AT(3,1):"SOURCE
NAME: DSK1.": :"OUTPUT NAME:
 DSK1.": :"** SORT ORDER **"
: :"1st FIRST COLUMN: 0":"SO
RT INC COLUMNS: 1": :"START
LINE: 1"
160 DISPLAY AT(13,9):"END LI
NE: 999": :"2nd FIRST COLUMN
: 0":"SORT INC COLUMNS: 1"
170 ACCEPT AT(3,19)BEEP SIZE
(-10):SN$ :: IF Q$<>"N" THEN
 DISPLAY AT(5,19)SIZE(-10):S
N$
180 ACCEPT AT(5,19)BEEP SIZE
(-10):ON$
190 ACCEPT AT(9,19)BEEP SIZE
(-2)VALIDATE(DIGIT):FC1 :: F
C1=FC1+1 :: IF FC1>80 THEN 1
90
200 ACCEPT AT(10,19)BEEP SIZ
E(-2)VALIDATE(DIGIT):IC1 ::
IF IC1<1 THEN 200 ELSE IF FC
1+IC1>81 THEN 190
210 ACCEPT AT(12,19)BEEP SIZ
E(-3)VALIDATE(DIGIT):L1
220 ACCEPT AT(13,19)BEEP SIZ
E(-3)VALIDATE(DIGIT):L2 :: I
F L2=999 THEN 230 ELSE IF L2
<L1 THEN 210 ELSE IF L2-L1>2
99 THEN L2=L1+299 :: DISPLAY
 AT(13,18)SIZE(-4):L2
230 ACCEPT AT(15,19)BEEP SIZ
E(-2)VALIDATE(DIGIT):FC2 ::
FC2=FC2+1 :: IF FC2=100 THEN
 250 ELSE IF FC2>80 THEN 230
240 ACCEPT AT(16,19)BEEP SIZ
E(-2)VALIDATE(DIGIT):IC2 ::
 IF IC2<1 THEN 240 ELSE IF I
C2+FC2>81 THEN 230
250 DISPLAY AT(18,1):"IS THI
S CORRECT? N" :: ACCEPT AT(1
8,19)BEEP SIZE(-1)VALIDATE("
YN"):Q$ :: IF Q$="N" THEN 17
0
260 !**LOADING FILE
270 DISPLAY AT(18,1):"ARE DI
SKS IN DRIVES 1  2 PE" :: A
CCEPT AT(18,28)BEEP SIZE(-1)
:Q$
280 F,N=0 :: DISPLAY AT(18,1
):" LINE #": : : :" 8000 300
 MAX":" BITS LINES SORT"
290 OPEN #1:"DSK1."&SN$,DISP
LAY ,VARIABLE 80,INPUT
300 OPEN #2:"DSK2.TEMPFILE",
RELATIVE,FIXED 80,OUTPUT
310 PRINT #2,REC 0:ON$
320 IF EOF(1)=1 THEN 430
330 LINPUT #1:B$
340 F=F+1 :: DISPLAY AT(18,1
0)SIZE(-4):F
350 PRINT #2,REC F:B$ :: IF
L1>F OR L2<F THEN 320
360 F$=STR$(1000+F):: N=N+1
370 IF FC2=99 THEN A$(N)=SEG
$(B$,FC1,IC1)&SEG$(F$,2,3)::
 GOTO 390
380 A$(N)=SEG$(B$,FC1,IC1)&S
EG$(B$,FC2,IC2)&SEG$(F$,2,3)
390 L=LEN(A$(N))+L :: Q$=SEG
$(A$(N),1,27):: DISPLAY AT(2
1,1):USING "##### ####": L,N
 :: DISPLAY AT(24,1):Q$ :: I
F L<7920 THEN 320
400 FOR DIS=1 TO 4 :: DISPLA
Y AT(2,1):: DISPLAY AT(2,1)B
EEP:"TERMANATED BY BITE SIZE
" :: NEXT DIS
410 L2=F :: DISPLAY AT(13,24
)SIZE(-4):L2
420 GOTO 320
430 CLOSE #1
440 N=N-1
450 CLOSE #2
460 CALL SORT(A$(),N)
470 !**SAVING SORTED FILE
480 OPEN #1:"DSK1."&ON$,DISP
LAY ,VARIABLE 80,OUTPUT
490 OPEN #2:"DSK2.TEMPFILE",
RELATIVE,FIXED 80,INPUT
500 DISPLAY AT(21,1): :"RECO
RD LINE": :"SAVING SORTED FI
LE"
510 IF L1=1 THEN 540
520 FOR I=1 TO L1-1 :: LINPU
T #2,REC I:B$ :: PRINT #1:B$
 :: IF I=F THEN 610
530 DISPLAY AT(21,7):USING "
####":I :: NEXT I
540 FOR I=L1 TO L2
550 J=I-L1+1 :: RE=VAL(SEG$(
A$(J),LEN(A$(J))-2,3)):: DIS
PLAY AT(21,1):USING "##### #
###":RE,I
560 LINPUT #2,REC RE:B$
570 PRINT #1:B$ :: IF I=F TH
EN 610
580 NEXT I
590 FOR I=L2+1 TO 999 :: LIN
PUT #2,REC I:B$ :: PRINT #1:
B$ :: IF I=F THEN 610
600 NEXT I
610 CLOSE #1
620 CLOSE #2
630 DISPLAY AT(24,1)BEEP:"EN
D" :: END
640 SUB SORT(A$(),N)!N=# OF
FILES TO BE SORTED
650 DISPLAY AT(24,1):"SORTIN
G" :: P=1 :: L(P)=1 :: R(P)=
N
660 IF P<=0 THEN 850
670 LB=L(P):: RB=R(P):: P=P-
1
680 IF RB<=LB THEN 660
690 I=LB :: J=RB :: T$=A$(I)
700 IF J<1 THEN 730
710 IF T$>=A$(J)THEN 730
720 J=J-1 :: GOTO 700
730 IF J>I THEN 750
740 A$(I)=T$ :: GOTO 820
750 A$(I)=A$(J):: I=I+1
760 IF I>N THEN 790
770 IF A$(I)>=T$ THEN 790
780 I=I+1 :: GOTO 760
790 IF J<=I THEN 810
800 A$(J)=A$(I):: J=J-1 :: G
OTO 710
810 A$(J)=T$ :: I=J
820 P=P+1 :: IF I-LB>=RB-I T
HEN 840
830 L(P)=I+1 :: R(P)=RB :: R
B=I-1 :: GOTO 680
840 L(P)=LB :: R(P)=I-1 :: L
B=I+1 :: GOTO 680
850 SUBEND
```

LETTERFORM

By Ollie Herbert

LETTERFORM tutorial for your word processor: BA-WRITER, FUNNELWEB, TK-WRITER (each is fairware), or TI-WRITER. I use it as a start for the letters that I write as it saves me the setup time.

TO create a LETTERFORM file, load the editor and <ENTER> the following:

```
0001 .LM 36;RM 72;FI;PL 58;HE
0002 .SP 4
0003 YOUR NAME
0004 P.O. Box 578341
0005 Chicago, IL 60657
0006
0007 .CO Tel# and SP Optional
0008 DATEXX 1989
0009 .SP 2;LM 8
0010 T.O. NAME
0011 P.O. BOX ####
0012 CITYXX, STATE ZIP##
0013 .SP2
0014 GREETING
0015 .SP;IN +5
0016 .HE 30 'slastName  Page%
0017 .CO START HERE
0018
     (19 TO 30 are also blank)
0031 .CO LAST LINE
0032 .SP 2;LM 36
0033 SALUTATION,
0034 .SP 4
0035 You R. Name
```

Function 9, T <ENTER> to get into tab settings. Function D over to column 33, R <ENTER> to set the right screen margin. Function 9, SF <ENTER>, DSKn.LETTERFORM <ENTER> to save the file that you just created as LETTERFORM.

It would be a good idea to use the formatter to make a printout of LETTERFORM at this point and make any adjustments that are necessary before you use it to write your letters. You may prefer a different arrangement, different margins, or need to insert some printer commands. You might also like to put your favorite greeting and salutation in the master.

To use the LETTERFORM file, load the editor, LF <ENTER>, DSKn.LETTERFORM <ENTER>. After the file finishes loading, edit lines 8, 10-12, 14 & 33 to reflect the data for your letter. Function 9, SF <ENTER>, DSKn.filenamexx <ENTER> using a filename of your choice as your first save. For my filenames, I use the first four letters of the addressee's last name followed by the current 6-digit date.

Start your letter by overwriting line 17. Use insert (function 2 or function 8) periodically to avoid overwriting the original line 31 which is only a flag to keep you from writing into the letter's ending. It is a good idea to function 9, SF <ENTER> (the filenamexx is retained) periodically in order to keep a fresh version of your file saved on disk. After the save, you are returned to the point where you were before the save.

When finished writing your letter, function 3 (delete) any remaining blank lines between the end of your letter and all the way down to, and including, the orginal line 31. That line won't print but delete it anyway. Function 9, SF <ENTER> to save the file. Then function 9 again, Q <ENTER> to quit and E <ENTER> to exit the editor.

You are now ready to print your letter using the formatter.

Ollie Herbert Rt. 4, Box 23 Brewton, AL 36426

TELECOMMUNICATIONS WITH TI-WRITER

Ever want to use TI-WRITER for sending information via a modem? The Magnetic Users Group, North Andover, MA has discovered a way to do it.

SENDING PARTY: - Compose text as usual with TI-Writer. When it's time to save it on disk, use PRINT FILE with no control characters:

PF <ENTER> C DSK1.README <ENTER>

Exit the EDITOR section of TI-Writer and enter the FORMAT section.

FILENAME = DSK1.README	WHAT PAGES = (ALL)
DEVICE NAME = RS232.LF	NUMBER OF COPIES = 1
USE MAILING LIST = N	PAUSE AT END OF PAGE = N

CHECK - to be sure that the RECEIVING PARTY IS READY before you toggle the sending modem on. When everything is ready, hit the final <ENTER>.

RECEIVING PARTY: - Enter the EDITOR section of TI-Writer and prepare to LOAD FILES;

LF <ENTER> RS232.LF

When the sending party is ready to send, wait until you hear the squeal of his modem, then toggle the receiving modem on and hit the ENTER key. You won't see anything on your screeen, but the lights on your expansion box will flicker. Then if everything has been done correctly, the file will suddenly appear. Then SAVE FILE to your own disk in the usual manner.

If you don't get your timing correct, you may lose part or all of one line. You can recover most of it with "OOPS!" (CTRL1), but it is easier to simply be sure your text starts with one or two blank lines.

****

PATCHING THE TI-WRITER FORMATTER

Using John Birdwell's DISKU you can alter the FORMA1 files on TI-WRITER or the FO files on FUNNELWEB to substitute other, less used, characters for the &, @ or *.

Determine from your printer's manual which replacement characters you would like to use. Note their HEX value (ie \ = 5C). Use DISKU's FIND STRING function to locate the hex string: 2A 23 21 40 26 located on sector 1. Replace the 2A(*), 40(&) and 26(@) with the hex values from your alternate characters.

You can also eliminate the double space after a period by changing the hex string >A067(located in sector 31) to >A002.

HINTS, TIPS & ANSWERS (HTA)

Author: All TIers

Editor: Bill Sponchia

Bill Sponchia
1051 Harkness Avenue
Ottawa, Ontario
Canada K1V 6N9
(613) 523-0878

Member of: Ottawa TI-99/4A UG
PO Box 2144, Stn D
Ottawa, Ontario
Canada K1P 5W3
BBS: (613)738-0617

.....

These hints, tips & answers were prepared by Bill Sponchia. The information was gleaned from various User Group newsletters and magazines. Please note that you will find NO references as to where the information was derived. In many cases the origin is not known or the same tip came from several different sources.

In addition to this excellent compendium of TI-Writer "shorts", Bill has a complete disk (aptly titled *HINTS, TIPS & ANSWERS*) which deal with Multiplan, Assembly, Extended Basic and PR-BASE. I would highly recommend this addition to your software library. If you would like to share additional tips not listed in H,T & A, please write Bill at the above address.

.....

TI-WRITER
(or equivalent)

1. Outdenting- This is the reverse of "indenting". It will allow first line of a paragraph to be start farther to the left then the remaining lines in the paragraph.
   eg - ".LM12;IN7;RM71" causes the first line to start at column 7 and subsequent lines to begin at column 12.

2. When using REPLACE STRING you must be in the same MODE (Word wrap or Block) that the original document was produced in.

3. This is for use in Funnelweb Text Editor mode (note: it may work for regular TI-Writer but I'm not sure.)
   Want to convert the character case (Upper to Lower; Lower to Upper)?
   i) Upper to Lower - depress CTRL and "." (period)
   ii) Lower to Upper - depress CTRL and ";" (semi-colon)
   By keeping the keys depressed the autorepeat function will take affect and every character the cursor passes over will be converted.

4. To save part of a document insert in front of the filename the first line number you wished saved then either a comma or a space and then the last line number you wished saved.
   eg - 5,30 DSK1.MINUTES or 5 30 DSK1.MINUTES - will save the lines 5 to 30, inclusive, onto disk drive #1 under the name MINUTES.

TI-WRITER
(or equivalent)

5. To "get rid" of the line numbers on the left of the screen press FCTN O (zero). To get them back press FCTN O again.

6. If you erase a line in error, press CTRL 1 (known as "OOPS!") and your line will be restored. Note: This will work ONLY if no other keys were pressed between the erasing and "OOPS".

7. When using the "SearcH" command remember that the search is only from the point that the cursor is located. Therefore to search the total document the cursor must be on line 1 before you go to the Command mode.

8. To backspace beyond the left margin press CTRL Y. This will temporarily disable the left margin. There is no right margin release.

9. When using the Header or Footer command with the page number it is possible to have NO value printed (such as for the introduction, etc) by using the .PA format command with a value of zero. The page numbering will begin on the following page. A .PA at the end of each page will delay the numbering further.

10. TI-Writer can save a file in other than the normal D/V80 format by using the PF command and either putting a "F" in front of the filename (ie F DSK1.MYFILE) or by putting a "C" in fron of the filename. "F" causes a file to be created in a Display/Fixed 80 format. "C" strips any control characters from the file as it is sent.

11. TI-Writer can be used as a database. Each line must be a record and set up exactly the same. For example if the data was names, addresses and phone numbers then all names must start in the same column; all addresses must start in the same column and all phone numbers must start in the same column. There can be no lines which are blank or which have other type of information on them (ie - titles). Then using the program SORT UTILITY (by D R Romer & J Clulow) you can sort this file. Once sorted, which is done very quickly, titles can be added if you are printing it out.

12. There are CTRL keys equivalents to most FCTN keys, plus a few others. For example to tab to the right you can go FCTN 7 or CTRL W; to tab back (to the left) you can go CTRL T (there is no equivalent FCTN key).

TI-WRITER
(or equivalent)

13. If you must go to the bottom of your text (and it is very long), insteadof paging down simply go to the Command Line and press "S" for "show line" and at the prompt "enter line number" just type "E" and press ENTER. "E" is a valid line number for the last line (end) of a document. This feature is active in all the commands requiring you to enter a line number.

14. If you wish to prevent the form feed at the end of printing when using the Formatter then make the last line of your text ".PL 1". This will suppress the form feed, but note: do not forget to reset PL if you have another document to print.

15. You can string the formatter commands on the same line separated with a semicolon. eg - LM 10;RM 70;IN +5;FI;AD

16. If you are having problems with formatter commands, make sure they are UPPERCASE letters.

17. To avoid a BUFFER FULL notice you just save the files as it gets larger, then use the SD command to see the file size. The BUFFER is full at 92 sectors.

18. When using the .CE command you MUST also use the .LM and .RM commands because .CE centres the text between the SET margins, not necessarily the middle of your paper.

19. The Formatter also ensures that you have two spaces after each period. To control this when you are typing such things as "Mrs. E Smith" or "1023 N. Queen Street" then use the "^" sign in place of the space after the period.
eg - Mrs.^E Smith; 1023 N.^Queen Street

20. If you must have a dot in column one of your text, transliterate it. ".TL 124:46" will allow FCTN A to print out a period.

21. To create a file with out line feeds yet Formatted, do the following:
    i) use the FORMATTER to print the text to disk
    ii) go back to EDITOR and do a Print File (PF) replacing PIO with C DSKn.filename.

22. If you wish to include a program listing in your document insteadof retyping it into TI-Writer just LIST the program to the diskette using the following command LIST "DSKn.filename". This will save the program in DISPLAY/80 format which allows it to be read by TI-Writer. You can now load this file into TI-Writer and place the carriage return character at the end of each program line.

TI-WRITER
(or equivalent)

23. If you wish to place a Carriage Return at the end of a line of text (an line without the return will usually occur when you have inserted blank lines in the text and then put text on them) then place the cursor at the end of the text and press CTRL 8. This will place a Carriage Return where you want it and insert a blank line below. If this line is not wanted you can delete it with FCTN 3.

24. You can get a print out of your file WITH LINE NUMBERS when printing out of the Editor mode by placing an "L" and a space before the printer name in the command instruction.
eg - L PIO
This will eliminate the last 6 characters at the end of each line (#75 to #80) therefore keep your line lengths to a maximum of 74.

25. Did you know that you can type anything you want after a carriage return ON THE SAME LINE and it won't print out; but it will SAVE. This is great for text notes for screen reading.

26. When typing up a document which uses certain long words or phrases frequently then a time saver (and also added insurance against typing errors) is to type the words in shortform or initials (ie - TI-ARTIST could become TIA; Ottawa TI-99/4A Users Group could become OTIUG). When you are finished with the document use the Replace String function (RS) to change the words back to the full spelling (eg - /TIA/TI-ARTIST/).
Care must be taken in three forms when using this:
   i) the search only begins from the spot that the cursor is on so to do the whole document insure that the cursor is on line 1 before starting.
   ii) the search will locate all occurrences of the string. Therefore if the string searched for is "at" it will find word "at" and also the "at" in "cat" and "that", etc. So before telling the machines to "Change all occurrences automatically be sure this program can't arise. If you are not confident of this it is best to walk through and change each separately as it is found.
   iii) as a reformatting will be done wherever a change is made it would be wise to review the document after to be sure that it is still formatted correctly.

27. When using the FIND STRING command you can specify which column range to search.
eg - 5 15 /text/ will look for the string "text" in the columns 5 through to 15 inclusive.

TI-WRITER
(or equivalent)

28. If your printer does not have a slashed zero and you want to print it out that way use the following Transliteration:
.TL 48:48,8,47
This will cause the normal zero (48) to be printed; then backspace (8); and then a printing of a slash (47).

29. Two files can be loaded into the Editor (assuming the total size is not too large for memory) by loading in the first file then doing a LF and entering E DSKn.YYY (where n=disk # and YYY=second file). This will load the second file after the end of the first file.

30. If you don't like the windowing when using the 80 column format then set the margins for 0 and 40 and turn off the line numbers (FCTN 0). When you are finished reset the left and right margins to what you desire and reformat each paragraph.

31. You can merge sections of a second file into the current document by the following entry using the LF command:
25 7 12 DSK1.YYY - This will load lines 7 to 12 (inclusive) from file YYY to the current document after line 25.

32. If you are using FUNNELWEB 4.1 or greater after you have the directory on the screen (using the SD command) you will be able to see how many lines are in a file by marking the file and then requesting (V)iew. The line count will be shown at the bottom of the screen.