T|MUG

T/MAKER USER'S GROUP NEWSLETTER
VOLUME 3, NUMBER 5, SEPTEMBER/OCTOBER 1984

In This Issue:

T/Mug: T/Maker Users Group Newsletter

T/Maker NewsFront

--Heidi Roizen

It seems every month there is so much to do around here, the newsletter slips another week every issue. So before we know it, your SEPTEMBER/OCTOBER issue is arriving well into October. I'd like to apologize and let you know we're on the road to catching up soon -- your patience has been greatly appreciated.

Where's The Anchorperson?

After a year of anchoring the T/MUG newsletter, Suzanne Osterlund has turned over the reins so that she may pursue her passion and talent in writing more fully. All of us here wanted to thank Suzanne for all she's done and wish her the best -- we look forward to seeing her articles and short stories on glossier paper than this.

Suzanne also wanted to pass along the following message to you:

"I would like to take this opportunity to share my appreciation for what you and T/Maker have given me. It has been a pleasure receiving your cards and letters. Thank you for sending in your contributions to T/MUG and thereby helping other users with your tips, tricks and suggestions. Keep them coming; we always appreciate letters from home.

"After over a year of working with T/Maker on the computer, I can now claim a relatively high degree of proficiency at this new skill. But it wasn't always that way. I remember a time when I thought 'software' was a new synonym for lingerie, when 'bytes' were something you took out of a sandwich, and when 'T/Maker' was the old enamel kettle on the stove (some users still have trouble with that one.)

"Fortunately, I have learned a lot since then. I think we all have stories like that as the overwhelming growth of this still young industry blooms before our eyes. In a year's time I have seen T/Maker Company move from Heidi's spare bedroom to a modern office complex in the center of fast-paced Silicon Valley. I have seen a list of a handful of distributors grow to nearly fifty worldwide. And I've seen the T/MUG subscription list expand dramatically!

"This progression has been an exciting thing to watch and be a part of. I hope you feel the same way because without your support, this growth wouldn't have been possible. Thank you again. As your correspondent in Mountain View, California, this is Suzanne Osterlund for TMUG NewsCenter 3 signing off.

Upgrades Still Available

Those of you who did not upgrade from T/Maker III to the new T/Maker Integrated software (all T/M III plus relational database, 55,000 word spelling checker, and on-line help) haven't missed your chance. You can still upgrade for $175. For upgrades, we need a photocopy of your T/Maker serial number, your machine type/format, shipping address, and a Mastercard or Visa number. We can also ship COD or will gladly take prepayment. There are some examples of the new pieces in action in this issue of T/MUG.

Also, those of you who would like T/Maker in a different operating system version for a second computer can purchase it for half the retail price. This will get you the software with no extra documentation (all versions use the same documentation.) For second system purchases, we need a photocopy of the serial number, machine type/format, shipping address, and payment as above.

On the ClickArt Front

We introduced another ClickArt product for the Macintosh a few weeks ago. This one, called ClickArt Publications, is designed for creating newsletters and announcements. It contains graphic images for use with MacPaint and MacWrite, including illustrated headlines and phrases, maps, a calendar, two- and three-column layout guides, and over 150 other images. The retail price is $49.95.

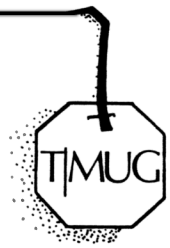
The Latest Deals

T/Maker Company and Multitech Industrial Corporation of Taipei, Taiwan have signed a distribution/OEM agreement. Not only will they distribute our English version of T/Maker, we and they together will also create a Chinese version of T/Maker.

Also, both the State of West Virginia and Nippon Univac Information Systems of Japan, two of our largest clients, have upgraded their contracts to include the new version of T/Maker with the database, spelling checker, and help command.

Big Move Completed

Our Virginia office is now officially closed, and Peter is now located here in Mountain View with the rest of us. He's still programming in his home study, though, because "the office is too noisy and everyone always wants me to be doing something else." He does frequent the office, however, and is occasionally on the phone lines here taking support calls. In fact, he's often



complimented by the callers on his excellent understanding of T/Maker. I guess most software companies do not have the head programmer (and chairman of the board) working the support hotline!

A further benefit of the Big Move: A bright addition to our office is Peter's wife Sonja, who will be our office manager. If you call, rest assured that her accent is genuine Norwegian, and you are welcomed to converse with her in Norwegian, Swedish, French, German, Spanish . . . or even English. I'm sure our foreign distributors will be pleasantly surprised next time they call. Sonja was formerly with the World Health Organization in Washington D.C.

A Special Note to Eagle II E 2 Users

One of our users, Dr. John Nadeau, has sent us a printout of an adaptation he has made to T/Maker. He uses the Eagle program called Flexible Menu to make up a menu driven approach to T/Maker. He has sent us a printout which contains the added files associated with the Flexible Menu, the modifications of the T/Maker files which were necessary, and the changes he made in TMODIFY to make the keystrokes as similar to the Spellbinder program as possible (we hope this means he is now a converted Spellbinder user!) Dr. Nadeau has kindly permitted us to share this information with other Eagle II E 2 users. If you would like a copy of the printout, please send a self-addressed 9 x 12 envelope to T/Maker and we will send you a copy.

T/ips

On Underlining

When using the backslash toggle (\) to underline, the backslash itself takes up a character space, which T/Maker does not delete during the printing process. Therefore, such a character ends up leaving a blank space. There are ways to get around leaving this blank space. One way is to place the toggles between the words, and leave no usual spaces, like this

toggle\between\words

which prints like this:

toggle between words

The shortcoming of this approach is that T/Maker views the entire connected set of words as one word for the purposes of aligning. We usually find it better to simply live with the extra spaces.

There is one place, however, where an extra space looks awkward, that being the first space of the line. In this case, a tricky solution is to put the backslash on the last position in the line preceding the line you want underlined, like this

and I say to you that you \
cannot\simply take me for granted anymore!

when printed, it will look like this:

and I say to you that you
cannot simply take me for granted anymore!

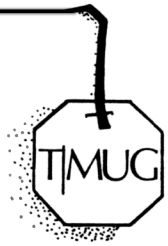
One final tip on this — spaces also look awkward when the word is the final one before a period, comma, or other punctuation mark. The toggle underline is smart enough to not underline these marks, so place the toggle off after the punctuation mark. Example:

This\line,\when printed, will look\like this.\

This line, when printed, will look like this.

Having Fun with the Spelling Checker

The spelling checker can be used for more than just checking your documents for errors. You see, the spelling checker's SPELL command allows use of "wild cards" to represent both individual letters or an indeterminate number of letters in a certain position. Here are a few suggestions for more light-hearted use of the T/Maker spelling checker.



Crossword Puzzles

Say for example you were doing a crossword puzzle. You are trying to find a word. The clue is "musical instrument" and you know from filling in other squares that it is spelled -r--n. To get T/Maker's suggestions, get to the WHAT NEXT? prompt and type

SPELL ?R??N

T/Maker will respond with the following list: brain, brown, crown, drain, drawn, drown, grain, green, groan, grown, organ, train, urban. Takes all the fun out of it, doesn't it?

Limmericks, Poems and Songs

If your a budding poet or songwriter, you can use T/Maker to help you come up with rhyming words. Let's say you wanted to find a word to rhyme with "antic". the command:

SPELL *ANTIC

will bring you atlantic, gigantic, and pedantic. And that's just with the small dictionary. The command

SPELL *ANTIC WITH EXTRA.DIC

added antic, frantic, romantic, semantic and transatlantic to my choices.

I tried

SPELL *ARTED

and got carted, charted, departed, parted, smarted, started and thwarted. Sounds like the makings for a terrific country and western song.

Word Puzzle

How about this one. A customer of ours called with an old favorite word problem of mine. She said "there are three words in the English language which end in GRY. Can you name them?" Since you don't care how many letters the words start with, the T/Maker command to do this would be

SPELL *GRY

Still not enough words? Try the command

SPELL *GRY WITH EXTRA.DIC

I suffered over this puzzle for a long time. So, suffer!

Printing Without Print Design Commands In the File

--John W Price

I often find it desirable to "print" files which do not (and cannot) contain T/Maker print design commands. BASIC language programs are a classic example, as would be most other "interpreted" language files.

In spite of the fact that these files may not contain print design commands, I want to use the automatic paging, the heading, and the footing (bottom) abilities that T/Maker provides. I also want to include special printer commands to control the page length, pitch and lines per inch of my printer. I routinely do this in T/Maker via the PRINT.UTL file and the use of high bit characters.

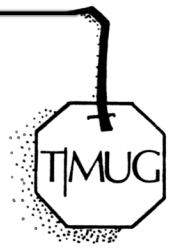
I have found a way to achieve all of this by combining the abilities of my CP/M+ SUBMIT facility, and several of the T/Maker III features. The SUBMIT file appears as a regular command file. To print a file, at the "A>" prompt enter "PRINT filename."

(CP/M+ Note: For the SUBMIT file to be treated as a COM file, the SETDEF [ORder=<COM,SUB) command sequence must have previously been issued.)

Here then is the file I use, and an explanation of how PRINT.SUB works.

```
LINE  PROGRAM CONTENT (A:PRINT.SUB)
----  ------------------------------------------------------------
(1)   TMAKER GET A:PRINT.SUB REPLACE %%% $1 ARRANGE 2 80 END 2 DO
(2)   <CLIP BEFORE ALIGN PRINT NONSTOP IT STOP
(3)   <.TOP
(4)   <S 6 5
(5)   <<<-                              >>
(6)   <%%%
(7)   <>><<
(8)   <0
(9)   <.END
(10)  <.BOTTOM
(11)  <
(12)  <5
(13)  <%%%
(14)  <.&
(15)  <                                                PAGE   #
(16)  <.END
(17)  <.PAGE 66
```

--Print Design Commands, Continued




```
(18)  <.LENGTH 61
(19)  <.INDENT 2
(20)  <.CONTINUE %%%
```

EXPLANATION

STEP	ENTRY/COMMAND	PURPOSE
1a.	TMAKER	Calls the T/Maker com file
1b.	GET A:PRINT.SUB	Tells T/Maker to get the submit file
1c.	REPLACE %%% $1	Replace every occurrence of %%% with the filename. $1 is the filename which was substituted by "CP/M submit" before the call to T/Maker (step 1a)
1d.	ARRANGE 2 80 END	This step removes the "<" from the beginning of each remaining line. The "<" symbol is there to "fool" the CP/M+ submit processor into thinking each line is an input line. Without the "<" symbol, submit would try to execute the lines as CP/M+ commands.
1e.	2	Tells T/Maker to goto line 2,
1f.	DO	and execute that line as T/Maker commands
2a.	CLIP BEFORE	Clears line 1 from the submit file. (Line 2 was cleared by the "do" command)
2b.	ALIGN	centers the "heading"
2c.	PRINT NONSTOP IT	Print (to the printer) nonstop the rest of the working area
2d.	STOP	After printing, return to CP/M
3a.	.TOP	Start of design "header"
4a.	S 6 5	"S 6 5" are actually "high bit" characters which instruct my printer (via PRINT.UTL) to print 66 lines per page, 6 lines per inch and 5 characters (expanded) per inch.
5a.	<<- >>	alignment marks to center next line

--Print Design Commands, Continued

6a.	%%%		this will be the filename as substituted from the submit facility, and replaced in step 1c
7a.	>><<		stops centering
8a.	0		High bit 0 to set printer to 10 char/inch
9a.	.END		
10a.	.BOTTOM		Start of footing
11a.			blank line
12a.	5		High bit 5 to set printer to 5 char/inch
13a.	%%%		Filename as substituted in step 1c
14a.	.&		print command to overwrite previous line
15a.		Page #	Page number is right justified by position
			Steps 13a, 14a and 15a should be viewed together. Since the filename is a variable length, the position of the page number would also vary, if they were on the same physical line. The method used here caused left justification of the filename, and right justification of the page number.
16a.	.END		end of footing
17a.	.PAGE 66		print command to set page length
18a.	.LENGTH 61		print command to determine lines per page
19a.	.INDENT 2		print command to allow for hole punching
20a.	.CONTINUE %%%		continue printing with filename. Now the object of the whole program is actually performed.

NOTE: CP/M+print (^P) should be off. Line numbers appear for explanation only. A "WARNING: PROGRAM INPUT IGNORED" message is normal, after the submit has completed. This is due to the "<" lines used to "fool" submit.

Editors Note: For those of you who want to accomplish some of the same things as John Price does in his approach, but not work through submit files, here's another idea. You could have a file called, say DESIGNS. This file would look like this:

```
.PAGE 66
.LENGTH 61
.INDENT 2
.TOP
   --------------This is the file FILENAME---------------

.END
.BOT

                                                    Page #

.END
.con FILENAME
```

To print from within T/Maker, you would use the following command:

GET DESIGNS REPLACE FILENAME (your-filename-here) PRINT IT

This is not as crafty as John's solution, as it does not allow alignment to take place. However, it does allow you to set up pagesize, indentation, headers and footers without messing up your file.

An Efficient Business Letter System

--Ron Troxell

I was just typing a bunch of letters today using T/Maker and it occurred to me some of the other T/MUGgers might find my procedure useful.

Since replies to business letters usually reference the date the original letter was written, I have established a filing procedure based on the date. For example, The file name on this letter is RT072484.08 because the date I wrote it was 07/24/84 and it was the eighth letter done that day. This procedure keeps the number of backup files to a minimum as there is only one created per day. That way my disk space is conserved.

The next trick I use is a file called DT which looks like this:

```
July 24, 1984
..indent 40
..newpage


<<                                                    ->>
Dear
```

The command line for creating a letter looks like this:

```
create RT072484.08 insert DT4 edit
```

This puts me on the screen ready to enter the name, address, etc. I then type the body of the letter and then drop to the line below the last line in the letter. I quit the editor and at the WHAT NEXT? prompt type:

```
insert RTEND 1 align save print it
```

The file RTEND looks like this:

>><<
Sincerely,

R. Ronald Troxell
Marketing Director

Having the aligning wedges preset in the DT file and putting the unaligned symbol in the RTEND file ensures that my letters are consistent and uniform, and that I don't forget to put in wedges.

Perhaps you are wondering why I have those two comment lines in the DT file? After the letter is printed, I create the address label for the envelope by typing the following command:

REPLACE .. . 3 PRINT IT

Thus, the comments become print design commands for the printer, the address is automatically indented to the appropriate spot on the envelope and, of course, printed exactly the same as on the letter. After printing the address, the print question returns to the screen because of the .newpage line. Responding with a Q for quit prevents the letter from being typed on the envelope.

That's about all there is to it. It works very conveniently for me and I hope others will find these tricks helpful too.

Tracking Softball Statistics with T/Maker

-- Michael J. McIntyre

Here's an applications program I designed for our softball team. The program computes various statistics, including Runs Created, a statistic invented by Bill James of Baseball Abstract fame. Each week, I put the new stats into a blank form labelled data0, and then combine it with the previous week's table. Then I compute the new table, relabel it, and save it to a new file. The columns are:

On Hitting & Fielding Statistics:

AB	-	at bat	PO	-	put outs
R	-	runs	A	-	assists
H	-	hits	E	-	errors
2B	-	doubles	OB	-	on base
3B	-	triples	Ap	-	appearances
HR	-	home runs	BA	-	batting average
Sac	-	sacrifices	OBA	-	on base average
K	-	strike outs	SA	-	slugging average
BB	-	walks	LBA	-	lifetime batting average
RBI	-	runs batted in	RC	-	runs created

On Pitching Statistics:

G	-	games	ER	-	earned runs
IP	-	innings pitched	BB	-	walks
H	-	hits	K	-	strike outs
R	-	runs	W	-	wins
L	-	losses	ERA	-	earned run average

Here's how I use it:

1. GET DATA0 (an empty table similar to the final output table but with the columns closer together so that the columns that have data entered into them fit onto one screen)
2. RENAME DATAN (DATA plus N, where N is the game number)
3. SAVE
4. EDIT (Enter the data from the form, and return the cursor to Home)
5. Quit the editor and SAVE
6. GET GAMEY (The last game before the current one, i.e. Y = N - 1)
7. RENAME GAMEN (GAME plus N, where N is the game number)
8. COMBINE DATAN + +=
9. EDIT (to see that all went OK) and quit the editor
10. SAVE, COMPUTE
11. EDIT (take out zeroes before decimal point, using macro, for BA OBA SA LBA. Also correct date and heading)
12. Return cursor to Home and quit the editor
13. SAVE and PRINT IT -- The table appears on the next two pages:

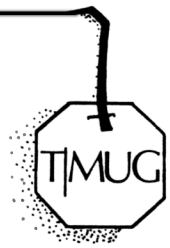
Video Horizons -- 1984 Hitting, Fielding

	Player	AB	R	H	2B	3B	HR	Sac	K	BB	RBI	PO	A	E	OE
ex		999	999	999	99	99	99	99	999	999	999	999	999	999	999
zv		-	-	-	-	-	-	-	-	-	-	-	-	-	-
ac1		+						+		+					
ac2		/		+											
ac3															+
ac4						sta	stb				stc	std	ste		
ac5*						2	3								
ac6		/		+	+	+	+								
ac7				+	+	+	+			+	=				
ac8				+						+		=			
ac9		+								+			=		
Ac10											+	*	/		
ac11		+													
ac12				+											
ac13															
ac14															
ac15						fta	ftb				ftc	ftd	fte		
+	A*****, Robbie	16	5	8	1	-	-	-	-	2	8	22	2	-	10
+	A*****, Rick	27	8	15	-	1	-	-	-	2	7	20	2	3	17
+	B*****, Cliff	16	5	7	-	-	-	-	1	1	6	7	-	-	8
+	F*****, Dave	16	2	4	1	-	-	1	-	1	4	6	1	2	5
+	H****, Mike	17	2	2	-	-	-	1	-	1	2	6	8	3	3
+	K******, Mike	14	6	2	-	-	-	-	-	5	3	2	12	2	8
+	L******, Steve	14	2	7	-	-	-	-	1	1	8	5	-	-	10
+	M*********, Bob	12	2	7	-	-	-	-	-	1	1	9	-	-	8
+	M*******, Mike	13	4	6	-	-	-	2	-	1	4	11	3	-	9
+	M*****, Ken	18	9	6	1	-	-	-	-	3	1	9	9	2	10
+	O*********, Dan	24	3	9	-	-	-	1	-	1	3	13	25	4	11
+	S*****, Pat	14	3	5	-	-	-	-	-	1	1	23	-	1	7
+	S****, Steve	10	2	5	1	-	-	-	-	-	1	3	-	-	5
+	T*****, Steve	19	6	12	1	-	-	-	-	2	4	7	5	1	14
=	TEAM	230	59	95	5	1	-	5	2	22	53	143	67	18	125

Pitching Statistics

		99	999.99	999	999	999	999	999	999	999	9.99
ex											
zv		-	-	-	-	-	-	-	-	-	-
ac16			sta								
ac17/			7								
ac18			/			+					=
ac19			fta								
	Pitcher	G	IP	H	R	ER	BB	K	W	L	ERA
+	K******	5	35.00	48	17	9	3	6	5	-	1.80
+	T*****	2	13.33	13	12	8	3	-	1	1	4.20

and Pitching Statistics (Game 7)



Ap	BA	OBA	SA	LBA	RC	83AB	83H	LH	LAB	
999	9.999	9.999	9.999	9.999	999.9	9999	9999	9999	9999	,
-	.000	.000	.000	.000	0					
=										
	=									
/		=								
			=							
					=					
						+			=	
							+	=		
								+	/	=
										stf
				ftf						
18	.500	.556	.563	.500	6.1	118	59	67	134	
29	.556	.586	.630	.475	11.1	32	13	28	59	
17	.438	.471	.438	.381	3.8	68	25	32	84	
18	.250	.278	.313	.352	1.8	106	39	43	122	
19	.118	.158	.118	.118	0.5			2	17	
19	.143	.421	.143	.356	2.6	132	50	52	146	
15	.500	.667	.500	.397	4.3	160	62	69	174	
13	.583	.615	.583	.439	4.9	29	11	18	41	
16	.462	.563	.462	.405	3.5	108	43	49	121	
21	.333	.476	.389	.270	4.3	82	21	27	100	
26	.375	.423	.375	.345	4.0	92	31	40	116	
15	.357	.467	.357	.324	2.4	57	18	23	71	
10	.500	.500	.600	.500	3.0			5	10	
21	.632	.667	.684	.479	10.0	121	55	67	140	
257	.413	.486	.443	.391	57.6	1105	427	522	1335	

The DO Command: Generating a Standard Bar Chart

-- Ron Roizen

DO may be one of T/Maker's smallest commands, but it's packed with power. DO is simple to invoke: all you need do is type "DO" after the WHAT NEXT? prompt and press the RETURN key. And what does DO do? It converts a line of text into a series of commands. How? DO removes and transports the line of text appearing at the top of your screen to a position after the WHAT NEXT? prompt. T/Maker then proceeds to regard that text as commands, just as if you had entered them conventionally.

For example, a screen that looks like this:

```
 GET LETTER.BOB ALIGN SAVE PRINT NONSTOP IT


 WHAT NEXT?  DO
```

Will look like the screen below immediately after the DO Command is invoked (that is, after the RETURN key is pressed):

```


 WHAT NEXT?  GET LETTER.BOB ALIGN SAVE PRINT NONSTOP IT
```

There are two virtues to this capacity: (1) DO permits you to store command sequences with a file, and (2) DO permits you to use all the flexibility of the Editor in constructing command sequences. Here, we consider only the first virtue. The storage of commands as text in files is T/Maker's passport to programming. It lets you construct, use, and reuse complex series of automated functions.

Automating a Standard Bar Chart's Production

Consider a simple example of this utility. I recently helped a client whose business involved the need periodically to report the results of certain measurements in a standard-format bar chart. Naturally, this task might be accomplished manually—that is to say, it might be done and redone each time the need arose. But it is much easier to establish a DO program to accomplish it automatically. Look at the individual steps of this task. To get a feel for how to use DO, the reader might wish to follow along with his own computer, constructing the exemplary files as they are described.

First of all, you will need a file in which measurements can be input. Fashion a file called "INPUT" that looks like the one below and SAVE it.

FILE: INPUT

```
|                                                                |
|example                  99999                                  |
|cn                        data                                  |
|                                                                |
|+         measure 1                                           |
|+         measure 2                                           |
|+         measure 3                                           |
|+         measure 4                                           |
|+         measure 5                                           |
|+         measure 6                                           |
|                                                                |
```

Use the "clear all tabs" and "set a tab" keystrokes and the TABS command at the WHAT NEXT? prompt to set and save a single tab under the model number. This permits you to input data directly into the desired column positions. What, then, is the process for inputting and converting such data to the client's desired bar chart? Each time a new chart needs to be made:

1. GET INPUT. Use the RETURN and TAB keys to put the data into the file, like this:

```
FILE:  INPUT
-------------------------------------------------------------------
|                                                                 |
|example                    99999                                 |
|cn                          data                                 |
|                                                                 |
|+          measure 1       235                                   |
|+          measure 2       147                                   |
|+          measure 3       198                                   |
|+          measure 4       255                                   |
|+          measure 5       302                                   |
|+          measure 6       276                                   |
|                                                                 |
-------------------------------------------------------------------
```

2. Next, RENAME the file so that you can work on it without destroying the original INPUT file. Give the command: DELETE CHART RENAME CHART <RETURN>. You first delete the name "chart" to make sure that the "chart" name is available for use.

3. Next, use the commands for designing bar charts. Suppose you desired a chart with the following characteristics: a maximum value of 500, bars 1 line wide, no space between bars, bars made up of $ characters, and a maximum bar length of 40 column positions. Give the command: BAR data MAX 500 WIDTH 1 SPACE 0 CHAR $ LEN 40 <RETURN>.

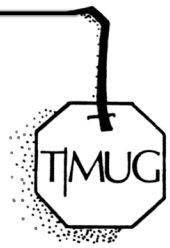
4. Finally, you must tell T/Maker what should be done with this chart. Say you want the working file to consist only of this chart, and you want T/Maker to SAVE it under the existing working file's name, CHART. Give the command: REPLACE END SAVE <RETURN>.

Having gone through these four steps you now have a file saved on disk named CHART that looks like this:

```
                  <0 data $                          500>

measure 1  235    $$$$$$$$$$$$$$$$$$
measure 2  147    $$$$$$$$$$$
measure 3  198    $$$$$$$$$$$$$$$
measure 4  255    $$$$$$$$$$$$$$$$$$$
measure 5  302    $$$$$$$$$$$$$$$$$$$$$$$
measure 6  276    $$$$$$$$$$$$$$$$$$$$$$
```

As mentioned, you might go through these four steps each time you wished a new chart. The simpler thing to do, however, is to lodge all these commands in a single string at the top of the INPUT file. With such a Command Line, after all data have been inputted, you simply have to invoke the DO Command.



This "Command Line" looks like this:

FILE: INPUT

```
|DELETE CHART RENAME CHART BAR data MAX 500 WIDTH 1 SPACE 0 CHAR $ LEN 40 REPLACE END SAVE |
|                                                                                          |
|example                  99999                                                            |
|cn                        data                                                            |
|                                                                                          |
|+          measure 1                                                                      |
|+          measure 2                                                                      |
|+          measure 3                                                                      |
|+          measure 4                                                                      |
|+          measure 5                                                                      |
|+          measure 6                                                                      |
|                                                                                          |
```

Be sure that the Do Command Line is the top line of the file. DO Command Lines can extend well beyond the right side of your computer's screen. Hence, it's a good idea to work in Carriage Return Mode "T" when constructing DO Command Lines. With the INPUT file outfitted in this way, you can reduce the job of making the bar chart to just two steps.

1. GET INPUT. Enter the Editor. Fill in the data.

2. Home the cursor, quit the Editor, and type DO <RETURN>.

Linking DO Command Lines

DO Command Lines can be linked to one another to form more elaborate automated series. For example, suppose that you wanted to dress up your bar chart a little by giving it a standard top and bottom border and a title. With embellishments it looks like this:

```
==========================================================
            CHART OF SIX STANDARD MEASUREMENTS
----------------------------------------------------------
                  <0 data $                          500>

measure 1  235    $$$$$$$$$$$$$$$$$$
measure 2  147    $$$$$$$$$$$
measure 3  198    $$$$$$$$$$$$$$$
measure 4  255    $$$$$$$$$$$$$$$$$$$
measure 5  302    $$$$$$$$$$$$$$$$$$$$$$$
measure 6  276    $$$$$$$$$$$$$$$$$$$$
==========================================================
```

Another DO Command Line will easily provide these embellishments. How?

First, CREATE a file named BORDER. Use the Editor to design its contents as below and then SAVE it:

```
.block
>><<

============================================================
            CHART OF SIX STANDARD MEASUREMENTS
------------------------------------------------------------
..here
============================================================
.end
```

The ".block" and "..here" Design Commands begin in column-position 1. Once having created this file, you have the wherewithall to automate the process of surrounding your bar chart with these embellishments. The command DELETE FINAL.CHT RENAME FINAL.CHT FIND here INSERT CHART 1/1 SAVE <RETURN> will carry out that very task. This command serves to insert the already prepared bars into a prefabricated set of borders and title. Now, place this command on the first line of the file named BORDER and SAVE the file. Once this is done, your new BORDER file should look like this:

```
DELETE FINAL.CHT RENAME FINAL.CHT FIND here INSERT CHART 1/1 SAVE
.block
>><<

============================================================
            CHART OF SIX STANDARD MEASUREMENTS
------------------------------------------------------------
..here
============================================================
.end
```

All that now remains to be done is to add a small instruction to the end of the INPUT file's DO Command Line. You must add the instruction that tells T/Maker to GET the BORDER file and carry out its DO Command Line. Thus, the revised Command Line on the file named INPUT will be:

DO command -- continued

FILE: INPUT

```
DELETE CHART RENAME CHART BAR data MAX 500 WIDTH 1 SPACE 0 CHAR $ LEN 40 REPLACE END SAVE GET BORDER DO
example                  99999
cn                        data

+         measure 1
+         measure 2
+         measure 3
+         measure 4
+         measure 5
+         measure 6
```

Now, you have two files, one named INPUT and the other named BORDER. To use them you GET the INPUT file and put fresh data into it by means of the Editor. That done, home the cursor, quit the Editor, and invoke the DO Command. The DO Command will thereafter automatically produce a standard bar chart of a format you have designed and then embellish that chart in borders and a title also of your design. That finished chart is now available under the file name FINAL.CHT. FINAL.CHT automatically comes out looking like this:

```
==========================================================
            CHART OF SIX STANDARD MEASUREMENTS
----------------------------------------------------------
                  <0 data $                          500>

measure 1  235    $$$$$$$$$$$$$$$$$$
measure 2  147    $$$$$$$$$$$
measure 3  198    $$$$$$$$$$$$$$$
measure 4  255    $$$$$$$$$$$$$$$$$$$
measure 5  302    $$$$$$$$$$$$$$$$$$$$$$$
measure 6  276    $$$$$$$$$$$$$$$$$$$$
==========================================================
```

Making Your own T/Maker Commands with HELP

--Heidi Roizen

Those of you who have updated to our new Integrated package may have tried out the on-line help facility included in the new package. If you did, you found a series of information screens which you can request by typing particular headings or acronyms. If you tried the optional command menu, you found you can execute commands like INFO, CREATE, and SAVE without typing the command at the WHAT NEXT? prompt, but rather by picking the command from a menu.

Both of these features, HELP and the command menu, are actually copies of the same program. What you may not have noticed is that there are four of five pages nestled in the back of your spelling checker documentation which explain how to use this little program to set up your own menu-driven systems or information screens. It's fairly easy to set up, and can be used as a source of trivial fun (as in this example) or to make your applications simpler for other people to run without your instruction. Here's a brief example of how to do it:

You start the process while still at your operating system prompt. Let's say you want to have a T/Maker command called PANIC. While at your operating system prompt, copy the HELP.TMK file, calling it PANIC.TMK.

You then enter T/Maker, and CREATE a file called PANIC.1. this file is where the PANIC command will look for its instructions. You can have more than one instruction file, but the first one must be called PANIC.1. This example will deal with only one instruction file. An instruction file must have a certain structure, as follows.

The first line of the PANIC.1 file will be used as the prompt when this command is issued. For the PANIC command, let's say we want the prompt to be "WHAT ARE YOU WORRIED ABOUT (OR END)?" END is always the response which takes you out of your created command and returns you to the WHAT NEXT? prompt.

Following the prompt line are lines that tell the new command where the various subjects are. Think of it as a table of contents of files and subjects. Since we are dealing with only one file in this example, we need only one line for our table of contents for the PANIC command. (To see a bigger example, edit your file HELP.1 and see the prompt and the table of contents in it.) Our file, with the prompt line and the table of contents line, now looks like this:

```
WHAT ARE YOU WORRIED ABOUT (OR END)?
= PANIC.1 MOM WORK MONEY FAT
```

--HELP command, continued

Next we need to design the first screen. It is the one which will be displayed when you first enter the PANIC command. Here it is:

```
--------  The T/Maker Panic Problem Solver  --------

My problem is:                              TYPE:

1.  My mother feels neglected                 MOM

2.  Too much work                            WORK

3.  I'm broke again                         MONEY

4.  I need to lose weight                     FAT

-------------------------------------------------
```

This and all subsequent information screens must have at their start a plus sign in the first column, followed by a level number. The level numbers are used to control which screens in a file are presented for a given subject. In this example, each subject will get only one screen, so the level numbers are all the same. After the level number comes the subject that the user would type to get to that screen.

Here is the way the PANIC file now looks:

```
WHAT ARE YOU WORRIED ABOUT (OR END)?
= PANIC.1 MOM WORK MONEY FAT

+ 1

--------  The T/Maker Panic Problem Solver  --------

My problem is:                              TYPE:

1.  My mother feels neglected                 MOM

2.  Too much work                            WORK

3.  I'm broke again                         MONEY

4.  I need to lose weight                     FAT

-------------------------------------------------
```

--HELP command, continued

```
+ 1 MOM

Send her some flowers, give her a call, and tell her you love her.
If this doesn't solve the problem, well, I'm only a computer.
I really don't know anything about interpersonal relationships.

+ 1 WORK

1.  Use T/Maker for more of your common problems
2.  Delegate
3.  Quit

+ 1 MONEY

1.  Stop paying taxes.
2.  Rob a bank.
3.  Get a higher paying job.
4.  Spend less.

+ 1 FAT

1.  Have your mouth sealed.
2.  Go on the alcohol diet -- you don't eat and you don't care.
3.  Exercise more.
4.  Eat only lettuce, celery and cabbage.


     (end of file)
```

With this file saved as PANIC.1 on your T/Maker disk, and your newly created PANIC.TMK file on the disk, you're ready to have T/Maker solve all your problems. Enter T/Maker and at the WHAT NEXT? prompt type PANIC. You will be presented with the first screen, and the prompt at the bottom will be WHAT ARE YOU WORRIED ABOUT (OR END)? You can choose to type MOM, FAT, MONEY, or WORK, after which you will be presented with the appropriate solution on the screen. Try it and see!

This is a rather trivial example in that it is only concerned with displaying screens of text, and uses only one file. The manual describes how you can create such commands which can have not only text screens, but built-in executable commands as well. This idea could be used to set up menus for your most common tasks. For example, you could set up a REPORT.TMK command and a REPORT.1 file which looks like this:

```
WHAT IS YOUR CHOICE (OR END)?
+ 1
- LETTER align print nonstop it
- TABLE compute clean print nonstop it

   ------------------------------------------------
   Type LETTER to align and print your letter
   Type TABLE to compute, clean, and print your table
   ------------------------------------------------
```

If you would type REPORT at the WHAT NEXT? prompt, your screen would look like this:

```
|                                                                |
|    ------------------------------------------------            |
|    Type LETTER to align and print your letter                  |
|    Type TABLE to compute, clean, and print your table          |
|    ------------------------------------------------            |
|                                                                |
|                                                                |
|                                                                |
|                                                                |
| WHAT IS YOUR CHOICE (OR END)?                                  |
|                                                                |
|________________________________________________________________|
```

Now all you need to do is type LETTER or TABLE, and the command line associated with that command would be executed. Note also that calling up a command like this does not affect your working file -- but the command issued can work on your working file.

You can look at the Command menu file to see an example of this type of menu system in action, or try entering the one above.

My Favorite <RULES>

-- Peter Roizen

Here are a few rules that you may find useful with the SET command.

```
tax 9,999.99 = 0.065 * amount when tax <> " "
```

Simply putting any nonblank character in the "tax" field will cause the tax to be calculated. This is easier than having a separate field that says whether or not the tax should be calculated.

```
ship 9,999.99 " " = 5.00 when ship <> " " and ship numeric = 0
```

Shipping charges are set to $5.00 when some non-numeric character is put in the shipping field. However, if a numeric value is entered into the shipping field, it is left as entered. This is a nice approach when a field typically gets the same value, but you need the facility to enter a specific value in a few cases.

Calculating Elapsed Time

```
time = (outhour - inhour) * 3600       when outhour >= inhour
time = (12 + outhour - inhour) * 3600 when outhour <  inhour
time = time + 60 * (outminute - inminute) + outsecond - insecond
hours 99 = (time / 3600) - .5
hours = 0 when hours < 0
time = time - hours * 3600
minutes 99 = time / 60 - .5
minutes = 0 when minutes < 0
seconds = time - minutes * 60
time = ' '
```

Assuming you enter a "time-in" as "inhour," "inminute," and "insecond," and a "time-out" as "outhour," "outminute," and "outsecond, this set of rules will calculate the elapsed time for up to 12 hours. This time is entered in the fields "hours," "minutes," and "seconds." The field "time" is used to hold intermediate results; you can set it to spaces after the calculation.

Registered T/Maker Consultants

To be registered as a consultant and listed in T/Mug, send us a copy of your system complete with explanation and documentation. If it is consistent in quality and degree of completeness with the systems we have previously accepted, we will register you as a consultant.

Dick Danielson, President
Strategic Advisory Consultants, Inc.
P.O. Box 137
Star Prairie, WI 54026
(715) 248-3434

Robert Payne, President
Business Computer Service, Inc.
1210 Smith Street
Charleston, WV 25301
(304) 343-9471

James L. Bullock
Bullock's Electronics
1016 West Street
Norton, KS 67654
(912) 877-3277

Ron Roizen
2011 Francisco Street
Berkeley, CA 94709
(415) 549-0103

Don Baack
6495 S. W. Burlingame
Portland, OR 97201
(503) 244-2741

Doug Beynon
Beynon Howes & Company
P.O. Box 944
Lawrence, KS 66044

James D. Gilbert
Gilbert's Data-Word Processing
237 Luff Lane
Warminster, PA 18974
(215) 443-9963

Robert D. Williamson, President
Can-American Electronics Corp.
7220 Taft Street
Hollywood, FL 33024
(305) 966-5588

Thaitronic Company Ltd.
P.O. Box 11-90 Phrakhanong
Bangkok, Thailand
Tel. 392-8307
TELEX Unitech Erawan TH 82189

Peter Channon, Frank McInnes
The Computer Factory
214 Harbord Road
Brookvale, NSW Australia
(02) 938-2522

Paul McAllister
Digital Concepts
45 Maple Street
Marlboro, Mass 01752
(617) 485-4491

Emil Widmer
Computertechnik fur Manager AG
Baarerstr. 45, Postfach 708
CH-6301 Zug, Switzerland
Tel. 042 21 0887

T/MUG BACK ISSUES

T/MUG back issues provide helpful information for those users who are just getting to know T/Maker and for seasoned users as well. The set of back issues is like a mini applications booklet in that it offers methods for performing relatively simple work such as setting up a checking account system, to more complicated tasks such as analyzing business performance. One of the most attractive features of the various applications found in the back issues is the versatility that they allow the user. Once the user becomes familiar with a particular method, he can customize it to fit his own individual needs.

The T/MUG back issue set includes issues dated from January 1982 to the present. They illustrate the history of T/Maker as well as provide valuable tips and tricks for using T/Maker more effectively. Following are some of the highlights offered in the set:

* 1040 Federal Tax Form * Checking Account System * Personalized Form Letters
* How To Do Multiple Calculations * Invoicing System * Opening an IRA
* Percent Change Calculations

T/MUG back issues are available for $25.00 for shipment to the U. S., Canada, and Mexico. Other countries please add $15.00 for air mail charges, and submit an international money order or U. S. bank check drawn in U. S. dollars. Make check out to T/Maker Company at 2115 Landings Drive, Mountain View, CA 94043.

- - - - - - - - - - - - - - - *clip here* - - - - - - - - - - - - - - -

Yes, I want to receive the entire set of back issues. Enclosed is my check for --

[ ] $25.00 *[ ] $40.00*

Please rush the set to my address below.

*Name*_____

*Company*_____

*Address*_____

*City*_____ *State*_____ *Zip*_____

SUBSCRIPTION/APPLICATION FORM

T/MAKER USERS GROUP
2115 LANDINGS DRIVE, MOUNTAIN VIEW, CA 94043

Members of the T/Maker Users Group receive a bimonthly newsletter providing examples of other members' experiences with T/Maker in the areas of text editing, financial modeling, personal and business accounting, mathematical and statistical applications, and many other fields of interest. The T/Maker newsletter will also provide answers and solutions to members' technical and nontechnical questions and problems relating to their use of T/Maker.

---

detach and mail

| | Enter | | **Renew my membership in the T/Maker Users Group and send me the bimonthly newsletter for one year.**

NAME:_____

COMPANY:_____

ADDRESS:_____

CITY:_____ STATE:_____ ZIP:_____

Please make check payable to T/Maker Users Group. Cost is $15.00 per year for US, Canada, and Mexico. Other countries please add $20 for air mail charges, and please submit an international money order or US bank check drawn in US dollars.

MY HARDWARE AND SOFTWARE CONFIGURATIONS ARE:

COMPUTER: MANUFACTURER:_____ MODEL:_____ MEMORY_____

TERMINAL: MANUFACTURER:_____ MODEL:_____

DISK FORMAT:_____

OPERATING SYSTEM:_____ VERSION:_____

T/MAKER VERSION:_____

T|MUG

T/MAKER RESEARCH CO.
812 Pollard Rd. Ste.8
Los Gatos, CA 95030

BULK RATE
U.S. POSTAGE
PAID
PALO ALTO, CA
PERMIT NO. 96