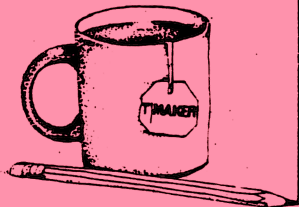


T MUG

T/MAKER USER'S GROUP NEWSLETTER
VOLUME 3, NUMBER 6, NOVEMBER/DECEMBER 1984

In This Issue:

T/Maker NewsFront	1
Printing Standard Labels	3
Importing Database Files	7
Attorney's Billing System	10
Using the Spreadsheet and Database Simultaneously.	13
Assigning Variable Equations	18
Job Hunting with T/Maker III	20
A Visicalc User's Guide to T/Maker.	23



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: T/Maker Users Group Newsletter

Copyright 1984 T/Maker Company. All rights reserved



T/Maker NewsFront

Radio Shack Road Show

-- Heidi Roizen

Tandy/Radio Shack, in a tremendous display of marketing savvy and organization, sponsored a series of Software Showcases in September and October. The shows, featuring 125 software companies displaying their wares on Tandy equipment, attracted both Radio Shack store personnel and end-users. Royal Farros put forth a heroic effort, not only orchestrating T/Maker's participation in the shows, but also doing booth duty ten hours per day for ten show days, from Houston to Atlanta to Los Angeles to New York to Chicago. And speaking as one who shared the booth with Royal during three of those five shows, he deserves a medal -- or at least a year of unlimited free podiatric services.

We want to thank the Tandy/Radio Shack people for such a terrific experience. Mark, Phil, LaDonna, Ed, Bob, Billy and all the rest of you . . . Thanks!

New T/Maker Version: TRS DOS

Due in part to the tremendous enthusiasm we received at the Tandy Showcases, Peter and Tom have just completed a TRS DOS version of T/Maker for the Radio Shack Model 4. This version should be available through Radio Shack by the end of the year.

On the Foreign Front

Nippon Univac has had many recent successes in the marketing of the Japanese version of the new package (with Database) and we hope to give you some of the details in the next issue.

The first Chinese version of T/Maker has been completed, and is now being beta-tested by co-producer Multitech of Taiwan. We expect to be seeing it right about now at the Comdex show in Las Vegas November 14 - 18.

The Compagnie Generale des Eaux (nation-wide water company of France) has purchased a 500-unit T/Maker license to make T/Maker the standard software package on their micros.

-- NewsFront, Continued

ClickArt News

The third in the ClickArt series was introduced in time for the Christmas season. This one, ClickArt:Letters, is a set of large size, quality type fonts for use on the Macintosh. Many of the headlines you see in this issue were done with ClickArt:Letters. The initial ClickArt product has made USA Today, Billboard magazine and the Softsel Hotlist as one of the top ten selling home software products in the US market.

T/MUG PRICES FOR SOFTWARE:

The following are prices for various T/Maker options that we often receive requests for. Orders can be placed through us and we'll take them over the phone for COD, VISA, or MasterCard payments. California residents add 6.5% sales tax. Shipping is \$6.50 for UPS blue and \$3.13 for UPS ground. And remember, we MUST know what kind of computer and operating system you have!

UPGRADE FROM T/MAKER II TO T/MAKER III MUST HAVE PROOF OF T/MAKER II SERIAL NUMBER	\$90.00
UPGRADE FROM T/MAKER III TO T/MAKER INTEGRATED SOFTWARE MUST HAVE PROOF OF T/MAKER III SERIAL NUMBER	\$175.00
T/MAKER III FOR SECOND OPERATING SYSTEM, NO DOCUMENTATION MUST HAVE PROOF OF T/MAKER III SERIAL NUMBER	\$137.50
T/MAKER INTEGRATED SOFTWARE FOR SECOND OPERATING SYSTEM NO DOCUMENTATION. MUST HAVE PROOF OF T/MAKER I.S. SERIAL NUMBER.	\$225.00



Printing Standard Labels

--Robert Simon

When creating different databases, you often use the same components; for example, name, address, and telephone. Therefore, it is useful to have a common base record structure from which you build each database.

This procedure is advantageous for a couple of reasons. First, the common record structure gives you a starting point, so you are not beginning from scratch each time. Another and more important reason, though, is that if you use the identical form and record definitions, you can use standard or pre-defined routines for printing or selecting records for typical purposes, such as form letters, invoices, summary reports, or labels.

For example, if you start with a standard record form (STAN.FRM) for the basic name and address data, you can use that as a starting point for your telephone list, order entry, and customer lists merely by adding the required fields.

Consider the following example in which we have also added a print field which determines whether or not the record gets selected to be printed.

STAN.FRM

<form>

Name:	{first	}	{last	}
Title:	{title	}		}
Address:	{address	}		}
City:	{city	}		}
State:	{state	}		}
Zip:	{zip	}		}
Phone1:	{phone1	}		}
Phone2:	{phone2	}		}
Print:	{prt}			

<end>

-- Standard Labels, Continued

```
<record>
a {first           } {last           }
b {title           }
c {address         }
d {city            }
e {state           }
f {zip             }
g {phone1          }
h {phone2          }
i {prt}
<end>
```

We have used the preceding STAN.FRM file for a Name, Address and Telephone database by renaming it to be PHONE.DAT.

PHONE.DAT

```
<form>
Telephone List
-----
```

```
Name:      {first           } {last           }
Title:     {title           }
Address:   {address         }
City:      {city            }
State:     {state           }
Zip:       {zip             }

Phone1:    {phone1          }
Phone2:    {phone2          }

Print:     {prt}
<end>
```

```
<record>
a {first           } {last           }
b {title           }
c {address         }
d {city            }
e {state           }
f {zip             }
g {phone1          }
h {phone2          }
i {prt}
<end>
```



-- Standard Labels, Continued

a Royal	Farros
b Director of Marketing, Grande	
c The B.M.I.	
d Paloeest of Alto	
e California	
f 99912	
g (415) 992-1212	
h (408) 423-4367	
i y	
a Mark	Jones
b LLB	
c 435 Broadway	
d Calamazoo	
e Washington	
f 95461	
g (703) 244-9292	
h	
i y	
a Bruce	Markem
b	
c 1028 57th Street	
d Spokane	
e Washington	
f 97112	
g (704) 929-8834	
h	
i n	

If we GET the STAN.LAB file and then issue the following T/Maker commands, those Name and Address records in the file PHONE.DAT which are marked to be printed, will be selected and printed. Notice that the print design command .pagelength is used to set the length of label.

The field formatting options used below commpress any blank spaces so that the last name immediately follows the end of the first name. Likewise for the state and zip.

-- Standard Labels, Continued

This is the STAN.LAB file:

```
SELECT Phone.dat WHEN (prt = y) END FIND ..start PRINT IT
```

Standard Label Print File

<record>

{first	!} {last	<}	
{title	<}		}}
{company	<}		}}
{street	<}		}}
{city	!}, {state	!} {zip}	}}

.newpage

<end>

..start

.pagelength 18

.indent 5

..<here>

Issuing the DO command would produce the following, ready for printing:

..start

.pagelength 18

.indent 5

..<here>

Royal Farros

Director of Marketing, Grande

The B.M.I.

Paloest of Alto, California 99912

.newpage

Mark Jones

LLB

435 Broadway

Calamazoo, Washington 95461

.newpage



Importing Database Files

-- Lawrence D. Haber

I work almost entirely with large statistical data files, derived from surveys and censuses. I do very little data entry and I avoid it whenever possible. I believe that anyone with an investment in an existing set of data files would of course prefer to transfer files rather than go through the laborious and expensive process of rekeying and rechecking data.

When I received the Integrated Software upgrade (which includes the database functions) to T/Maker III, one of the first things I looked for was a section on "importing" data files created outside the database system (sometimes referred to as "foreign" or "external" files) into a format for T/Maker's database features to understand. I was somewhat disappointed to find no reference to files transfer or importing capabilities in the new material.

Importing ASCII Files

Fortunately, T/Maker's database features work with the same files the rest of the T/Maker commands use; namely, straight text, or ASCII standard files. Therefore, an external file can be imported or transferred into T/Maker if the file is in text editor or ASCII (aka direct access) file format and if the file is small enough to fit in the T/Maker work area (less than 44KB in length on an MSDOS system.)

When both these conditions are true, importing a file is simple. You can enter T/Maker, and GET the file directly. All you need for the T/Maker database functions to understand this imported data file is a record definition at the top of the file. You can add the record definition directly through the editor, or create it in a separate file and use the INSERT command.

Importing Other Files

When the files are in Mailmerge (quotes and comma delimiters), DIF or other formats, file importing is more complicated. A few intervening steps are necessary and some precautions should be observed. There are also alternative ways of handling file size problems.

When I first started using T/Maker, I adapted my files to the T/Maker environment by using the text editor to transform the data, removing commas, adding spaces and aligning columns. This non-automated

-- Importing Database Files, Continued

method is allright if you have small data files, or are switching from one database to T/Maker and don't plan on going through the exercise more than once. But it is more than a mild nuisance if you have many large files or expect to do this more than once.

A Program Aid

I now use a flat file program, PC-FILE III, to convert my files to a text-editor direct access file. It can also be used to split the files into segments small enough to fit the T/Maker work area, although there are more direct ways to do this with T/Maker commands and MSDOS or other operating systems. PC-FILE is a clean and very helpful file manager program, relatively inexpensive and with good file "import" and "export" facilities.

As a first step, I import my BASIC or MailMerge formatted file into PC-FILE and export it as a text editor or direct access file. Next I write a record definition file in the T/Maker text editor. If the data file is under 44KB, you can then combine the two in the text editor by using the INSERT command to add the record definition to the top of the data file.

For Oversize Files

If the data file is too large for the working area, return to the operating system and use the COPY/APPEND command in MSDOS or other operating system to add the record definition to the top of the data file. With the record definition in place, the SELECT function can now be used to pick subsets of records from the complete file, either by sequence or with a conditional command, such as SELECT FILENAME FROM 1 TO 300 END or SELECT FILENAME WHEN FIELD < 250000 END.

Error Checking

The greatest strengths of the T/Maker Database are its ease of use and flexibility. Its weaknesses are its size limitations and the lack of error control. Errors in the spacing of the record definition brackets can cause disturbing changes in the distribution of data values among fields. This kind of error is unlikely to occur with highly structured database systems, but those systems also sacrifice the direct accessibility and flexibility of T/Maker. Fortunately, this type of error is easy to detect and correct, using the UPDATE command, a form definition and the SELECT command.

If you're using transferred files, you wouldn't have used the form definition for data entry. However, the form definition can be useful for error checking and correction if the form definition is ordered or spaced differently from the record definition. The UPDATE command will show if the values for each field are placed properly or have been misaligned. Using a "self-select" command (SELECT it END)



-- Importing Database Files, Continued

is also a good check that the record definition fits the data correctly.

The RENAME command is a great help in isolating and correcting errors. You can RENAME a suspect file and check it out without affecting your original file. Remember to wipe out either the old or new file when you're done; otherwise you wind up with a large number of strange files.

Summary

Though specific instructions may not be found in the manual, T/Maker can use imported or transferred files, though some intermediate steps may be required to adapt files which are too large to enter the work area, or which are not in a text editor or ASCII format. A program which can adapt formats (such as PC-FILE) is a big help. Error-checking the resulting file is a problem (always, everywhere) but the procedures given here can be a big help.

Though bringing other database files into T/Maker can be a challenge, the result is worthwhile. Importing your data allows you to exploit the flexibility, integration with word processing, spreadsheet, and bar-charting, and ease of use that T/Maker offers.

=====

SEND US YOUR APPLICATIONS!

T/Maker users, send us your applications! Every other month, you have been getting the benefits of other users' tips, trials and triumphs. By contributing your own, you get not only the fame associated with being a T/Maker master, but you will also receive one of our infamous trinkets, like the T/Maker Teamug or the Keychain.

We love any and all contributions, from simple to complex. And we like them even better if they come on disk, any format. We hope to be seeing you in print . . .

An Attorney's Billing System

-- Kenneth K. Watts, P. S.

As an attorney, I had been using Lotus 123 for such things as real estate closing statements, billing statements and trust account statements to clients. I was using 123 because of the ease in entering the formulas for two-column arithmetic. However, I soon missed the zv function of T/Maker which I could not duplicate in 123. I also missed the word processing capability of T/Maker. But the sorest point was that I could not make row deletions or insertions on my 123 statements without having to recopy formulas to affected cells in the row. This made it very difficult to make changes.

I decided I better study T/Maker some more to see if I could convert my two-column statements to it. I found that the fetch and store functions and a third column were necessary to do the calculations I wanted, but I did not want this other column to appear on my statements.

My statements are 80 columns wide. The solution was to use the table mode in T/Maker and place the third column beyond the 87th column position when editing on the screen. A comma placed in front of the example number above the third column makes the results there unprintable.

Below are my sample billing statements and trust account statements to illustrate this technique. They are saved, of course, as generics without the sample data. Although not shown in these samples, I have also incorporated top and bottom routines for statements which would extend for more than one page.

The jc1 row stores the result of charges less payments in the third column. The jc2 line then fetches this result for printing under the charge column, the second column for the calculation.

The uc1 and uc2 rows multiply the hours entered by the hourly rate, in this case \$60 per hour, and print the result in the charges column. If one desires to enter a flat fee or cost amount in the charge column, then sc must be entered in the vertical calculation strip just above the row for such an entry in order to suspend the hourly calculation for that row. Otherwise, the amount entered would be erased since the uc formulas would multiply 0 hours by 60 and print the result of 0, a space due to the zv row, in the charge column.

This form minus the data is saved on disk as BILL.FMT. To start a new bill, the commands from the WHAT NEXT? prompt are GET BILL.FMT RENAME FILENAME.STM SAVE EDIT. These can be saved in a disk file to DO in order to save retyping. Substitute a name up to eight characters long for FILENAME for each new bill created.

-- Billing System, Continued



The trust account statement is attached to the billing statement when there is trust account activity to report. On disk this file is saved, minus the sample data, under the name TRUST.FMT. When editing the client's bill, INSERT the TRUST.FMT just below the last line of the bill. The .new design command causes the trust account statement to be printed on a separate page.

Row insertions and deletions to correct mistakes or make revisions on either of these statements can easily be done without affecting the formulas.

At the end of the month, the statements and duplicates for your records can be printed automatically. Create a disk file named STMPRN and for each statement created add a line to this file as follows:

PRINT NONSTOP FILENAME.STM PRINT NONSTOP FILENAME STM DO

When you are ready to print, type at the WHAT NEXT? prompt, GET STMPRN DO

The TRUST.FMT insert for DOEOCT.STM

```
.new
.4
ex          9,999,999.99  9,999,999.99  ,9,999,999.99
TRUST ACCOUNT STATEMENT OF KENNETH K. WATTS, P.S. FOR JOHN DOE
```

The following is a statement of funds received and disbursed through the trust account. This is not a bill. It is provided for your information.

			TRUST FUNDS	
Date	Name	Description	Received	Disbursed
Sep 84				
+	09	John Doe	delinquent Smith contract payments	1,000.00
+	21	Clerk of Court	deposit delinquent payments	1,000.00
jc3			+	-
=			1,000.00	1,000.00
jc4			ftb	
zv			NONE	
=		REMAINING FUNDS IN TRUST: \$	NONE	

=====

-- Billing System, Continued

The File DOEOCT.STM

.clean

.4

KENNETH K. WATTS, P.S.
Attorney at Law
3312 West 17th
Spokane, Washington 99204

Telephone: (509) 456-8882

STATEMENT

October 1, 1984

To: John Doe
3314 E. Main St.
Anytown, Washington 99201

ex	999.99	9,999,999.99	9,999,999.99	,9,999,999.99
zy				
uc1	+		=	
uc2t			60	

			FEES AND COSTS	
Date	Description	Hours	Payments	Charges
Sep 84				
+	09 Conference with client & Mr. Thompson	1.00		60.00
+	09 retainer from client		250.00	
+	12 letter to Smith Corporation	0.50		30.00
+	19 prep. summons & complaint	2.00		120.00
sc				
+	21 filing fee			70.00
sc				
+	24 process service			35.00
+	25 phone confs. with Mr. Smith & client	1.00		60.00
+	26 prep. & filing of release & order	1.00		60.00
	dismissing complaint			

jc1			-	+	=sta
=	Rate: \$60/hr.	Totals:	5.50	250.00	435.00

jc2		fta
=	AMOUNT DUE:	\$ 185.00
		=====

Using the Spreadsheet and Database Simultaneously

— Heidi Roizen

The addition of the database functions allows you to perform mathematical calculations without the structure of the spreadsheet. The database's calculation function, as opposed to the spreadsheet's row and column orientation, represents mathematical relationships as formulae.

For example, a spreadsheet that figures out the total of retail and dealer sales might look like this:

example	999	999	9,999
aci	+	+	=
	RETAIL	DEALER	TOTAL
+	111	222	333

While a database might present the same problem like this:

```
<rules>
TOTAL 9,999 = RETAIL + DEALER
<end>
```

```
<record>
RETAIL: {RETAIL>}
DEALER: {DEALER>}
TOTAL: {TOTAL >}
<end>
RETAIL: 111
DEALER: 222
TOTAL: 999
```

Each method of representing mathematical relationships has its advantages.

The spreadsheet method is excellent for straightforward calculations, like adding many rows or columns of data. It is also useful for more complex situations which require the use of the built-in features like the NET PRESENT VALUE, PROJECTION, or GROWTH RATE calculations. Also, numeric data must be in spreadsheet form in order to take advantage of the bar chart command.

The database's method, on the other hand, has certain advantages. Some complex relationships are more easily expressed in formula format, like this:

-- Database/Spreadsheet, Continued

TOTAL = (PRICE * QUANTITY) * (1 + TAX.RATE) + SHIPPING + COD.CHARGE

Also, the database allows for logical relationships, like

TAX.RATE = .065 WHEN STATE = "CA"

The database format has other advantages, like an easier method for entering data, and sorting by naming the field rather than the column positions, as in the spreadsheet.

Luckily, with T/Maker, you don't have to choose between the database and the spreadsheet -- because you can set your problem up as a spreadsheet and a database simultaneously. Here's an example of one application using this technique.

TEMPERATURE STATISTICS

Let's say we are doing a report on temperatures in the Bay Area during the midday hours. We will ask friends in a number of cities to call in at 11:00, 12:00, 1:00 and 2:00 with temperatures. Unfortunately, some of those friends use centigrade scales, while others use Farenheit. Of course, the temperatures must all be reported in the same manner in order for the averages to have any meaning.

By setting this up as a database, we can accomplish this through a set of rules. The fields we will need are CITY, indicator of scale, whether Farenheit or centigrade (I called it "S" for scale) the temperature readings (TMP11, TMP12, TMP13, TMP14) and the average, AVG.

We want T/Maker to convert the Farenheit temperatures to centigrade, but leave the centigrade temperatures as they are. Here are the rules to do this:

<RULES>

TMP11 999.9 = (TMP11 - 32) * (5 / 9) WHEN S = "F"

TMP12 999.9 = (TMP12 - 32) * (5 / 9) WHEN S = "F"

TMP13 999.9 = (TMP13 - 32) * (5 / 9) WHEN S = "F"

TMP14 999.9 = (TMP14 - 32) * (5 / 9) WHEN S = "F"

S = "C"

<END>

The first four rules reset the temperatures to centigrade when the scale indicator is set to F. The final rule resets the scale indicator to C, as all entries have been converted to centigrade by then.



-- Database/Spreadsheet, Continued

Also, a form definition has been added to make the task of entering data easier.

Here's the entire file, prior to any data being inserted:

<FORM>

BAY AREA TEMPERATURE STATISTICS

CITY: {CITY< > }

CENTIGRADE (C) OR FARENHEIT (F)? : {##S}

TEMPERATURE AT 11:00: {TMP11>}

TEMPERATURE AT 12:00: {TMP12>}

TEMPERATURE AT 13:00: {TMP13>}

TEMPERATURE AT 14:00: {TMP14>}

<END>

<RULES>

TMP11 999.9 = (TMP11 - 32) * (5 / 9) WHEN S = "F"

TMP12 999.9 = (TMP12 - 32) * (5 / 9) WHEN S = "F"

TMP13 999.9 = (TMP13 - 32) * (5 / 9) WHEN S = "F"

TMP14 999.9 = (TMP14 - 32) * (5 / 9) WHEN S = "F"

S = "C"

<END>

<RECORD>

+ {##S} {CITY< > } {TMP11>}{TMP12>}{TMP13>}{TMP14>}{AVG >}

<END>

.NEW

.CLEAN

EXAMPLE

UC1

999.9 999.9 999.9 999.9 999.9
+ + + + AVR

BAY AREA TEMPERATURE STATISTICS

SCALE CITY 11:00 12:00 13:00 14:00 AVERAGE

.. <HERE>

AVR

AVERAGES

-- Database/Spreadsheet, Continued

Now let's add some data through UPDATE. The data portion, before any functions have been performed, now looks like this:

EXAMPLE			999.9	999.9	999.9	999.9	999.9
UC1			+	+	+	+	AVR

BAY AREA TEMPERATURE STATISTICS							

SCALE	CITY		11:00	12:00	13:00	14:00	AVERAGE

.. <HERE>							
+	F	SAN FRANCISCO	64	66	67	67	
+	F	SAN JOSE	76	78	78	79	
+	F	MOUNTAIN VIEW	72	74	75	76	
+	C	MARIN	18	19	20	20	
+	C	BERKELEY	16	17	17	18	

AVR	AVERAGES						

After invoking the SET command at the WHAT NEXT? prompt, the data portion will look like this:

EXAMPLE			999.9	999.9	999.9	999.9	999.9
UC1			+	+	+	+	AVR

BAY AREA TEMPERATURE STATISTICS							

SCALE	CITY		11:00	12:00	13:00	14:00	AVERAGE

.. <HERE>							
+	C	SAN FRANCISCO	17.8	18.9	19.4	19.4	
+	C	SAN JOSE	24.4	25.6	25.6	26.1	
+	C	MOUNTAIN VIEW	22.2	23.3	23.9	24.4	
+	C	MARIN	18	19	20	20	
+	C	BERKELEY	16	17	17	18	

AVR	AVERAGES						

We still need to invoke COMPUTE to get the averages by city and by time. After COMPUTE and PRINT IT, we would get this:



-- Database/Spreadsheet, Continued

BAY AREA TEMPERATURE STATISTICS

SCALE	CITY	11:00	12:00	13:00	14:00	AVERAGE
C	SAN FRANCISCO	17.8	18.9	19.4	19.4	18.9
C	SAN JOSE	24.4	25.6	25.6	26.1	25.4
C	MOUNTAIN VIEW	22.2	23.3	23.9	24.4	23.5
C	MARIN	18.0	19.0	20.0	20.0	19.3
C	BERKELEY	16.0	17.0	17.0	18.0	17.0

AVERAGES	19.7	20.8	21.2	21.6	20.8
----------	------	------	------	------	------

Of course, any column or row could also be represented as a bar chart.

Another Way to Place Lists Side by Side

-- Hal Bayman

In the July/August issue there was an article about placing addresses side-by-side for printing purposes. I have had cause to do this same thing with T/Maker, but as my method was somewhat different, I thought I would share it with the T/MUG members.

After saving my list, I begin by deciding where I would like to break my list and then position the file on the screen with the top line of my indented second column on the top line of the screen. I then execute CLIP BEFORE to remove the top portion of the complete list. Next, I RENAME LIST2 SAVE. I now have my second column saved and can GET LIST (the original file).

I position LIST on the line just before the line I used as the top line for the second column. Next I issue the command CLIP AFTER to delete that portion that I have set aside (in LIST2) to become column 2. I now position the file to line 1 and use MERGE LIST2 and a position parameter to move the second column into position in the working area. The steps for the sample list of four names that were included in the July/August issue then would be:

```
11 CLIP BEFORE RENAME LIST2 SAVE GET LIST 10 CLIP AFTER 1 MERGE LIST2
36
```

A Database/Spreadsheet Approach to Assigning Variable Equations

-- Peter Roizen

Editor's Note: When T/Mug time came around again this month, I asked Peter to do an article. I told him I was writing an article on mixing the database and spreadsheet to perform complex operations which would not be possible using only one or the other. (My article, which calculates a temperature chart, appears elsewhere in this issue.)

Peter thought the concept was a good idea, but that my article was too simplistic. So here, for the benefit of those of you who can understand it, is Peter's attack on the same theme. This one's rated "X" -- FOR EXPERTS ONLY!

Here is another example of mixing data base and spreadsheet commands. Consider the problem of a spreadsheet in which all the numbers in each row are calculated according to one of a number of formulae. Such a spreadsheet could be constructed by placing the appropriate JUST COMPUTE lines ahead of the row to which they apply. For each new row, we would have to enter or copy in the appropriate equations and, of course, the row itself.

If we decided at some later point to change the equations, we would have to replace the JC lines with new ones. Suppose, however, that there were only a limited number of formulae in use. We might ask, isn't there any way to get the equations put in automatically?

Of course, the answer is yes -- otherwise, we would not have asked the question. In the example that follows, a <RECORD> Definition is used which includes two equations. And, a set of <RULES> has been defined which loads the equations with the correct symbols based on a field which is entered in the row. A <FORM> has also been put in so that the UPDATE command can be used to revise or enter rows.

Because of the <RULES>, the SET command can be used ahead of the COMPUTE command to put in the appropriate JC equations. Let's try it.

```
<FORM>
      Title      Type      VALUE 1 VALUE 2 VALUE 3 VALUE 4 VALUE 5
+      (title ) (type ) { val1} {      } {      } {      } {      }
<END>
```


-- DB/SS for Variable Equations, Continued

T/MUG

```

999,999 999,999 999,999 999,999 999,999
<rules>
eq1 = ' '
eq2 = ' '
when type = CUM
    eq1 = "+      =+      =+      =+      ="
    eq2 = "cum  +      +      +      +      +"
when type = SAME
    eq1 = "sta      fta      fta      fta      fta"
when type = LOG
    eq1 = "+      log+     log+     log+     log"
<end>
<RECORD>
JC1      { eq1      } }}
JC2      { eq2      } }}
+        {title } {type } {>val1} {      } {      } {      } {      } }}
<END>

```

```

-----
Title      Type      VALUE 1 VALUE 2 VALUE 3 VALUE 4 VALUE 5
-----
ex          999,999 999,999 999,999 999,999 999,999
.. <here>
+   SALES      LOG      120,000
+   GARBAGE     CUM       111
+   FUTURE     SAME      12
-----
=   TOTAL
-----

```

Below is how the spreadsheet portion of the file would look after the SET command. You can see the appropriate lines have been added. Simply, changing the "Type" and reusing the SET command will change the equations. Of course, COMPUTE must still be used to calculate the values.

```

-----
Title      Type      VALUE 1 VALUE 2 VALUE 3 VALUE 4 VALUE 5
-----
ex          999,999 999,999 999,999 999,999 999,999
.. <here>
JC1          +      log+     log+     log+     log
+   SALES      LOG      120,000
JC1          +      =+      =+      =+      =
JC2          cum  +      +      +      +
+   GARBAGE     CUM       111
JC1          sta      fta      fta      fta      fta
+   FUTURE     SAME      12
-----
=   TOTAL
-----

```

Job Hunting with T/Maker III

-- Hal Bayman

I found myself recently looking for employment several states away. Obviously it's great having word processing available for this sort of task, but mere word processing pales by comparison to what can be done with T/Maker and a little imagination.

I realized that I would be sending basically the same cover letter with a copy of my resume to scores of companies. It occurred to me that with T/Maker I could automate not only the generation of this generic cover letter, but the printing of envelopes, and the maintenance of a contact record. I accomplished this by the creation of four basic files, which I named as below:

ADDRESS - this is the file that contact information is entered into and accumulated so that this becomes the record file.

AUTOLETR - this is the cover letter file. It is a mask with the appropriate zones for LOAD to move data from ADDRESS into.

AUTOLOPE - this is the envelope file. It is a mask with the appropriate zones for LOADING data from ADDRESS.

TEMPLATE - this is the data input template that is INSERTed on ADDRESS at the end of printing.

To print several letters all that is required is to sit down at the terminal with the want ads or other lists of company names that you wish to contact. Load T/Maker and GET ADDRESS. At the top of ADDRESS is a blank data template ready to be filled with the appropriate information. Since some addresses are fewer lines long than the template, just leave any unused lines off the top of the stack so there will be no blank lines in the printed result. The additional lines associated with Status are for entering notes for your records -- acknowledgments, rejections, dates of phone contacts, etc. Once you've got a long list of contacts it's great to be able to use FIND to locate the contact file on a company and see just how long they've been ignoring you!

After entering the information, quit the editor, type DO at the WHAT NEXT? prompt, and T/Maker will LOAD the data and halt at the Printer's Prompt so you can proof the data (and be sure that you have some of that nice 25% rag, water-marked bond in the printer). Type "Y" to start to print. At the end of printing T/Maker will LOAD the data into AUTOLOPE and halt at the Printer's Prompt as before. If all is well and you've got that expensive envelope in the printer, type "Y" and T/Maker will print your "custom" envelope. After

-- Job Hunting, Continued



printing the envelope T/Maker GETs ADDRESS, INSERTs TEMPLATE, SAVES ADDRESS and obediently waits for you to enter the next data set or STOP. (Obviously our old friend DO is a key factor in all this hocus pocus.)

Here are samples of the four files I use:

FILE: ADDRESS

```
save get autoletr do
line1      = "
line2      = "
line3      = "
address    = "
city,st    = "
zip        = "
dear       = "
date       = "
status     -
line1      = "Tex Hanson"
line2      = "BIG STATE ENGINEERING"
line3      = "Ste. 1275 E"
address    = "6350 LBJ Freeway"
city,st    = "Dallas, TX"
zip        = "75240"
dear       = "Mr. Hanson"
date       = "15 OCT 84"
status     -
```

FILE: AUTOLETR

```
load address print it get autoloop do
```

```
Dallas, TX
(>date  )
```

```
{! line1      }
{! line2      }
{! line3      }
{! address    }
{! city,st    }  {! zip      }
```

```
Dear {!dear    },
```

Your organization, like every other, needs problem solvers, people who can synthesize unexpected solutions from the commonplace. You need generalists with a broad base of experience and interests who can bring . . . (and so on)

-- Job Hunting, Continued

FILE: AUTOLOPE

load address print it get address insert template save

Newton
2552 Republic Drive
Dallas, TX 75229

{1 line1	}
{1 line2	}
{1 line3	}
{1 address	}
{1 city,st	} {1 zip }

FILE: TEMPLATE

save get autoletr do	
line1	= "
line2	= "
line3	= "
address	= "
city,st	= "
zip	= "
dear	= "
date	= "
status	-



Understanding T/Maker's Spreadsheet from a Visicalc Point of View

-- Royal Farros

Editor's Note: Some people seem to love T/Maker's spreadsheet for its free-form approach. This approach is different, however, from the traditional "cell" based approach.

After fielding a number of support calls and explaining T/Maker's spreadsheet at numerous trade shows, help-file author extraordinaire Royal Farros has developed an explanation he feels will help with the transition of you VisiCalc users to the T/Maker way. In this first article, Royal explains how to label the columns, and how to think of and use row equations. Future installments will follow in subsequent T/MUG issues.

Setting Up T/Maker To Look Like Visicalc

The Example Line:

T/Maker does "Calculation by Example" - that is, you give T/Maker an example of where it will find numbers and how you want them to look. You do this by creating an EXAMPLE LINE containing MODEL NUMBERS.

Instead of starting this EXAMPLE LINE with a cryptic "ex" in the first two columns, why not type out the entire word "Example" as shown below?

Notice that the word "Example" contains 7 letters. Remember that T/Maker allows you to put column equations in the first 7 column positions. By starting an EXAMPLE LINE with the full word "Example", we are forming a VISUAL RULER displaying the area under which we can legally enter math symbols.

Following this non-cryptic approach, why not label our columns with LETTERS instead of the "999"s the T/Maker documentation suggests? MODEL NUMBERS made up of LETTERS will help us distinguish particular columns of numbers much more easily (and look a bit more like the traditional VisiCalc approach to column.)

-- A Visicalc Approach to T/Maker, Continued

Here's an example of what I mean:

EXAMPLE	aaa	bbb	ccc, ccc. cc	dd	ee, eee
+	2	3	4.00	5	6
*	2	2	2.00	2	2
=	4	6	8.00	10	12

The BLANK EXAMPLE LINE:

Once you've begun a T/Maker spreadsheet, T/Maker doesn't like you to enter anything but math symbols in columns 1 through 7 throughout the remainder of the file. There is, however, a way to trick T/Maker into letting you use the first 7 columns for text again after ending a spreadsheet.

The trick is to always end every spreadsheet with a BLANK EXAMPLE LINE. (That is, end with an EXAMPLE LINE containing NO MODEL NUMBERS.) The BLANK EXAMPLE LINE says to T/Maker, in effect, I now have a spreadsheet with no numbers -- therefore, compute does not need to worry about what goes on in this part of the spreadsheet. Since this can never hurt you, it is a good idea to form the habit of ending every spreadsheet with a blank example line.

EXAMPLE	aaa	bbb	ccc
+	2	3	4
+	2	3	4
=	4	6	8

EXAMPLE

See! we have no problems putting text under a BLANK EXAMPLE LINE (like this sentence right here.)



-- A Visicalc Approach to T/Maker, Continued

Back to Basics: The JC Row Code:

If you are having trouble keeping up with AC's, UC's and RC's, here's a good tip for doing ROW EQUATIONS (horizontal calculations).

To aid you in working with this method of performing row equations, I want you to forget everything that you know about horizontal row codes. We're going to start from scratch.

From now on (or until you feel comfortable doing otherwise) you should put all relevant equations right above each row of numbers.

In other words, the only row code you need to use for any math work going across a table is the JUST COMPUTE Row Code. The JC Row Code will only operate on the very next row of data it encounters.

That means that every row of numbers on which you want to perform math must have its own JC Row Code (or Codes) right above that row.

Notice that

Row Equations: Always begin with a Row Code in column 1 (e.g., the jc in "jcl" below)

Always have a Row Code NUMBER (e.g., the 1 in "jcl")

Contain the appropriate Math Sign ABOVE the number it will work on.

Example	aaa	bbb	ccc

jcl	+		=
+	10		10

jcl	+	+	=
+	10	5	15

jcl	+	-	=
+	10	5	5

jcl	+	*	=
+	10	5	50

jcl	+	/	=
+	10	5	2

Example

-- A Visicalc Approach to T/Maker, Continued

ROW CODE NUMBERS:

Up to 30 Equations can operate on a SINGLE row of numbers. T/Maker stores an equation according to it's ROW CODE NUMBER.

At a row of entered data, T/Maker will get the equation stored in the FIRST ROW CODE NUMBER location (jc1) and perform that calculation on the row of data.

T/Maker then looks at the SECOND ROW CODE NUMBER (jc2) location and performs that equation on the row of data, and so on.

Note in the example below that the row code number is important, not the placement. That is, even though the second example has the jc2 line above the jc1 line, the jc1 is performed first.

Also note that each equation to be performed on the row MUST have a unique row number. Reusing a number will cancel the previously stored equation. (See third part of below example.)

Example	aaa	bbb	ccc	ddd	eee	

jc1	+	+	=			<-- our standard case
jc2			+	-	=	
+	10	5	15	12	3	

jc2			+	-	=	<-- jc2 above jc1 makes no difference
jc1	+	+	=			
+	10	5	15	12	3	

jc1	+	+	=			<-- does not work with the same row code numbers
jc1			+	-	=	
+	10	5	0	12	-12	

Example



-- A Visicalc Approach to T/Maker, Continued

Multiple Row Equations: A Short Cut:

Equations that operate on numbers from LEFT to RIGHT can be placed on the same ROW CODE LINE.

For example, the table below shows two different ways of performing the same calculation; that is, the second JC1 performs the same math on the data row as the first series of JC'S (jc1, jc2, jc3).

Example	aaa	bbb	ccc	ddd	eee	fff	ggg

jc1	+	+	=				
jc2			+	-	=		
jc3					+	*	=
+	10	5	15	12	3	2	6

jc1	+	+	=+	-	=+	*	=
+	10	5	15	12	3	2	6

Example

Homework (not graded):

Until next time, you should play around with the JC Row Code. In particular, try **STORING & FETCHING** numbers with this JC approach. Understanding how T/Maker operates on each row is crucial to understanding how T/Maker operates on larger spreadsheet tasks.

Remember - everything you'll ever need to do horizontally can be accomplished with the Just Compute Row Code - (every row of numbers has a row of formulas above it).

All the other codes are just "JC Short Cuts" - a topic we'll discuss further in the next T/MUG issue.

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: _____

TMUC

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