T/MUG

T/MAKER USER'S GROUP NEWSLETTER
VOLUME 4, NUMBER 6, NOVEMBER/DECEMBER 1985

In This Issue:

Appointment Schedule 2

Bar Charting Program 4

Bibliography Using Database 8

Portable Rules for Multi-File Databases 17

Spacing Following Punctuation 19

T/Maker Pop-Up Calculator 20

Finding Records in Linked Databases 23

T/MUG 1985 INDEX 25

T/Maker Users' Group Newsletter

T/MAKER NEWSFRONT

-- Heidi Roizen

When You Upgrade -- REALLY Upgrade!

As most of you know, we are working on a new product, very similar to T/Maker but which takes full advantage of the greater memory, color, function keys, and other features of the PC-compatibles. However, we came from the CP/M world and realize that many of our best customers still operate with CP/M machines.

So here's our idea. We'll make a deal with one of the PC-clone vendors to provide machines with our upgrade. In a sense, we've come full circle from when hardware manufacturers bundled software with their machines. We'd bundle the hardware.

Our new product will sell for $495. As a current T/Maker customer, you could get the upgrade for $195 if you owned a PC. Or, you could buy the upgrade-bundle for $995. That would include a 256K PC-compatible, with 2 disk drives and a monochrome monitor, along with the new software product. The only thing you'd have to buy is DOS (2.0 or greater).

We're trying to decide on this promotion right now. So, if you could, please drop us a postcard saying whether you'd consider an offer like this. We'll announce our policy in the next issue. Hopefully, this unique offer will be a great way to upgrade your current system or get that always-wanted second system.

T/Mug List Destroyed

Well, it finally happened. One of our hard disks bombed. Of course, we always back up all our hard disks monthly -- except that particular one, which hadn't been backed up for six months. We have since re-generated the list from the paper order forms. So, if your subscription is incorrect in any way, please let us know. We'd appreciate a photocopy of your cancelled check or credit card receipt if you feel you have been short-changed.

What is Spare 1?

A number of you more avid T/Maker users have written or called us over time to uncover the use of Spare 1 in T/Modify. One writer suggested we publish the answer in T/Mug. So here it is: Spare 1 is precisely that -- a spare spot Peter left himself so he wouldn't have to revise T/Modify for specific version ideosyncracies.

For example, in the Japanese version, it is used to specify the character used to designate Japanese (kanji) alignment. In the Radio Shack Model 4 version, it changes the printer initialization. And, in certain Apple versions, it changes the default carriage return mode -- now don't ask me for a copy of that!

T/Maker Teeshirts Available

I have finally found a way to get articles. The only way you can get a T/Maker teeshirt is to contribute an article, both in print and on disk, to T/MUG. They are lovely teeshirts, and are very rare! Any good application gladly accepted.

Appointment Schedule

-- by Kenneth K. Watts

Tired of entering the date and time for every appointment you make? If you use a one-line entry system like that described on page 12 of T/MUG March 1983, that is what you must do for each appointment.

```
mmdd  time      description
----  --------  ------------------------------------------------
0101            NEW YEARS DAY
0102  8:30 am   meet-John Smith
0102  9:00 am   court-for Smith Motion
0103  8:00 am   office-staff meeting
0103  9:00 am   office-John Johnson
0103  9:30 am   office-Johnson deposition
0104            SATURDAY
```

The advantage of such a system is that appointments may be entered randomly, and then put into order by the sort command.

Well, here is another simple system that some users may find useful. The dates and times do not need to be entered for each appointment. You just move the cursor, tab and screen movement commands to get to the appropriate date and time intersection and enter the appointment.

```
 1986    8:00 a.m.      8:30 a.m.      9:00 a.m.      9:30 a.m.     10:00 a
      ////////////////////////////////////////////////////////////////////
JAN 01/  NEW YEARS DAY
JAN 02/                  M-JOHN SMITH   CT-SMITH MOTION--------------
JAN 03/  O-STAFF MEETING-------------- O-JOHN JOHNSON O-JOHSON DEPOSITION--
JAN 04/  SATURDAY
JAN 05/
JAN 06/
JAN 07/
JAN 08/
JAN 09/
JAN 10/
```

First enter T/Maker and create a format file named SCHEDULE.FMT. Use your change mode keystroke until you are in the Table mode. Then enter the times of the day that cover

-- continued from page 2

your typical work day as in the above example, except go as far to the right as you need. I use the CP/M version of T/Maker and have enough room to go to 5:00 PM in half hour intervals giving 15 spaces for each time interval. Then on the left, enter the year and below that as many months and dates as you want. I can put a whole year of dates in one file.

This task can be made easier by the use of the buffer, the REPLACE command, and the keystroke macro. First enter the dates for January, then go to 01 and copy to the buffer each date line through JAN 31. Then go to the next blank line and insert the buffer. Move up to the second JAN 01 that was entered when you inserted the buffer. Now use the REPLACE command to replace the JAN string with the FEB string. Delete the excess date lines for February, then repeat the process for each month through December.

In TModify I have instructed T/Maker to save tabs. If you have done this also, then go back to the first line and after clearing all tabs, set a tab every 15th space starting at the : in 8:00 AM. (You can do this easily by using Keystroke Macro to record one sequence of fifteen right movements and one set tab keystroke.)

By saving this file you will have a format that can be used for each year simply by changing the year entry and checking February to add or delete the date lines needed at the end of February. Then for the current year, rename the file SCHEDULE.85 and save it.

When you call up SCHEDULE.85, make sure you enter the table mode of T/Maker so you can freely move about the schedule. Then place your cursor on the space right after JAN 01 and type the command ESC F to enter the frame mode. This is indicated by the / symbols in the above example. This will allow you to move about your schedule and always see the date and time for any particular entry.

To save space you can develop a set of abbreviations for common words like office-O or court-CT and use these when filling in your schedule. You can block out the remaining time you anticipate for a matter by filling it in with dashes as in the above example. You can use the FIND command to find anything you want, such as all the appointments for JOHNSON or the JAN 04 date line. You can also use the KEEP command to collect all occurrences of a certain individual's name, etc.

If you are running out of memory for the file (check this using the INFO command), you can split the file into two or more files as needed covering portions of the year. You do this using the CLIP and RENAME commands making sure the top line of each file is the year and time line.

As you can see, you can move about this type of file using cursor, tab and screen movement commands. You need not type the date unless you choose to in response to a FIND command.

A BAR-CHARTING PROGRAM

-- Chris Shaw

Bar charts are sometimes nice to have in your documents, and T/Maker lets you define them in a variety of ways -- all horizontal! But I can never remember the commands needed to turn an array of numbers into a bar chart. So, naturally, I wrote a program to do this for me.

The name of my program is MAKEBARS. To use it, all I have to do is enter the command:

```
INSERT MAKEBARS DO
```

at the WHAT NEXT? prompt, and wait a while, as my specially identified lists of numbers are automatically converted into bar charts and inserted into my document at the appropriate places.

Here's how I type a list of numbers into my document so MAKEBARS will know it's supposed to make it into a bar chart:

```
.chart single do
.bar     first          23.5
.bar     second         37.1
.bar     third          18.0
.bar     fourth         28.6
.bar     last           33
.bar     ::             50
.end
```

I can also generate a double bar chart, which I enter as follows:

```
.chart double do
.bar     uno            23.5        27.2
.bar     dos            37.1        40.8
.bar     tres           18.0        41.3
.bar     quatro         28.6        44.1
.bar     ultimo         33          50
.bar     ::             50
.end
```

In both examples, the bar with the label :: eventually gets dropped, and serves only to provide a maximum value for scaling the chart.

At the end of my document, or anywhere after the last chart, I must include the following line,

```
.chart last do
```

which tells MAKEBARS there are no more charts to generate, and allows it to wrap things up gracefully.

One final point about using MAKEBARS before I show you how it works: At the end of the process, the document in my working file is renamed REPORT -- that's so I don't have to tell MAKEBARS its real name -- so I ordinarily rename it before saving it away.

Here's what MAKEBARS looks like:

```
replace .chart 'iNfO insert' do
replace .bar '+ ::'          do
find iNfO                    do
```

It doesn't look like it does much, but it's enough to get things started.

The first command replaces the .chart keyword with a pair of T/Maker commands: info, and insert. The strangely capitalized info command serves as an innocuous place marker -- one that can be executed without causing any trouble. The capitalization serves to make it highly unlikely that it will occur naturally in a document.

The second command is mainly cosmetic. It serves to identify lines containing numbers to be bar-charted to the T/Maker bar command.

The last command finds the first of the previously-generated iNfO commands and executes it. This command will now look like this:

```
iNfO insert single do    (or)    iNfO insert double do
```

As you may have guessed, the actual bar-charting work is done by the two programs named SINGLE and DOUBLE. I can define similar bar-charting programs to generate any kind of bar chart that T/Maker itself can manage.

Here's a listing of SINGLE:

```
        find delete do
        .chart
        >><<
*       delete report rename report save
*       delete chart rename chart
*       find .end clip after
*       1 find .chart
*       bar qqqqqqq char = names 11 width 1 space 0 replace end
*       save
*       get report
*       find .chart
*       replace .chart .block
*       find ::
*       insert chart
*       find iNfO do
                            ::      qqqqqqq
        ex ::                       9999999
        <<<
```

Those commands identified with * in the left margin *must* all be on one line in the program if things are to work properly. The bar command, which ends up somewhere left of the middle in this one line, does the actual bar charting. To generate a different kind of bar chart, it would need to be modified, as shown in DOUBLE, listed next.

```
        find delete do
        .chart
        >><<
*       delete report rename report save
*       delete chart rename chart
*       clip before
*       find .end clip after
*       1 find .chart
*       bar qqqqqqq char = names 11 width 1 space 0 numbers insert
*       bar ppppppp char - insert cut end
*       sort 1 3
*       arrange 6 300 end
*       save
*       get report
*       find .chart
*       replace .chart .block
*       find ::
*       insert chart
*       find iNfO do
                            ::      qqqqqqq     ppppppp
        ex ::                       9999999     9999999
        <<<
```

Again, the commands identified with * must all be on one long line.

Finally, here is the wrap-up program, LAST.

```
1 drop :: save edit
```

It deletes all remaining traces of the bar-charting programs from the working file, which has been renamed REPORT, and saves it away.

Oh yes. Here are the results of using MAKEBARS on the two examples given earlier:

```
>><<
                    <0 qqqqqqq =                                        50>

first         23.5  ============================
second        37.1  ===========================================
third         18.0  ====================
fourth        28.6  ================================
last            33  ====================================
<<<


>><<
                    <0 ppppppp -                                        50>

                    <0 qqqqqqq =                                        50>

uno           27.2  ---------------------------
uno           23.5  ========================
dos           40.8  ----------------------------------------
dos           37.1  ====================================
tres          41.3  ----------------------------------------
tres          18.0  ==================
quatro        44.1  --------------------------------------------
quatro        28.6  ============================
ultimo          50  --------------------------------------------------
ultimo          33  ================================
<<<
```

There's a little bit of manual editing left to replace the p's and the q's with proper headings, but I haven't figured out a way yet to do this automatically.

A BIBLIOGRAPHY REPORT WRITER

-- Michael J. McIntyre

The T/Maker database program can be used to create, order and sort a bibliography of books and articles. Elyse Sommer explained how to set up a bibliography program in Vol. 4, No. 1 of T/MUG (Jan./Feb. 1985). I have expanded somewhat on the program she designed to allow for the complicated punctuation requirements of some bibliographies.

My program has two separate databases, one for articles and one for books. I keep these separate because the punctuation requirements for articles and books are very different. The bibliographical information required for books, moveover, may differ substantially from that required for articles.

Here is the form and record definition I use for the bibliography of an article I have been writing on progressive taxation and distributive justice. Note that the forms provide for the possibility of multiple authors and for VERY long article titles. I have also appended a couple of actual articles from my bibliography. My system follows pretty much the pattern I learned from Elyse Sommer.

The File: JusticeA

```
<form>
              BIBLIOGRAPHY FOR DISTRIBUTIVE JUSTICE ARTICLE
                          *** Articles ***
  AUTHOR(S)
   Last Name: {LASTNAME          }<
  First Name: {FIRSTNAME         }< }}     (Include Middle Name or Initial
2 Last Name:  {LASTNAME2         }< }}     as part of first name)
2 First       {FIRSTNAME2        }< }}
3 Last Name:  {LASTNAME3         }< }}
3 First       {FIRSTNAME3        }< }}
      Title:  {TITLE                                                    }<
(continued)   {TITLE2                                                }< }}
    Journal:  {JOURNAL                                                  }<
Volume/Page:  {VO#LUME}/{PAGE} }}
       Year:  (19{#YEAR}) }}
      Topic:  {TOPIC                                                 }< }}
 Ref. Codes:  {CODES                                                 }< }}
      Notes:  {NOTES                                                 }< }}
(continued)   {NOTES2                                                }< }}
<end>
```

-- this file continued on next page

-- this file continued from prior page

```
<record>
1 {LASTNAME          }
2 {FIRSTNAME         } }}
3 {LASTNAME2         } }}
4 {FIRSTNAME2        } }}
5 {LASTNAME3         } }}
6 {FIRSTNAME3        } }}
7 {TITLE                                                          }
8 {TITLE2                                                      } }}
9 {JOURNAL                                                        }
10 {VO#LUME}/{PAGE} }}
11 {#YEAR} }}
12 {TOPIC                                                        } }}
13 {CODES                                                        } }}
14 {##CHANGED}
15 {NOTES                                                        } }}
16 {NOTES2                                                       } }}
<end>
1 Bittker
2 Boris I.
7 Equity, Efficiency, and Income Tax Theory:
8 Do Misallocations Drive Out Inequities?
9 San Diego Law Review
10 16  /735
11 79
14 1
1 Blum
2 Walter J.
7 Revisiting the Uneasy Case for Progressive Taxation
9 Taxes--The Tax Magazine
10 60  /16
11 82
14 1
1 Hurley
2 S.L.
7 The Unit of Taxation Under an Ideal Progressive Income Tax
9 Oxford Journal of Legal Studies
10 4   /157
11 85
14 1
1 Schmalbeck
2 Richard
7 Income Averaging After Twenty Years: A Failed Experiment in
8 Horizontal Equity
9 Duke Law Journal
10 1984/509
11 84
14 1
```

To produce a printout of the entries in the database in an appropriate format, I have created a record definition, which I call CardsA, and an intermediate database, which I call RulesA. The function of RulesA is to SET the punctuation needed for CardsA. The intermediate step is necessary because CardsA squeezes many of the fields in the database. For an explanation of the problem of squeezed data and the SET command, see "Using 'Set' and 'Squeezed' Data" by Bert Zitek in Vol. 4, No. 1 of TMUG (Jan./Feb. 1985). The CardsA record is reproduced below.

The File: CardsA

```
GET rulesa SELECT justicea end SET DELETE temp RENAME temp SAVE GET
this should continue on prior line cardsa SELECT temp end DELETE temp ALIGN E
..Form for printing JusticeA on 5x3 cards
<record>
.new
., {LASTNAME       ! }, {FIRSTNAME        }{P1}{P2#>}
   {LASTNAME2      ! }, {FIRSTNAME2       }{P3}{P4#>} }}
   {LASTNAME3      ! }, {FIRSTNAME3       }, }}
   "{TITLE          !                                                  }{P5}
   {TITLE2         !                                                },"  }}
   {VO#LUME!} {JOURNAL                                                        }
   {PAGE} (19{Y#EAR}).

<end>
.indent 1
.pagesize 18
     <<<                                    ->>
..<HERE>
```

The top line of my CardsA file is the "DO" line, which I will explain shortly. (It should be all on one line.) The <record> portion of the file sets up the fields in the order I want them printed. Note that I have put the author's last name first and have followed standard law review format for the citation. Of course the record form could easily be modified for other citation formats, including a bibliography listing with the author's name in normal order. The interesting aspect of the record form is the fields for punctuation, labelled {P1} to {P5}. The contents of these fields are established using the SET command and some rules. Those rules are found in my RulesA file, reproduced below.

The File: RulesA

```
<rules>
P1 = , when LASTNAME2 = " " or LASTNAME3 <> " "
P2 = " and" when LASTNAME2 <> " "  AND LASTNAME3 = " "
P3 = , when LASTNAME3 = " " and LASTNAME2 <> " "
P4 = " and" when LASTNAME3 <> " "
P5 = ',"' when TITLE2 = " "
P5 = " " when TITLE2 <> " "
<end>
<record>
1 {LASTNAME          }
2 {FIRSTNAME         } }}
3 {LASTNAME2         } }}
4 {FIRSTNAME2        } }}
5 {LASTNAME3         } }}
6 {FIRSTNAME3        } }}
7 {TITLE                                                          }
8 {TITLE2                                                      } }}
9 {JOURNAL                                                        }
10 {VO#LUME}/{PAGE} }}
11 {#YEAR} }}
12 {TOPIC                                                        } }}
13 {CODES                                                        } }}
14 {##CHANGED}
15 {NOTES                                                        } }}
16 {NOTES2                                                       } }}
17 {##P1} }}
18 {P2#>} }}
19 {##P3} }}
20 {P4#>} }}
21 {#P5} }}
<end>
```

The first two rules define the contents of fields {P1} and {P2}, which set the punctuation after the first author. If there is only one author, or if there are three authors, the first name of the first author should be followed by a comma. If there are two authors, however, the first name of the author should be followed by a space and the word "and". The first case is provided for by field {P1} and the second case by field {P2}. Note that I inserted the "right justify" command ">" into the field {P2} in order not to lose the space before the "and". Fields {P3} and {P4} perform a similar function for the punctuation after the second author. Field {P5} provides for punctuation after the title when the title is short enough to fit into the field {TITLE}. An extended title is continued to field {TITLE2}. The punctuation after {TITLE2} is provided in the record definition. Because I used double curly brackets "}}" on the line with that field, neither the field nor the punctuation will appear if the field is empty.

To produce a printout, I call up CardsA and at the prompt type "DO". The "DO" line instructs T/Maker to GET RulesA, to SELECT the entire contends of my articles bibliography from the file JusticeA, to SET the contents of the punctuation fields, and to save the resulting file to disk. In order to keep my file RulesA in its original state, I rename it "temp", and it is the file "temp" that is actually saved to disk. (Those with a ram disk can save "temp" to their ram disk.) The "DO" line then calls up CardsA again and selects all the entries from "temp". I have set up some aligning wedges and printing instructions below the record definition. The "DO" line then tells T/Maker to align the data entries, and I am ready to print. Because I plan to print the bibliography entries on tractor-feed 5x3 index cards, I defined the pagesize as 18 (6 lines per inch x 3 inches) and included a new page command, ".new" in the record definition.

The record form and the rules for my books database are somewhat more complicated. I have had to make provision for many possibilities that would not arise for articles, including (1) that the author of the book might be an institution, such as the Ways and Means Committee of Congress, which has a long "last name" and no first name; (2) that the listed author(s) might be editor(s); (3) that the data entry might refer to the author(s) of a chapter of a book, which may also have an editor(s); (4) that the book may be one of several volumes.

To produce the desired printout, I have used the same general procedure I developed for my articles database. The file for my books database I have labeled JusticeB. The <form> and <record> definitions are shown below. Note that I used a simple checkoff to indicate whether the principal authors are editors. I use the SET command to insert the appropriate designation at the time the file is to be printed.

The File: JusticeB

```
<form>
              BIBLIOGRAPHY FOR DISTRIBUTIVE JUSTICE ARTICLE
                          *** Books ***
  AUTHOR(S)
   Last Name: {LASTNAME                                    }<
  First Name: {FIRSTNAME          }< }}    (Include Middle Name or Initial
2 Last Name: {LASTNAME2           }< }}    within First Name)
2 First    : {FIRSTNAME2          }< }}
3 Last Name: {LASTNAME3           }< }}    Mark "x" in box if author(s) is edito
3 First    : {FIRSTNAME3          }< }}      [{##ED}]
 Book Title: {BOOK                                                        }<
(continued)  {BOOK2                                                    }< }}
Chap. Title: {CHAPTER                                                  }< }}
  Editor(s): {EDITOR                                                   }< }}
Volume/Page: {VO#LUME}/{PAGE} }}
  Publisher: {PUBLISHER                                                }< }}
       Year: (19{#YEAR}) }}
      Topic: {TOPIC                                                    }< }}
 Ref. Codes: {CODES                                                    }< }}
      Notes: {NOTES                                                    }< }}
(continued)  {NOTES2                                                   }< }}
<end>
```

-- this file continued on next page

-- this file continued from prior page

```
<record>
1 {LASTNAME                                    }
2 {FIRSTNAME        } }}
3 {LASTNAME2        } }}
4 {FIRSTNAME2       } }}
5 {LASTNAME3        } }}
6 {FIRSTNAME3       } }}
7 {BOOK                                                          }
8 {BOOK2                                                      } }}
9 {CHAPTER                                                    } }}
10 {EDITOR                                                     } }}
11 {VO#LUME}/{PAGE} }}
12 {PUBLISHER                                                  } }}
13 {#YEAR} }}
14 {TOPIC                                                      } }}
15 {CODES                                                      } }}
16 {##CHANGED}
17 {NOTES                                                      } }}
18 {NOTES2                                                     } }}
19 {##ED} }}
<end>
1 Institute for Fiscal Studies
7 The Structure and Reform of Direct Taxation: Report of a
8 Committee Chaired by Professor J. E. Meade
12 George Allen & Unwin
13 78
14 Comprehensive tax reform of income tax
16 1
1 McIntyre
2 Michael J.
3 Sander
4 Frank E. A.
5 Westfall
6 David
7 Readings in Federal Taxation (2nd ed.)
12 Foundation Press
13 83
14 collection of readings
16 1
19 x
1 Musgrave
2 Richard A.
7 Tax Policy Options in the 1980's
9 Tax Reform or Tax Deform
10 Wayne Thirsk & John Whalley, Eds.
```

-- this file continued on next page

-- *this file continued from prior page*

```
11     /19
12 Canadian Tax Foundation
13 82
14 Horizontal Equity
16 1
1 Vartiainen
2 Henri
7 Reforms of Tax Systems
9 Progressive Income Taxation, Disincentives and Barter
10 Karl Roskamp and Francesco Forte, Eds.
11     /103
12 Wayne State University Press
13 81
16 1
```

I also have a file with the rules that control the SET command, labeled RulesB. That file is displayed below. Note that it has a set of fields for punction, P1 etc. and a set of fields, E1 etc., that specify, in the appropriate cases, that the person listed as the author(s) is an editor.

The File: RulesB

```
<rules>
..CHANGED = " "
E1 = ", Ed." when ED <> " " and LASTNAME2 = " "
E2 = ", Eds." when ED <> " " and LASTNAME2 <> " " and LASTNAME3 = " "
E3 = ", Eds." when ED <> " " and LASTNAME3 <> " "
P1 = , when (LASTNAME2 = " " or LASTNAME3 <> " ") and FIRSTNAME <> " "
P2 = " and" when LASTNAME2 <> " " and LASTNAME3 = " "
P2 = " " when LASTNAME2 = " "
P3 = , when LASTNAME3 = " " and LASTNAME2 <> " "
P4 = " and" when LASTNAME3 <> " "
P5 = '\' when BOOK2 = " "
P5 = " " when BOOK2 <> " "
P6 = , when PAGE = " " and BOOK2 = " "
P7 = , when PAGE = " " and BOOK2 <> " "
<end>
<record>
1 {LASTNAME                                    }
2 {FIRSTNAME        } }}
3 {LASTNAME2        } }}
4 {FIRSTNAME2       } }}
5 {LASTNAME3        } }}
6 {FIRSTNAME3       } }}
7 {BOOK                                                    }
8 {BOOK2                                                 } }}
```

-- *this file continued on next page*

-- *this file continued from prior page*

```
9 {CHAPTER                                                    } }}
10 {EDITOR                                                     } }}
11 {VO#LUME}/{PAGE} }}
12 {PUBLISHER                                                  } }}
13 {#YEAR} }}
14 {TOPIC                                                      } }}
15 {CODES                                                      } }}
16 {##CHANGED}
17 {NOTES                                                      } }}
18 {NOTES2                                                     } }}
19 {##ED} }}
20 {##P0} }}
21 {##P1} }}
22 {P2#>} }}
23 {##P3} }}
24 {P4#>} }}
25 {#P5!} }}
26 {#P6!} }}
27 {#P7!} }}
28 {E1!} }}
29 {E2! } }}
30 {E3! } }}
<end>
```

The <record> form for the printout of book citations is labeled CardsB. It has the same "DO" line as CardsA, except that it calls up a different set of files. I included in the <record> form an instruction to underline the title of the book. T/Maker is smart enough to know not to underline punctuation marks. It is not smart enough to eliminate the space taken up by the underline command. Some techniques for dealing with that problem are described in the manual for T/Maker. The manual also explains how to avoid extra space afer a middle initial. The CardsB file and some examples of formatted output are displayed below.

The File: RulesB with Formatted Output

```
GET rulesb SELECT justiceb end SET DELETE temp RENAME temp SAVE GET
this should continue on above line cardsb SELECT temp end DELETE temp ALIGN E
..Form for printing JusticeB (books) on 5x3 cards
<record>
.new
., {LASTNAME       ! }, {FIRSTNAME        }{E1 }{P1}{P2#>}
   {LASTNAME2      ! }, {FIRSTNAME2       }{E2  }{P3}{P4#>} }}
   {LASTNAME3      ! }, {FIRSTNAME3       }{E3  }, }}
   "{CHAPTER                                          !          }" in }}
   {VO#LUME!}\{BOOK          !
   {BOOK2          !                                                }{#P7!}\ }}
```

-- *this file continued on next page*

-- this file continued from prior page

```
   {PAGE}, }}
   {EDITOR                                                    }, }}
   {PUBLISHER                                                 } }}
   (19{Y#EAR}). }}
<end>
.indent 1
.pagesize 18
     <<<                                    ->>
..<HERE>
.new
., Institute for Fiscal Studies,
     The Structure and Reform of Direct
     Taxation:  Report of a Committee Chaired
     by Professor J.  E.  Meade,  George Allen
     & Unwin (1978).

.new
., McIntyre, Michael J.,
     Sander, Frank E.  A.  and Westfall,
     David, Eds.,  Readings in Federal
     Taxation (2nd ed.),  Foundation Press
     (1983).

.new
., Musgrave, Richard A.,
     "Tax Reform or Tax Deform" in  Tax Policy
     Options in the 1980's  19, Wayne Thirsk &
     John Whalley, Eds., Canadian Tax
     Foundation (1982).

.new
., Vartiainen, Henri,
     "Progressive Income Taxation,
     Disincentives and Barter" in  Reforms of
     Tax Systems  103, Karl Roskamp and
     Francesco Forte, Eds., Wayne State
     University Press (1981).
```

I could have simplified my system somewhat by eliminating the intermediate files, RulesA and RulesB, and used the SET command on the original database. Because it is easy to mess up using the SET command, however, I decided to keep my original file intact. One advantage of my system is that is saves some storage space on disk. A disadvantage is that it requires extra disk space at the time of formatting, which is a problem on my Kaypro but not on my IBM, which has a hard disk.

Portable RULES Definitions for Databases that Stretch Across Several Data Files

-- Ron Roizen

An annoying thing about T/Maker's database system is its small file size capacity. One consequence of small file size, of course, is that big databases must be distributed across several data files. Ordinarily, such data files will be linked together with the <continue> option. That way, you can SELECT from the entire database with a single SELECT command. Unfortunately, the <continue> option does not apply to T/Maker's SET command. Every time you want to use the SET command on the whole database you must write the appropriate new rule into each data file and execute a separate SET command for each.

Fortunately, there's a relatively simple way to speed up this work. It requires creating two new files, NEWRULE and WORK. Consider a simple example. Suppose that you have a database made up of five data files -- call them DATA.1 through DATA.5. The new file called NEWRULE contains only the new rule (or rules) you want to employ. Suppose this rule adds two fields (field.1 and field.2) to provide the data for a third field (field.3). In order to simplify things as much as possible, further suppose that a field for field.3 already exists in your data files' Record Definitions. The NEWRULE file would then consist of a single line:

```
field.3 = field.1 + field.2 end ***
```

I've added three asterisks to the rule for a reason that will be clear in a moment. The next problem is that of transporting this new rule to each of the five data files and executing it. I'll assume that each of your data files has a pre-existing Rules Definition comprised of the rules you regularly use in manipulating that file. In that case, the following command can be used:

```
GET DATA.1 FIND <rules> FIND = INSERT NEWRULE 1 SET DROP *** SAVE
```

And what does this command do? First, it GETS DATA.1. Next (with FIND <rules>), it sends the top of the screen to the DATA.1's <rules> line. Next (with FIND =), it sends the top of the screen to the next line down, the line containing the first pre-existing rule in the data file -- this places the top of the screen at the right position for inserting the NEWRULE file. Next (with INSERT NEWRULE), it inserts the new rule into the old Rules Definition. Next (with 1), it sends the top of the screen back to the top of the data file. Next (with SET), it SETS the file. Next (with DROP ***) it deletes the new rule from the file, retaining the original data-file rules only -- this feature is optional. The asterisks provide a convenient means for DROPPING the new rule if you wish to. Finally, the command SAVES the file.

Next, CREATE the WORK file. This file is comprised solely of a series of DO lines. In the case at hand, the WORK file would consist of the following five lines:

```
G DATA.1 FIND <rules> FIND = INSERT NEWRULE 1 SET DROP *** S G WORK 2 DO
G DATA.2 FIND <rules> FIND = INSERT NEWRULE 1 SET DROP *** S G WORK 3 DO
G DATA.3 FIND <rules> FIND = INSERT NEWRULE 1 SET DROP *** S G WORK 4 DO
G DATA.4 FIND <rules> FIND = INSERT NEWRULE 1 SET DROP *** S G WORK 5 DO
G DATA.5 FIND <rules> FIND = INSERT NEWRULE 1 SET DROP *** S G WORK.DUN
```

How do you use the WORK file? You GET the file on your screen. Then, you give a single DO command. What does the file do? Once again, it (1) transports the NEWRULE file to successive data files, (2) SETS the file, (3) DROPS the new rules (optional), and (4) SAVES the newly-SET data file. The last segment of each line (5) re-GETS the WORK file, (6) sends the top of the screen to next line, and, finally, (7) invokes the DO command on that line (putting the new line into action). In this way T/Maker will automatically continue down the five lines of the file, carrying out the new rule on all five of the database's data files.

Notice that the fifth line of the WORK file ends with the command "G WORK.DUN." This is a little frill for your routine -- WORK.DUN is a file containing the following:

```
THE NEW RULE(S) HAVE BEEN APPLIED

TO ALL DATA FILES
```

Thus, when your routine has finished its work, the WORK file puts this notice on your screen.

Henceforth, then, it will be an easy matter to impose new rules on a multi-file database. All you need do:

1. GET NEWRULE, inscribe your new rules into it, and SAVE the file.

2. GET the WORK file and use the DO command.

T/Maker does the rest!

Spacing Following Punctuation

-- Richard S. Russell

T/Maker aligns text so that there are two spaces behind each period. exclamation point! question mark? or colon: which is followed by a space in the original text. It also places two spaces behind each period." exclamation point!" or question mark?" which is followed by a double quotation mark in the original. It places no spaces behind each period, exclamation point!, question mark?, colon:, or (for that matter) any other character followed by anything other than a space or double quotation mark.

But suppose you prefer just one space after the colon. Or suppose you're using a period to indicate an abbreviation, rather than the end of a sentence. Or suppose you want to refer ot the "!" in a discussion of how to punctuate. How do you get just one space after the punctuation mark? Here are two techniques, both of which assume that you can use your computer to create "high bit" characters, as described in the T/Maker manual.

METHOD ONE: It's possible to set up lines in the print.utl table which will print a high bit character as a regular one, rather than an italicized one (or however your printer interprets high bit characters. Here's what I use for the Epson MX-100:

Char. Found	Characters Printed	Remarks:
161	033 000	alternate ! = normal !
174	046 000	alternate . = normal .
186	058 000	alternate : = normal :
191	063 000	alternate ? = normal ?

Since the lines reflect plain-vanilla ASCII codes, they ought to work on a wide variety of other printers as well.

With these lines in place, simply use the high bit character whenever you want it to be followed by only one space. The align command won't recognize the high bit character as a regular one, but the printer will.

METHOD TWO: You can use a high bit space following the regular punctuation mark and immediately preceding the next word. This creates a double word which the T/Maker align command treats as a single word with am embedded punctuation mark. The only space you get in this double word is the high bit space you put there yourself.

The second technique can produce funny-looking results if the double word gets to be too long. Since align treats ti as a single word, it may be moved as a unit onto a separate line even though it would normally have been broken in the middle, after the punctuation mark, to appear on separate lines.

A variation on the second technique is useful for parenthesized comments. Unfortunately, T/Maker doesn't put two spaces after a period.) exclamation point!) or question mark?) which is followed by a parenthesis, even though you would normally want two spaces there. The solution: put one space there yourself, using a high bit character for the purpose.

A T/Maker Pop-Up Calculator

-- Richard S. Russell

One of the hottest-selling pieces of software on the market today is Borland International's SideKick. This program sits quietly in the main memory of the computer, waiting for you to call it up to do all sorts of nifty little jobs for you. The emphasis is on *little* jobs. SideKick isn't a full-fledged spreadsheet, word processor, or communication package, and it doesn't pretend to be. But it can handle a lot of quickie tasks that a computer should be good for.

Perhaps the most frequently used feature of this popular package is the pop-up calculator. How many times have you been partway through a report or a letter and needed to run some simple set of numbers through a couple of quick calculations? There you are, sitting in front of one of the most powerful number crunchers every to occupy a desktop, and you can't even make it add a column of figures together without an act of Congress.

The pop-up calculator is the answer to this problem, and now one can be created by T/Maker users as well. It's simply a file entitled CALC, and you bring it into your work area by escaping from the editor andy typing INSERT CALC. The screen is immediately filled with a spreadsheet that's designed for some simple calculations, using the rules of T/Maker's spreadsheet command structure. You can re-enter the editor and type in whatever numbers are appropriate for your immediate needs, home the cursor, escape from the editor and type COMPUTE, and the answers will be generated right before your eyes.

The T/Maker calculator has certain advantages over others. Because it uses T/Maker's spreadsheet format, you can see all of the numbers you're dealing with laid out visibly. You can initialize CALC so it shows up on your screen with certain frequently used constants already in place. You can tailor the example line so it will produce numbers to the degree of precision you want. And, most significantly, you can keep the results of your calculation a part of the document you're working on, so you can move them around wherever you want.

To make the calculator vanish from the screen, type "do", and a pre-written T/Maker command will clear it all out, leaving you with your original document intact. (If you wish to retain any of the results of the calculation, you must remove the word "eXaMpLe" from the lines they're on, since that's the word T/Maker looks for when deciding which lines to discard when clearing the screen.)

On the next page (sideways) is wht the CALC file looks like:

THE FILE: CALC

drop eXaMpLe drop jC2 drop uC2

eXaMpLe	99,999,999.99	99,999,999.99	99,999,999.99	99,999,999.99	99,999,999.99	99,999,999.99	99,999,999.99	
jC28=	2.718281828	3.141592654						
jC29	ste	stp						
=								eXaMpLe
uC28	+	+	=+	*	=+	/	=	
+	0	0	0	1	0	1	0	eXaMpLe
+	0	0	0	1	0	1	0	eXaMpLe
+	0	0	0	1	0	1	0	eXaMpLe
+	0	0	0	1	0	1	0	eXaMpLe
+	0	0	0	1	0	1	0	eXaMpLe
+	0	0	0	1	0	1	0	eXaMpLe
=+	SUBTOTAL							eXaMpLe
+	0	0	0	1	0	1	0	eXaMpLe
+	0	0	0	1	0	1	0	eXaMpLe
+	0	0	0	1	0	1	0	eXaMpLe
+	0	0	0	1	0	1	0	eXaMpLe
+	0	0	0	1	0	1	0	eXaMpLe
+	0	0	0	1	0	1	0	eXaMpLe
=+	SUBTOTAL							eXaMpLe
=	TOTAL							eXaMpLe
eXaMpLe								eXaMpLe

The very first line, which contains the "drop" commands, is the one which you will activate with a DO command after you' re done with the worksheet. The bizarre capitalization is to prevent these commands from discarding anything following the calculator which might contain the word "example" or the letter combination "jc" or "uc".

The "jC" lines create and store constants for later use. Even though the constants will be displayed on the line beginning with "=" to only three significant digits of precision, as specified by the example line, they are stored internally by T/Maker to the full 10 significant digits that were used in the "jC28" definition. (Obviously, if you have other constants that are more appropriate to your own work, you would substitute those here; if you generally don't need constants, you could eliminate these three lines altogether.)

The worksheet has been set up to provide for simple columnar addition to be done vertically (with two subtotals for intermediate results) and for multiplications and divisions to be done horizontally. Again, this can be altered to suit your own requirements.

The pre-entered zeroes and ones serve two purposes. First, the zero is what mathematicians call the "additive identity", that is, you can add it to anything and not change what you started with. By the same token, the one is the "multiplicative identity". These values, then, serve as place-holders which enter into the computation, prevent you from dividing by 0 and allow the values which you have entered to be passed intact from column to column and line to line. Second, these values sit at the very beginning of each column, so that you can get to the column easily for data-entry purposes by using the "advance to next word" keystrokes. (It's very unlikely that the tabs you have set up for your document will exactly match the columns of the CALC file, so some other convenient way is needed for navigation within CALC.)

The words "SUBTOTAL" and "TOTAL" will, of course, be replaced by actual numbers as soon as you issue the Compute command. They're only there at the beginning to keep you from inadvertently entering data in a place where it will only get wiped out.

The last "eXaMpLe" line is there to stop the compute commands and the number-positioning effects of the first "eXaMpLe" line from spreading to the remainder of your file.

Here are a couple of practical suggestions. (1) For data entry, start by getting to the first "0" in the spreadsheet, then use the "screen right" keystrokes so that the left edge of the screen moves over to the left edge of your first data column. This lets you see more of the work area and also allows you to simply hit "return" to get to the next line if all you're doing is entering a column of figures. (2) Don't change the "+" signs in position 1 to "-"s if you want to subtract. Instead, enter negative numbers in the data area: adding a negative number is the same as subtracting a positive one. (3) Remember to home the cursor before escaping the editor to type Compute -- otherwise you won't get very interesting results.

Finding Your Place in a Linked Series of Database Files

-- Royal Farros

Suppose we wanted to update a client's address. Because we have so many clients, we've had to link five address databases together. (See T/MUG August/Sept 1985.) The problem is we've forgotten which database our client is in. Looks like we'll have to get each database and search through them individually -- unless we can use some T/Maker Trickery. By "indexing" our records, we'll be able to quickly find which record we need to be in. We'll need to alter the current databases by inserting a DATABASE field at the bottom of the record definition.

```
<Record>
   .
   .
99 {DATABASE}
<End>
```

Next, we'll insert a rule that sets this database field to the name of the database we're in.

```
<Rules>
   .
   .
DATABASE = "filename"
<End>
```

Be sure to SET this new rule in motion before you save these changes. Now we will want to create a special file called SEARCH.1. The file will look like this:

```
<Form>
Who name do you want to find?  {SEARCHITEM}
In what database field is this contained?  {DATAFIELD}
In what database should we start looking for this name?  {SERIES}
<End>
<Record>
get search.2  select {SERIES} when {DATAFIELD}={SEARCHITEM} end 4 do
<End>
```

Next, create a special file called SEARCH.2. This file will look like this:

```
<Record>
Get {DATABASE} Update
<End>
```

To use this system, GET search.1, fill it in (do not save) then at the WHAT NEXT? prompt, type: `9 DO` <Enter>. This little routine should place you directly in the file you want, ready to UPDATE it and use Search for String to find your data.

T/IP: CLEANING A DATABASE?

Bob Ackerman passes along this database tip dealing with database printing and field labels.

The T/Maker print design commands ".clean" and ".noclean" can be used to surpress the printing of field labels. A "clean" will temporarily wipe out the first 7 columns in a T/Maker file during a print.

Your database can look like this:

```
<Record>
.clean
01      {Name                   }
02      {Address                }
03      {State                  }
.noclean
<End>
```

Remember, don't put anything except field labels (that you don't want printed) in the first 7 columns.

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 7% 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! Our number is (415) 962-0195.

UPGRADE FROM T/MAKER III TO T/MAKER INTEGRATED SOFTWARE MUST HAVE PROOF OF T/MAKER III SERIAL NUMBER:	
FOR SAME COMPUTER TYPE: WILL RECEIVE BOUND MANUAL	$175.00
FOR OPERATING SYSTEM OR COMPUTER OTHER THAN CURRENT LICENSE: WILL RECEIVE BOUND MANUAL	$250.00
T/MAKER INTEGRATED SOFTWARE FOR SECOND OPERATING SYSTEM BOUND MANUAL. MUST HAVE T/MAKER I.S. SERIAL NUMBER.	$225.00

T/MUG 1985 INDEX

Topic/Article Name	Volume, Number, Page
Accumulating Winnings/Losings (Sports)	V4 N2 P5
Addresses (2-up)	V4 N1 P11
Aging Outstanding Balances	V4 N5 P2
Aligning Wedges	V4 N1 P3
Anadex WP-6000 Print.utl	V4 N2 P8
Appointment Schedule	V4 N6 P2
Bar Charting Made Automatic	V4 N6 P4
Bibliographies (simple)	V4 N1 P14
Bibliographies (complex)	V4 N6 P8
Billing System	V4 N5 P16
Bowling Application	V4 N2 P20
Commands, via Database	V4 N4 P10
Correlations	V4 N1 P24
DIF File Creation	V4 N2 P3
Data Fields, Protecting	V4 N3 P2
Data Fields, shortening	V4 N5 P14
Database Made Easy	V4 N4 P3
Database Search Across Linked Files	V4 N6 P23
Database, Multi-file	V4 N2 P9
Dates with T/Maker	V4 N1 P5
Do Lines - Creating systems	V4 N5 P12
Form Letters	V4 N5 P25
Inventory Count	V4 N1 P9
Multi-user tips	V4 N5 P14
Order Entry	V4 N1 P9
Panasonic KX-P1091 Print.Utl	V4 N3 P14
Pop-up Calculator	V4 N6 P20
Printing Multiple Copies	V4 N5 P13
Real Estate Tracking	V4 N3 P21
Room Scheduler	V4 N3 P20
Rules Definitions Across Multi-Files	V4 N6 P17
Sales Forecasting	V4 N4 P16
Screen Forms for Database	V4 N4 P19
Spacing Following Punctuation	V4 N6 P19
Squeezed Data, problems with	V4 N1 P17
Surveys	V4 N3 P4
TRS 80 and Double Spacing	V4 N5 P11
Typewriter Simulator	V4 N4 P15
Unemployment Compensation	V4 N2 P13
Visicalc User's Guide:	V4 N1 P19
	V4 N2 P16
	V4 N3 P15

T/Maker Consultants

T/Maker consultants are T/Maker specialists who are willing to consult on your applications for a fee. Please contact them directly for any further information on their work. If you are interested in being listed as a consultant, please send us your applications (on paper and on disk) and a brief statement about your area of specialization.

Don Baack, 6495 S.W. Burlingame, Portland, OR 97201, (503) 244-2741, TELEX 277331 SMA UR: *Manufacturing; operations analysis; cost accounting; inventory control; timber and wood products.*

Dick Danielson, Strategic Advisory Consultants, Inc., P.O. Box 137, Star Prairie, WI 54026, (715) 248-3434: *Fully integrated accounting system including general ledger, financial statements, accounts receivable, account analysis, operations analysis. Multiple divisions, accounts, and ledgers can be accommodated.*

Gus Korman, English Only Computer Co., 5222-1 Lindley Ave., Encino, CA 91316, (818) 344-2422: *Accounting system including many of the most popular accounting modules.*

D. Michaut, Dept. Org. Societe Generale Des Coop., 27, 33, quai A. le Gallo, 92517 Boulogne-Billancourt Cedex, FRANCE (1) 604-9178: *Retail and warehouses, Price and margin management, stock management, merchandising, invoicing, mailings.*

Robert Payne, BCS, 1210 Smith Street, Charleston, WV 25301, (304) 343-9471: *Lawyer time accounting, general office accounting, A/P, A/R, payroll, general ledger.*

Ron Roizen, 1818 Hearst Street, Berkeley, CA 94703, (415) 848-9098: *Automating small businesses, including accounts receivable/billing, mail-order accounting and inventory. Survey research, tracking organization membership, dues. Also training and documentation.*

Elyse Sommer, Box E, 962 Allen Lane, Woodmere L.I. NY, (516) 295-0046: *Text-related data base use; custom macros with keyboard enhancers; on-screen writing "helpers" for correspondence, reports, manuals; fact sheet outlines; phrase files; boiler plate.*

Emil Widmer, CFM, Baarerstrasse 45, Postfach 708, CH-6301 Zug, Switzerland, tel:042 21 08 87: *consulting on management of uses of computers, management seminars using T/Maker in Switzerland and Germany, engineering and design, production engineering, personal resources, finance.*

Robert D. Williamson, Can-American Electronics, 7220 Taft Street, Hollywood, FL 33024, (305) 966-5588: *Small business and manufacturing software. Hardware.*

T/Mug Back Issues

If you found this issue useful, you might want to consider picking up the back issues. T/Mug has been published since 1982, each issue aiming for a mix of articles for the novice T/Maker user through the expert.
The T/Mug also serves as a forum for users of particular machines and in particular industries. We try to include the applications most requested in letters and phone calls. Past issues have included: invoices, checking account management, statistics, personalized form letters, transferring data files, inventory, tracking athletic events, time calculations, and many more. The applications, like the product, have evolved over the years, so you may find the more recent issues to be of more value.
To order, please enclose this form with your check (drawn in US dollars) and mail it to the address below. Customers outside continental US please add $5 per box checked for shipping.

Back Issues -- Please send me:

[] Back issues: 1985 (to present) $12.00

[] Back issues: 1984 (six issues) $12.00

[] Back issues: 1983 (five issues) $10.00

[] Back issues: 1982 (five issues) $10.00

T/Mug Subscriptions:

You might also want to subscribe (or renew your subscription) to the T/MUG newsletter. This entitles you to six bimonthly issues like the one in your hands.

To order, please enclose this form with your check (drawn in US dollars) and mail it to the address below. Customers outside continental US please add $20 for first class mailing charges.

Subscription -- please:

[] Enter my subscription for one year $15.00

[] Renew my subscription $15.00 (my T/Mug mailing label is enclosed)

Name_____

Company_____

Street_____

City:_____State_____Zip_____

T/Maker Serial Number(s)_____

Please make check out to T/Maker Users' Group and Mail to
T/Maker Company * 2115 Landings Drive * Mt. View, CA 94043

T/MUG
T/MAKER USER'S GROUP
T/MAKER COMPANY
2115 LANDINGS DRIVE
MOUNTAIN VIEW, CA 94043

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