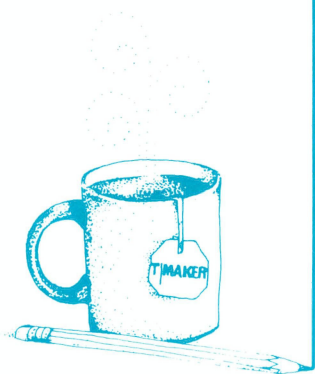


T/MUG

T/MAKER USER'S GROUP NEWSLETTER
VOLUME 4, NUMBER 5, SEPTEMBER/OCTOBER 1985

In This Issue:

Aging Outstanding Balances, Allocating Payments. . .	2
Double-Spacing and Model 4/4p	11
Using Do Lines to Create Systems	12
T/lps: Multi-user, short data fields, Multiple copies. .	13
Monthly Billing System	19
Personalized Form Letters	25



T/Maker Users' Group Newsletter
Copyright 1985 T/Maker Company. All Rights Reserved.

T/MAKER NEWSFRONT

-- Heidi Roizen

Office Gears Up for Holiday Season

Though T/Maker Integrated does not exhibit seasonal sales cycles, our Macintosh products do. So, we're gearing up for the holiday season with increased inventory and special promotions. Since the office will be busier from now until December, please pardon any delays with tech support. As always, we will try to get you the right answer as quickly as possible.

Calling all Applications:

I'd like to do some special T/Mug issues in the future, with vertical application themes like legal, doctor/dentist, accountant, construction, investment management, etc. If you've developed some applications for any of these professions, please consider sharing them with other T/Mug readers. You will be rewarded for your efforts with T/Maker trinkets and/or special upgrade offers when we come out with our new product early next year. Please submit your applications on paper and on disk!

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 PROOF OF T/MAKER I.S. SERIAL NUMBER.	\$225.00
--	----------

Aging Outstanding Balances and Allocating Incoming Payments

-- Ron Roizen

I recently developed a simple accounts receivable system -- called "T/Bill" -- written wholly in T/Maker. The system is designed to manage the billing needs of a small company. It logs and calculates charges and payments, it prepares a monthly bill for all clients needing one, and -- once a month's bills have been prepared -- it archives the current month's charges and payments for historical purposes. Among other things, this system needed to (1) "age" outstanding client balances and (2) allocate incoming payments to the client's oldest balance first. This article describes how T/Bill accomplishes these two tasks.

Aging balances or debts means simply that the system keeps track of the age of an outstanding balance amount. T/Bill thinks in terms of four ages: a balance left over from the month just closing is regarded as "1-30 days old," a balance left over from the month previous to that is regarded as "31-60 days old," a balance left over from the month previous to that is regarded as "61-90 days old," and all balances left over for more time than that are regarded as "90-days-plus balances."

T/Bill allocates payments to outstanding balances following an "oldest first" rule. For example, suppose that Client A001 charged \$100.00 in services in the current month and in addition showed a 1-30 day balance of \$50.00. Further suppose that this client sent in a check for \$75.00 in the current month. T/Bill would pay off the outstanding 1-30 day balance of \$50.00 first, and then apply the remaining \$25.00 to the current month's \$100.00 balance. That would leave a current balance of \$75.00.

A specific file carries out these aging and payment allocation tasks. This file--let's call it FIFO--is a database file. Its record definition consists primarily a simple table. Imagine, for a moment, that this file looked something the one illustrated in Figure 1, below.

At month's end, the FIFO file needs to be informed of each client's total charges and payments. As it happens, such charge and payment information is kept in two other T/Bill files -- two files specifically designed for logging and calculating charges and payments. Therefore, the FIFO file must "import" that information from these two files. The system uses T/Maker's Complete command to carry out the importation. The Complete command uses the client's number (in the "cl.num" field) to match charge and payment figures to the correct client. The FIFO file's "amount.ch" and "amount.pd" fields provide destination places for these imported figures.

Figure 1: FIFO

```
<record>
CLIENT NUMBER:      {cl.num  >}
```

```
CLOSING MONTH'S FEEDER VALUES:
```

```
    Amount Charged:  {amount.ch>}
    Amount Paid:     {amount.pd>}
```

```
AGING AND ALLOCATION TABLE:
```

	Current Balances	Previous Balances
1-30 Day Balance	{amt.A >}	{amt.AP >}
31-60 Day Balance	{amt.B >}	{amt.BP >}
61-90 Day Balance	{amt.C >}	{amt.CP >}
90+ Day Balance	{amt.D >}	{amt.DP >}

```
<end>
```

Suppose, for instance, that in the month just closing Client A001 registered totals of \$100.00 in charges and \$35.00 in payments. After the COMPLETE command had been used, then, Client A001's record would hold those charge and payment amounts in the FIFO file's "amount.ch" and "amount.pd" fields.

Once they've been successfully imported, the FIFO file needs to figure out what to do with these amounts. T/Maker's SET command will carry out the necessary calculations. Consider the illustration below. In it, I have simplified the original table a little, substituting capital letters for the various fields and table cells -- this will make it easier to describe the calculations we are going to do.

Figure 2: FIFO

<record>

CLIENT NUMBER: A001

CLOSING MONTH'S FEEDER VALUES:

Amount Charged (X): 100.00

Amount Paid (Y): 35.00

AGING AND ALLOCATION TABLE:

	Current Balances	Previous Balances
1-30 Day Balance	A	E
31-60 Day Balance	B	F
61-90 Day Balance	C	G
90+ Day Balance	D	H

<end>

Allocating the Payment Amounts

Let's first consider the handling of Client A001's \$35.00 payment.

Following the "oldest first" rule, the system should first apply this payment to any existing 90+ day balance (which will be found in Cell D of the table). Let's call cell D's new value, D'. Let D' be equal to the result of subtracting the current payment, Y, from the current 90+ day balance, D. Written as a T/Maker rule, this operation can be expressed as:

D' = D - Y end [Rule 5]

D', then, represents the difference between the 90+ day balance and the current payment. If Client A001 happened in fact to have no 90+ day balance or if his actual D balance were less than the payment amount, then D' will be negative.

A negative D' value, in turn, constitutes a dollar amount that can be applied to the next younger balance, the 61-90 day balance found in cell C of the table. Let C' be the newly calculated 61-90 day balance. We can express the calculation of the C' value in the following rule:

$C' = C + D'$ when $D' < 0$ end [Rule 6]

In words, C' equals the old value of C plus D' whenever D' happens to be negative.

By the same token, the presence of a negative C' implies a money amount left over for the payment of the next younger balance, the 31-60 day balance found in cell B of the table. Let B' be the newly calculated 31-60 day balance. We can express the calculation of the B' value in the following rule:

$B' = B + C'$ when $C' < 0$ end [Rule 7]

Finally, exactly the same reasoning permits us to calculate the new value for cell A, A' :

$A' = A + B'$ when $B' < 0$ end [Rule 8]

In all, then, we now have four rules:

$D' = D - Y$ end	[Rule 5]
$C' = C + D'$ when $D' < 0$ end	[Rule 6]
$B' = B + C'$ when $C' < 0$ end	[Rule 7]
$A' = A + B'$ when $B' < 0$ end	[Rule 8]

These serve to allocate any payment to the oldest balance first, and then the next younger, and next younger, and so on. We need to clean up a remaining difficulty, though. As it now stands, the system will report negative balances in the FIFO table. Say, for example, that Client A001 had a 90+ day balance of \$45.00 and made a payment of \$90.00. The system would generate a new D value (i.e., a D' value) of \$-35.00. We shall have to use three more rules to convert negative D' , C' , and B' balances to blank or zero balances--we do not want to convert a negative A' value to blank or zero because the system needs to retain the fact that there is a credit balance in Client A001's account. Our rules would be:

$D' = "$ " when $D' \leq 0$ end	[Rule 9]
$C' = "$ " when $C' \leq 0$ end	[Rule 10]
$B' = "$ " when $B' \leq 0$ end	[Rule 11]

We now have a total of seven short rules that serve properly to allocate a current payment amount by an "oldest-balance-first" method. Let's turn to the problem of aging Client A001's balances.

Aging Balances

Having carried out the payment part of the problem, the FIFO field's table can be thought of as looking like the one below:

Figure 3: FIFO

<record>

CLIENT NUMBER: A001

CLOSING MONTH'S FEEDER VALUES:

Amount Charged (X): 100.00

Amount Paid (Y): 35.00

AGING AND ALLOCATION TABLE:

	Current Balances	Previous Balances
1-30 Day Balance	A'	E
31-60 Day Balance	B'	F
61-90 Day Balance	C'	G
90+ Day Balance	D'	H

<end>

To age these values the system must transport them to the next older balance category. Let's distinguish the new, aged cell values by giving them two prime marks. For example, the present 31-60 Day Balance (B') must become the new 61-90 Day Balance (call it C''). As it happens, we will need to pay special attention to the handling of the A' and D' cells.

We begin with the calculation of the D'' value. D'' is equal to the value of D' plus the value of C'. That is so because the 90+ Day Balance cumulates--that is to say, it cumulates its own previous value and the previous C' value. The appropriate rule is:

$$D'' = D' + C' \text{ end} \quad [\text{Rule 12}]$$

C'' is simply B' . The rule:

C'' = B' end [Rule 13]

The calculation of B'' is special. It is contingent on whether A' happened to be positive or negative. If A' was negative, there can be no outstanding B'' balance. Therefore B'' must be set equal to blank in this case. The rule:

B'' = " " when A' <= 0 end [Rule 14]

If A' was positive--that is, showed a positive outstanding balance--then B'' is simply made equal to A'. The rule:

$B'' = A'$ when $A' > 0$ end [Rule 15]

The determination of the A'' value also needs special consideration. If the A' value were positive, then -- as we just saw -- that positive value must be transported to the B cell and the A'' value, in turn, must be set equal to amount X, the total amount the client charged in the closing month. The rule:

$A'' = X$ when $A' \geq 0$ end [Rule 16]

If, however, the A' value is negative, then the client had a credit balance in the 1-30 Day Balance cell. This A' figure, therefore, should be added to amount charged in the closing month (i.e., amount X). The sum of these two amounts (A' and X) becomes the new A'' value. The rule:

$A'' = A' + X$ when $A' < 0$ end [Rule 17]

In all, then, we have a total of six rules that properly age the FIFO table:

$D'' = D' + C'$ end	[Rule 12]
$C'' = B'$ end	[Rule 13]
$B'' = "$ " when $A' \leq 0$ end	[Rule 14]
$B'' = A'$ when $A' > 0$ end	[Rule 15]
$A'' = X$ when $A' \geq 0$ end	[Rule 16]
$A'' = A' + X$ when $A' < 0$ end	[Rule 17]

It is essential that these rules be applied in the order specified--in other words, determine D'' first, determine C'' second, determine B'' third, and determine A'' fourth and last.

The "Previous Balances" Column of the Table

So far we have concerned ourselves only with the left column of the Aging and Allocation Table. The right column of the table -- the column containing cells E, F, G, and H -- is intended as a reminder of the previous month's balances. The reason for this column is simple. The column permits the user to jog his memory or check the accuracy of the system's calculations. The "Previous Month" figures combined with the "Amount Charged (X)" and "Amount Paid (Y)" permit the user to calculate on his own the correct correct "Current Balances."

These rules are quite simple. At the outset of the Rules Definition you simply provide the following four rules:

$E = A$ end	[Rule 1]
$F = B$ end	[Rule 2]
$G = C$ end	[Rule 3]
$H = D$ end	[Rule 4]

These will, of course, make cells E, F, G, and H the remembrance of cells A, B, C, and D.

Putting It All Together!

We now have 17 rules for carrying out the aging and payment allocation tasks. I have used primes and prime-prime designations in the descriptions of these rules (for example, A' and A'') to make clear the changing values in a given cell of our table. As it happens, however, T/Maker's SET command does not require the prime designations--it will happily go through a sequence of rules employing new values for a given field as such new values are calculated by preceeding rules. Therefore, we can express our 17 rules in a single Rules Definition without the need for primes:

<rules>	
E = A end	[Rule 1]
F = B end	[Rule 2]
G = C end	[Rule 3]
H = D end	[Rule 4]
D = D - Y end	[Rule 5]
C = C + D when D < 0 end	[Rule 6]
B = B + C when C < 0 end	[Rule 7]
A = A + B when B < 0 end	[Rule 8]
D = D - Y end	[Rule 5]
C = C + D when D < 0 end	[Rule 6]
B = B + C when C < 0 end	[Rule 7]
A = A + B when B < 0 end	[Rule 8]
D = " " when D <= 0 end	[Rule 9]
C = " " when C <= 0 end	[Rule 10]
B = " " when B <= 0 end	[Rule 11]
D = D + C end	[Rule 12]
C = B end	[Rule 13]
B = " " when A <= 0 end	[Rule 14]
B = A when A > 0 end	[Rule 15]
A = X when A >= 0 end	[Rule 16]
A = A + X when A < 0 end	[Rule 17]
<end>	

It bears mentioning, as well, that T/Maker is usually ill-served with field names that consist of only a single letter. Therefore, it is well that we return to the field names described in Figure 1 of this article. Following good T/Maker procedure, we will also want to (1) create an economical record definition for the this file, (2) format final numerical values, (3) specify the appearance of exactly-zero values in fields, and (4) add a "<here>" line to the file. All together, then, our finished FIFO file might look something like this:

Figure 4: FIFO

```
<form>
CLIENT NUMBER:      {cl.num  >}
```

CLOSING MONTH'S FEEDER VALUES:

```
Amount Charged:  {amount.ch>}
Amount Paid:     {amount.pd>}
```

AGING AND ALLOCATION TABLE:

	Current Balances	Previous Balances
1-30 Day Balance	{amt.A >}	{amt.E >}
31-60 Day Balance	{amt.B >}	{amt.F >}
61-90 Day Balance	{amt.C >}	{amt.G >}
90+ Day Balance	{amt.D >}	{amt.H >}

```
<end>
<rules>
amt.E xx,xxx.xx " " = amt.A end [Rule 1]
amt.F xx,xxx.xx " " = amt.B end [Rule 2]
amt.G xx,xxx.xx " " = amt.C end [Rule 3]
amt.H xx,xxx.xx " " = amt.D end [Rule 4]
amt.D = amt.D - amount.pd end [Rule 5]
amt.C = amt.C + amt.D when amt.D < 0 end [Rule 6]
amt.B = amt.B + amt.C when amt.C < 0 end [Rule 7]
amt.A = amt.A + amt.B when amt.B < 0 end [Rule 8]
amt.D = " " when amt.D <= 0 end [Rule 9]
amt.C = " " when amt.C <= 0 end [Rule 10]
amt.B = " " when amt.B <= 0 end [Rule 11]
amt.D xx,xxx.xx " " = amt.D + amt.C end [Rule 12]
amt.C xx,xxx.xx " " = amt.B end [Rule 13]
amt.B = " " when amt.A <= 0 end [Rule 14]
amt.B xx,xxx.xx " " = amt.A when amt.A > 0 end [Rule 15]
amt.A xx,xxx.xx " " = amount.ch when amt.A >= 0 end [Rule 16]
amt.A xx,xxx.xx " " = amt.A + amount.ch when amt.A < 0 end [Rule 17]
<end>
<record>
1 {cl.num  >}{amount.ch>}{amount.pd>}
2 {amt.A  >}{amt.E  >}{amt.B  >}{amt.F  >}
3 {amt.C  >}{amt.G  >}{amt.D  >}{amt.H  >}
<end>
<here>
```

All you need do now to use this file is:

- (1) Import the correct "Amount Charged" ("amount.ch") and "Amount Paid" ("amount.pd") values from their origin files with T/Maker's COMPLETE command. (Keep in mind that the FIFO file must hold a client number--"cl.num"--in order for these amounts to be brought to correct record in the file.)
- (2) Use T/Maker's SET command to--
 - (a) allocate a current payment to the correct balances, and
 - (b) age the newly calculated balances one month.

Don't forget to SAVE the file once a cycle has been run through. Good luck!

DOUBLE-SPACING AND TRS80 MODEL 4/4P MACHINES

-- Royal Farros

Because TRS-80 Printers are non-standard, some Tandy printer owners may be experiencing double-spacing problems. Try making the following changes to the current TMODIFY portion of T/Maker.

From the DOS prompt, type: TMODIFY

Choose OPTION #2

Scroll through these Options (by hitting ENTER) until you get to . . .

SPARE 1 (CURRENTLY:) ?

Type one of the following values for the appropriate printer-type.

- 0 for all printers other than TRS printers.
- 1 for all TRS-80 DMP (Dot Matrix Printers) Series.
- 2 for all TRS-80 DWP (Daisy Wheel Printers) Series.

Next, choose OPTION #4

Hit the ENTER Key until you see . . .

LINE FEED

FIRST CONTROL BYTE (CURRENTLY: 10) ?

To use a TRS-80 printer, make following changes to the first two Bytes:

FIRST CONTROL BYTE	(CURRENTLY: 10) ?	1
SECOND CONTROL BYTE	(CURRENTLY: 0) ?	10
THIRD CONTROL BYTE	(CURRENTLY: 0) ?	0
FOURTH CONTROL BYTE	(CURRENTLY: 0) ?	0
NULLS TO SEND	(CURRENTLY: 0) ?	0

(DMP-120 owners should also set TOP OF FORM to 0 in Option #4. Start each document with .PAGESIZE 66 to make your printer print full pages correctly.)

Continue hitting the ENTER Key until you are back at the TMODIFY MENU.
Finally . . . Choose OPTION #99 to save new printer specs.

Make sure the following TRS-80 Printer "Dip Switches" are selected:

CR LF (All other Switches should be in the OFF Position.)

USING "DO" LINES - CREATING SYSTEMS WITH T/MAKER

-- *Royal Farros*

Using "DO" lines can be valuable in cases where you repeat the same series of commands over and over again.

A "DO" line is a series of T/Maker commands that is saved and can be executed many times.

The Top Line of the screen (the one with commands on it) is executed by typing at the WHAT NEXT? prompt: DO <enter>

For example, suppose you are the Club Correspondent in charge of sending out personalized monthly membership newsletters. You've already created the generic newsletter (in a file called "LETTER"), and you've been keeping a membership listing database (in a file called "MEMBERS").

We could now create a file called "NEWS.DO" that has the following series of commands on the first line of the file:

(Example "DO" file called "news.do")

```
GET letter SELECT members END PRINT NONSTOP IT
```

This is what will happen if we type at the WHAT NEXT? prompt:

```
GET news.do DO <enter>
```

T/Maker will get the file called "NEWS.DO" and execute the first line in this file. The "DO" Line in this file will:

- 1) GET the file called "letter"
- 2) SELECT from the file "members"
- 3) PRINT a personalized newsletter to the Club Membership.

Of course, you don't have to put a "DO" line in a separate file. For example, this "DO" line could have gone at the top of the file called "LETTER".

In fact, a DO line can be anywhere in a file (as long as it is on its own line!). To execute a DO line that happens to be the 5th line of the file, simply type at the WHAT NEXT? prompt: 5 DO <enter>

T/MAKER T/IPS

-- *Royal Farros*

PRINTING MULTIPLE COPIES

Ever need to print something more than once? Here's a good printing tip:

Suppose we'd like to reproduce the same mailing label 500 times. If we put our mailing label in a file called LABEL and include a ".continue" command at the end of it, printing our label 500 times will be easy.

(File called "LABEL")

```
.pagesize 6
GM Nameplate, Inc.
10 Francestown Turnpike
Mount Vernon, NH 03057
.new
.con LABEL
```

To start producing labels, insert 500 labels into a printer, and at the T/Maker WHAT NEXT? prompt, type: PRINT NONSTOP LABEL. When you see the page number on the screen pass 500, use your "kill" key to stop the printing (the ESC key or CTRL Q, depending on your hardware.)

A DATABASE TIP: SHORT DATA FIELDS WITH LONG NAMES:

T/Maker has the ability to do a "short data field" if you place a #-sign in the data field to signify the desired length.

For example, suppose you want to allow only a two-character entry space for the database field STATE. (You only want people to entry CA for California, NY for New York, etc.)

You could do it this way:

```
<Record>
01 {#STATE}
<End>
```

Suppose you wanted to allow a one character entry space. You could do this this way:

<Form>

Check the appropriate box.

Are you married? {##MARRIED}

<End>

<Record>

01 {##MARRIED}

<End>

If you wanted a five-character entry space for a zip code, you could use the following:

<Record>

01 {ZIP#CODE}

<End>

Using the above concept, we could do a two-character date field like this:

<Record>

01 {#MONTH}/{#DAY}/{#YEAR}

<End>

T/MAKER MULTI-USER TIPS

Stuart Goldberg of Market Vision (creator of T/Maker Travel Agent Templates) passed along this tip to people who are involved in Multi-user situations.

Suppose you are working in a travel agency and have three phone-operators logging in travel agendas.

A problem arises when all three operators try to use the same "daily" file to enter in orders. Because T/Maker is essentially for single-users, each entry file is updated separately. The problem is the last person who saves his or her work overrides (as in destroys) the work of the other two operators. Ugggh!

Suppose, however, that each operator used a separate file for daily entries - for example, "daily1" , "daily2" , and "daily3".

At the end of the "daily1" file, place a `.continue <daily2> .`

At the end of the "daily2" file, place a `.continue <daily3> .`

At the end of the day, all three files can be summarized into a single master-daily file - maybe called "daily".

The following "DO" line (see DO LINE tips above) can be placed in a special file called "summary.do".

```
GET daily SELECT daily1 END
```

Other Multi-user Tricks:

Remember, the PATH and MIX commands can "segment & hide" files that shouldn't have immediate access. Each user can have his or her own subdirectory - and within the subdirectory, files can be jumbled (rendering them unreadable) to assure further security.

And Getting Tricky -- Creating a System!

You could automate a system that would divide the three entry files into airlines company categories - say daily totals for PanAm, TWA, American, and Others.

You can create files called "panam", "twa", "united", and "others" that contain database information about each of these airlines.

You can then create a file called "airline.do" that looks like the following:

File called "airline.do"

```
GET panam  SELECT daily1 WHEN AIRLINE = "PAN AM" END SAVE GET airline.do 2
GET twa    SELECT daily1 WHEN AIRLINE = "TWA"     END SAVE GET airline.do 3
GET united SELECT daily1 WHEN AIRLINE = "UNITED"  END SAVE GET airline.do 4
GET others SELECT daily1 WHEN AIRLINE = "OTHERS"  END SAVE GET airline.do
```

To categorize our airline activity at the end of a day, we simply type at the WHAT NEXT? prompt: `GET airline.do DO <enter>`

MONTHLY BILLING SYSTEM

-- Phyllis A. Campbell

I run a bookkeeping operation for clients with varied needs. I found that many of my clients' needs went beyond straightforward bookkeeping, however. Many needed a system in place for consistent and timely billing. Using T/Maker I created a monthly billing system which seems to fulfill their needs quite well.

The objective of the system was to have a database with a series of forms that, in one run, will produce a complete customer billing operation, including posting sheet, invoices (with sequential numbering automatically done), mailing labels, and monthly work sheet.

To create such a system, it is important to establish the forms you will need, and to ensure that the forms contain all the necessary data for your recordkeeping needs. The following example was designed for a service business, and the database is used for several of our customers, all of whom are also service businesses.

First create a file named WESDATA. The first line in the wesdata file creates a file named WESJUST. This file is the result of set and keeps the original data file in its pure form.

Two corrections to the rules: 1) "Period" should be changed to current date 2) "in" should be changed to last invoice number used.

The File WESDATA

```
delete wesjust rename wesjust set s get wespost do
<form>
Invoice:      {in}
Company name: {co.name          <} }}
Db Name:      {dba              <} }}
Name:         {name             <} }}
Street:       {street           <}
City:         {city             !} state:{s#t<} zip:{zip<}
Phone Number: {ac }{phone       <}
Professional: {pro.serv.        <} }}
Description:   {adisc            <} }}
Description:   {bdisc            <} }}
Description:   {cdisc            <} }}
Explanation:   {aexplan          <} }} Explan.amt:{da>#}{aamt  >} }
Explanation:   {bexplan          <} }} Explan amt:  {bamt    >} }}
Explanation:   {cexplan          <} }} Explan amt:  {camt    >} }}
Explanation:   {dexplan          <} }} Explan amt:  {damt    >} }}
Explanation:   {eexplan          <} }} Explan amt:  {eamt    >} }}
```

-- this file continued on next page

-- this file continued from prior page

```
Period:          {period          <> }}Period amt:{dp>#}{pamt  >} }}
Total Amount:{dt>#}{total  >} }}
Sub total: {subtot} }}
Under line: {under  >} }}      Double line: {double  } }}
<end>
<rules>
in = number + 2735
When ac = " "
    ac = "(415)"
When pamt > 0
    double = "====="
    pro.serv. = "For professional maintenance"
    dp = "$"
When period = month
    period = "For the month of June:"
When aamt >= 1
    total 9,999.99 = aamt+bamt+camt+damt+eamt+pamt
    da = "$"
    dp = " "
When total > 0
    under = "          "
    double = "====="
    pro.serv. = "For professional maintenance"
    dt = "$"
<end>
<record>
a {in}
b {co.name          <> }}
c {dba              <> }}
d {name             <> }}
e {street           <> }
f {city              !} {s#t<} {zip<}
g {ac }{phone       <> }
h {pro.serv.        <> }}
i {adisc             <> }}
j {bdisc             <> }}
k {cdisc             <> }}
l {aexplan           <> }}      {da>#}{aamt  >} }}
m {bexplan           <> }}      {bamt  >} }}
n {cexplan           <> }}      {camt  >} }}
o {dexplan           <> }}      {damt  >} }}
p {eexplan           <> }}      {eamt  >} }}
q {period            <> }}      {dp>#}{pamt  >} }}
r {dt>#}{total  >} }}
t {under  >} }}      {double  } }}
<end>
```

-- this file continued on next page

-- this file continued from prior page

a		
b	POTS 'N PANS	
d	Mary Mason	
e	448 Main Street, Suite 29	
f	Novato, CA 94947	
g	894-4111	
q	month	240.57
a		
b	KITCHEN REBUILDERS	
d	Bill Davies	
e	333 Fourth Street	
f	San Rafael, CA 94901	
g	456-8475	
i	Waxed showroom floor in	
j	addition to normal maintenance	
q	month	275.00
a		
b	CAKES AND COOKIES	
d	David Vaughn	
e	27 F Grant Ave.	
f	Novato, CA 94947	
g	894-3902	
l	June 6	50.00
m	June 13	50.00
n	June 20	50.00
o	June 27	50.00

This is what the WESJUST file looks like.

```
..This is data file for WESTERN MAINTENANCE
.. Orginal input MAY 1984
.. last update July 1, 1985
<form>
Invoice:      {in}
Company name:{co.name      <} }}
Db a Name:    {dba         <} }}
Name:         {name        <} }}
Street:       {street      <}
City:         {city        <}
Phone Number:{ac }{phone   <}
!} state:{s#t<} zip:{zip<}
```

-- this file continued on next page

-- this file continued from prior page

```

Professional:{pro.serv.          <} }}
Description: {adisc              <} }}
Description: {bdisc              <} }}
Description: {cdisc              <} }}
Explanation: {aexplan            <} }} Explan.amt:{da>#}{aa
Explanation: {bexplan            <} }} Explan amt:  {bamt
Explanation: {cexplan            <} }} Explan amt:  {camt
Explanation: {dexplan            <} }} Explan amt:  {damt
Explanation: {eexplan            <} }} Explan amt:  {eamt
Period:      {period             <} }}Period amt:{dp>#}{pam
Total Amount:{dt>#}{total    >} }}
Sub total: {subtot} }}
Under line: {under    >} }}      Double line: {double    } }}
<end>
<rules>
in = number + 2735
When ac = " "
    ac = "(415)"
When pamt > 0
    double = "======"
    pro.serv. = "For professional maintenance"
    dp = "$"
When period = month
    period = "For the month of June:"
When aamt >= 1
    total 9,999.99 = aamt+bamt+camt+damt+eamt+pamt
    da = "$"
    dp = " "
When total > 0
    under = "          "
    double = "======"
    pro.serv. = "For professional maintenance"
    dt = "$"
<end>
<rules>
CHANGED = " "
<end>
<record>
a {in}
b {co.name          <} }}
c {dba              <} }}
d {name             <} }}
e {street           <}
f {city              !} {s#t<} {zip<}
g {ac }{phone       <}
h {pro.serv.        <} }}

```

-- this file continued on next page

-- this file continued from prior page

```

i {adisc          <} }}
j {bdisc          <} }}
k {cdisc          <} }}
l {aexplan        <} }}      {da>#}{aamt    >} }}
m {bexplan        <} }}      {bamt    >} }}
n {cexplan        <} }}      {camt    >} }}
o {dexplan        <} }}      {damt    >} }}
p {eexplan        <} }}      {eamt    >} }}
q {period         <} }}      {dp>#}{pamt    >} }}
r {dt>#}{total    >} }}
t {under          >} }}      {double   } }}
<end>
a 2736
b POTS 'N PANS
d Mary Mason
e 448 Main Street, Suite 29
f Novato, CA 94947
g (415)894-4111
h For professional maintenance
q For the month of June:          $      240.57
t =====
a 2737
b KITCHEN REBUILDERS
d Bill Davies
e 333 Fourth Street
f San Rafael, CA 94901
g (415)456-8475
h For professional maintenance
i Waxed showroom floor in
j addition to normal maintenance
q For the month of June:          $      275.00
t =====
a 2738
b CAKES AND COOKIES
d David Vaughn
e 27 F Grant Ave.
f Novato, CA 94947
g (415)894-3902
h For professional maintenance
l June 6          $      50.00
m June 13         50.00
n June 20         50.00
o June 27         50.00
r $      200.00
t =====

```

The third file called WESPOST is a posted spread sheet of all the invoices in wesjust. The "do" has been put on two lines to allow you to check for input errors in billing amounts. If errors have been made, make corrections in WESDATA file. REMEMBER while in the edit mode to change date of form. When correct, you must type "do" after "What Next?".

```

select wesjust set compute e
print it g wesform do
..CHECK HERE FOR ERRORS
..first set up May 1984
..This file is Monthly POSTING PAGE for WESTERN MAINTENANCE
<RULES>
total = aamt+bamt+camt+damt+eamt+pamt
aamt = " "
bamt = " "
camt = " "
damt = " "
eamt = " "
pamt = " "
<END>
<RECORD>
+      } { in}      {co.name      <}this line continued on next
    {total  >}{aamt>    }{bamt  >}{camt  >}{damt  > }{eamt  > }{pamt  >
<END>
.8
.indent 6
.length 64
.. Spread sheet

```

	WESTERN MAINTENANCE	JUNE 1985
	=====	
ex	\ INVOICE CLIENT'S NAME	TOTAL \
zv		9,999.99
..	(place total and example so they line up with total in record definition	
	<HERE>	
=	TOTAL	715.57
		=====

The fourth file is named WESFORM and is the billing invoice. REMEMBER while in the edit mode to change dates on invoice then type "do" after "What Next?".

```
Select wesjust e
print it g wesenv do
..This is billing invoice form for WESTERN MAINTENANCE
..Original set-up MAY 1985
..update June 1985
.3
.indent 8
.length 39
.pagelength 42
<record>
.new
```

```
WESTERN MAINTENANCE
250 San Marin Drive
Novato, CA 94947
(415) 894-4873
```

```
07/01/85 { in}
```

```
{co.name          <}
{name             <} }}
{street           <} }}
{city             !} }}{s#t<}}}{zip}}}
```

```
{pro.serv.        <} }}
```

```
{adisc            <} }}
{bdisc            <} }}
{cdisc            <} }}
{aexplan          <} }}
{bexplan          <} }}
```

```
{da>#}{ aamt      >} }}
      {bamt       >} }}
```

-- this file continued on next page

-- this file continued from prior page

```
{cexplan          <> }}          {camt      >} }}
{dexplan          <> }}          {damt      >} }}
{eexplan          <> }}          {eamt      >} }}
{period           <> }}          {dp>#}{ pamt  >} }}
                                   {under    >} }}
                                   {dt>#}{ total  >} }}
                                   {double    >}

<end>
.bottom

                                DUE AND PAYABLE JULY 10, 1985

.end
<HERE>
```

The fifth file is named WESENV. It produces the mailing label.

```
select wesjust print it g weswork do
..This file is mailing list for WESTERN MAINTENANCE
..Original set-up May 1984
..MAKES * MAILING LABELS *
..Label size 3-1/2" X 15/16"
<record>
.new

{name              <>
{title             <> }}
{street            <>
{city              !}{s#t<}{zip}
<end>
.length 6
.pagesize 6
.indent 2
.l
<HERE>
```

The sixth file is named WESWORK. This is a work sheet given to the billing manager each month to record work and amounts to be billed for and any changes needed in the data base for the next months billing. REMEMBER while in the edit mode to change date then type "do" after "What Next?". (File is on next page.)

```

select wesjust compute e
print it
..first set up May 1984
..This file is Monthly Work Sheet for WESTERN MAINTENANCE
<RECORD>
+      ){co.name          <}
      {name              <}
      {street            <}
      {city              <}{s#t<}{zip<}

<END>
.new
.6
.indent 6
.clean
.length 64
.. monthly work sheet
.number 1
.top

```

```

          |WESTERN MAINTENANCE|          |JUNE 1985|
          =====
.end      \ CLIENT'S NAME                \    \ PERIOD COVERED
          \                               \
          \ |WORK SHEET|\

```

```

          |WESTERN MAINTENANCES|          |JUNE 1985|
          =====
          \ CLIENT'S NAME                \    \ PERIOD COVERED
          \                               \
          \                               \

ex
zv

..      <HERE>

=      TOTAL

```

Form Letter Generation

This page shows the result! Next two pages show the files used to generate this letter.

ROBERT ACKERMAN
1101 Schooner Street - Foster City, CA 94404
415-573-1807

August 27, 1985

Heidi Roizen
T/Maker Company, Attn: T/MUG
2115 Landings Drive
Mountain View, CA 94043

Dear Heidi,

This is still one more sample of a form letter. As you know, I am a staunch believer of the power of T/Maker's database functions. The strength of my conviction is partly due to the fact that I never learned how to use LOAD and UNLOAD. At any rate, there are two files, SAMPLE.FIL, containing names, etc., and SAMPLE.LTR, containing the text of the letter.

Basic trick is that the DO lines in the letter select all the names out of the database, get rid of extraneous material, and print without intervention. This letter was written with the routine. Notice the insertion of your first name and comments in the body of the letter.

It is always a pleasure to talk to the President of a company involved in software publishing. I think you run a great company with a terrific staff!

Something people may forget: DON'T ALIGN the form letter before the SELECTION. That will remove all the spaces between the brackets that you so carefully put in the right places. Also, include the .NEWPAGE command just before the end of the record, or the individual letters will not be paged.

Best regards,

The file SAMPLE.LTR

```
SELECT SAMPLE.FIL END ALIGN DO
FIND <HERE> CLIP B DROP <HERE> DELETE TEMP RENAME TEMP SAVE PRINT IT
<RECORD>
.indent 8
.top
<<-
```

>>>

ROBERT ACKERMAN
1101 Schooner Street - Foster City, CA 94404
415-573-1807

>><<

August 27, 1985

```
.end
{!FIRST      } {!LAST          }
{COMPANY      }
{ADDRESS      }
{!CITY        }, {#ST} {ZIP}
```

Dear {!FIRST },

<<<

->>

This is still one more sample of a form letter. As you know, I am a staunch believer of the power of T/Maker's database functions. The strength of my conviction is partly due to the fact that I never learned how to use LOAD and UNLOAD. At any rate, there are two files, SAMPLE.FIL, containing names, etc., and SAMPLE.LTR, containing the text of the letter.

Basic trick is that the DO lines in the letter select all the names out of the database, get rid of extraneous material, and print without intervention. This letter was written with the routine. Notice the insertion of your first name and comments in the body of the letter.

It is always a pleasure to talk to the {!JOB } of a company involved in {!ACTIVITY }.
{COMMENT

Something people may forget: DON'T ALIGN the form letter before the SELECTION. That will remove all the spaces between the brackets that you so carefully put in the right places. Also, include the .NEWPAGE command just before the end of the record, or the individual letters will not be paged.

Best regards,
.new
<END>
<HERE>

The file SAMPLE.FIL

<FORM>

```
FIRST      :{FIRST      }:  
LAST       :{LAST       }:  
COMPANY    :{COMPANY    }:  
ADDRESS    :{ADDRESS    }:  
CITY       :{CITY       }:  
STATE      :{#ST}:  
ZIP        :{ZIP}:  
  
JOB         :{JOB         }:  
ACTIVITY    :{ACTIVITY    }:  
COMMENT     :{COMMENT
```

<END>

<RECORD>

```
1{FIRST      }  
2{LAST       }  
3{COMPANY    }  
4{ADDRESS    }  
5{CITY       }  
6{#ST}  
7{ZIP}  
8{JOB        }  
9{ACTIVITY   }  
0{COMMENT    }
```

<END>

<HERE>

```
1Heidi  
2Roizen  
3T/Maker Company, Attn: T/MUG  
42115 Landings Drive  
5Mountain View  
6CA  
794043  
8President  
9software publishing  
0I think you run a great company with a terrific staff!
```

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

TMUG

TMMAKER USERS GROUP
TMMAKER COMPANY
2115 LANDINGS DRIVE
MOUNTAIN VIEW, CA 94043

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