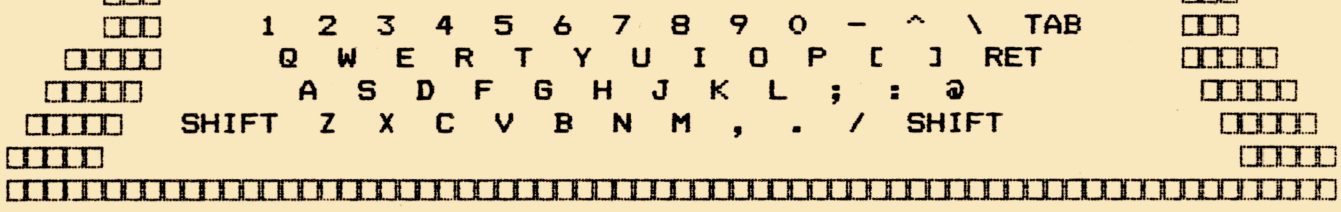
T/MUG NEWSLETTER No. 6 March 1, 1983

IN THIS ISSUE:



T/Maker Users Group, 2813 Flagmaker Drive, Falls Church, VA 22042

DEAR READERS,

Many of you may have been wondering where your November/December and January/February issues went. I wish I could say they were destroyed in our warehouse fire, but we do not have a warehouse and did not have a fire. Quite simply, they never got written. As a result, all subscriptions will automatically be extended two issues.

Up until now, Nabil Issa, the editor, has been kind enough to produce the Newsletter in his spare time, but most good things must come to an end. T/Maker Company will thus start producing the Newsletter itself. The publication will move out to our California address with the next issue. If those of you requesting back issues could hold off a month or two while we get reorganized, we would appreciate it. Right now, there are no back issues remaining.

I should warn you that the new Editor/Manager will be an MBA from Stanford, so you will probably see subscription fees skyrocket to a point where they may even cover the costs.

Peter Roizen

T/Maker III

T/Maker III is out. Needless to say, this program bears a strong resemblence to one you own. It is available as an update for approximately $100 from (among others):

Computer Potentials	**Santa Clara, Calif.**	**(408) 980-9100**
Lifeboat Associates	**New York, N.Y.**	**(212) 860-0300**
WESTICO Inc.	**Nowalk, Conn.**	**(203) 853-6880**

To get an update, you will have to return your orginal disk as proof of purchase. Please phone any of the above first to get accurate information on pricing, procedures, and disk formats available.

The price includes a complete new set of documentation which was considerably rethought and rewritten. We believe it represents a big improvement over the T/Maker II manual you are holding. The Quick Refernce Card has been replaced by a Quick Reference Booklet. It is printed on thick stock with colored diagrams and specially cut pages that form tabs. Everyone who worked on it is very proud of the result.

The program has undergone a number of enhancements. The majority of these fall under the heading of word processing. T/Maker III incorporates such features as:

- -facilities for indented and numbered paragraphs
- -underlining and boldface
- -single and double spacing
- -automatic page numbering
- -placement of repeated headings and page-endings
- -placement of footnotes
- -facilities for stringing together many files to form a long document
- -preview printing on the screen, and
- -the possibility to print only specific pages from a document

We believe that you will find the result competitive with other word processors. And, of course, the commands that control these options can be put in tables, as well, to achieve the same sort of result.

One simple, but very handy feature is a command that automatically "cleans" a table while printing it without changing the file itself. If you use T/Maker with any regularity, we think the update is well worth the price. The new manual and/or booklet can be purchased separately if you want to study the matter more before making a decision.

In this newsletter, we will use T/Maker III notation and terminology. Some of you may have already noticed the "print it" commands in the previous issue. Also, you may hear us talk about "calculation strips" instead of "row equations" or "aligning wedges" instead of "margin markers." We think the new vocabulary helps make the underlying concepts easier to understand.

Though not available as an update, T/Maker III can also be purchased for CP/M 86, MS DOS, or the IBM using PCDOS. These systems offer 44,000 character working files and the possibility of constructing tables with up to 50 columns instead of 25.

Under these operating systems, you can go from Compute to Edit to Compute to Edit, etc. without any delays resulting from a disk operation. This speeds things up considerably when you are trying out a series of "What If" options.

If you are an addicted T/Maker user and are thinking of getting another computer, it would make sense to consider a 16-bit machine.

Also on the subject of hardware, if you attended the CP/M 83 show in San Francisco, you may have seen a small box for sale that incoorporated one 390K disk drive, 128K of memory, CP/M, two serial ports, and a free copy of T/Maker III for the show-only price of $500! The regular price is around $1,000. Just in case the thought might cross your mind, you can not get the package without T/Maker for $275 less; we already asked. For more information, contact PMC Inc. in Mt. View, California at (415) 962-0220.

ANALYZING SURVEY DATA WITH T/MAKER
Part I: Setting Up the Data Set

Note: This is the first in a series of articles on how to analyze survey data with T/Maker. In this one we show how to set up a survey data set in a form that is convenient for T/Maker's analytical abilities.

A questionnaire survey is just a specified set of questions asked of a specified set of people. Each person's completed questionnaire is called a "case." Each case should be assigned a unique number--a "case number" that differentiates it from all other cases. Case numbers should run from 001 to the final case in the sample. Each case number, of course, will have that case's data associated with it.

The data derive from the person's responses to the survey questions. Let's call each survey question a "variable," because the answers people give vary. There are two basic kinds of survey questions: closed-ended and open-ended. Closed-ended questions, like multiple choice questions on a quiz, provide a fixed series of options that respondents must choose among. Open-ended questions, on the other hand, allow respondents to express themselves more freely, responding in a narrative way.

Open-ended questions questions can be handled in a number of ways in survey analysis. For example, you may want to quote responses directly in order to provide some of the flavor of respondents' reactions. But if open-ended questions are going to be analyzed quantitatively they must be "coded"--that means the answers must be converted into "codes" representing the various answers. In a sense, such "coding" reduces open-ended questions to the form of closed-ended questions, the main difference being that the response categories of open-ended questions are assigned after rather than **before** the question was answered.

Setting Up the Data Set

Figure 1

THE DATA SET

	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10
case 01	---	---	---	---	---	---	---	---	---	---
case 02	---	---	---	---	---	---	---	---	---	---
case 03	---	---	---	---	---	---	---	---	---	---
case 04	---	---	---	---	---	---	---	---	---	---
case 05	---	---	---	---	---	---	---	---	---	---

. .

In its simplest form, survey data can be represented by a series of lines, each line holding the data of one case, as in

Figure 1, above.

The dash-marks in Figure 1 represent the places data will be inscribed. It's a good idea, though, to introduce a number of modifications to this simple form: First of all, it will be handy (for reasons we'll see later on) to have a one-letter name for each question. This "name" simply represents the question number. It is also a good idea to put plus signs in column one and move the word "case" over to a starting point in col. 8. The data matrix will now look like this:

Figure 2

THE DATA SET

		Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10
+	case 01	A--	B--	C--	D--	E--	F--	G--	H--	I--	J--
+	case 02	A--	B--	C--	D--	E--	F--	G--	H--	I--	J--
+	case 03	A--	B--	C--	D--	E--	F--	G--	H--	I--	J--
+	case 04	A--	B--	C--	D--	E--	F--	G--	H--	I--	J--
+	case 05	A--	B--	C--	D--	E--	F--	G--	H--	I--	J--
+	.										
+	.										
+	.										

Another modification is to allot dash-spaces according to the demands of the question actually asked and its response codes. For example, if you ask a respondent's gender you will only need one "dash-space" to "code" the answer: that column will contain either an "m" for male or an "f" for female. But if you ask a person's exact income it will be necessary to allot several dash spaces for the answer. Figure 3 reflects an hypothetical data set that has been apportioned this way:

Figure 3

THE DATA SET

		Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10
+	case 01	A-	B------	C--	D-	E-	P-	G-	H-	I-	J--
+	case 02	A-	B------	C--	D-	E-	P-	G-	H-	I-	J--
+	case 03	A-	B------	C--	D-	E-	P-	G-	H-	I-	J--
+	case 04	A-	B------	C--	D-	E-	P-	G-	H-	I-	J--
+	case 05	A-	B------	C--	D-	E-	P-	G-	H-	I-	J--
+	.										
+	.										
+	.										

The third, and final, modification is the addition of a column ruler. It will be convenient for running tables later on:

Figure 4

```
                                        THE DATA SET

                          Q1   Q2        Q3    Q4   Q5   Q6   Q7   Q8   Q9   Q10

0000000001111111111222222222233333333334444444444555555555566666
1234567890123456789012345678901234567890123456789012345678901234
+       case 01       A-   B------  C--   D-   E-   P-   G-   H-   I-   J--
+       case 02       A-   B------  C--   D-   E-   P-   G-   H-   I-   J--
+       case 03       A-   B------  C--   D-   E-   P-   G-   H-   I-   J--
+       case 04       A-   B------  C--   D-   E-   P-   G-   H-   I-   J--
+       case 05       A-   B------  C--   D-   E-   P-   G-   H-   I-   J--
+         .
+         .
+         .
```

We now have, in Fig. 4, the basic form of a data set conveniently organized for analysis with T/Maker. Of course, we haven't placed any real data into the matrix yet. So, just for the sake of example, let's pretend that we have carried out a survey of twenty people's attitudes towards opera and towards pro football. Suppose, too, that a number of other questions were included.

Survey Questions and Responses

Questions		Responses
A. Gender?		m-male f-female
B. Yearly income?	[write in]	_____
C. Age?	[write in]	__
D. Rent or own?		r-rent o-own
E. Education?		a-h.s. grad or less b-college grad c-grad school or more
F. Political party?		d-democrat r-republican
G. Frequency of visiting mother?		0-none in last year 1-one or two times 2-three to five times 3-six to ten times 4-eleven or more times
H. Like opera?		1-hate it 2-like it a bit 3-like it some 4-like it 5-like it a lot
I. Like pro football?		1-hate it 2-like it a bit 3-like it some 4-like it 5-like it a lot
J. Height in inches	[write in]	__

Now, let's lastly suppose that we have done our survey, coded our results, and put the findings into the matrix. The result might look something like this:

Figure 5

THE COMPLETED DATA SET

```
                          Q1  Q2        Q3   Q4  Q5  Q6  Q7  Q8  Q9  Q10

000000000111111111122222222233333333334444444444555555555566666666
123456789012345678901234567890123456789012345678901234567890123456789012345

+       case 01       Am  B 15000  C35  Dr  Ea  Fr  G1  H2  I4  J70
+       case 02       Af  B 27546  C46  Dr  Eb  Fd  G2  H2  I4  J64
+       case 03       Am  B 35123  C67  Dr  Ec  Fr  G2  H4  I4  J73
+       case 04       Am  B 22456  C33  Do  Ea  Fd  G0  H5  I1  J69
+       case 05       Af  B 57343  C22  D-  Eb  Fr  G2  H4  I1  J62
+       case 06       Af  B 67111  C25  Do  Ec  Fd  G2  H3  I1  J66
+       case 07       Am  B 11342  C56  Dr  Ea  Fr  G2  H2  I3  J70
+       case 08       Am  B109345  C47  D-  E-  Fd  G3  H4  I4  J66
+       case 09       Am  B  6700  C21  Dr  Ec  Fr  G0  H1  I5  J71
+       case 10       Af  B 19800  C26  Do  Ea  Fd  G1  H1  I2  J61
+       case 11       Am  B 34678  C54  Do  Eb  F-  G2  H1  I4  J67
+       case 12       Af  B 22333  C42  Do  Ec  Fd  G1  H1  I2  J67
+       case 13       Am  B 33222  C70  D-  Ea  Fr  G2  H4  I2  J68
+       case 14       Am  B 29999  C33  Dr  Eb  Fd  G1  H5  I2  J74
+       case 15       Af  B 30123  C51  Dr  Ec  Fr  G-  H4  I1  J59
+       case 16       Am  B 56677  C56  Do  Ea  Fd  G1  H1  I5  J70
+       case 17       Af  B  9800  C24  D-  Eb  Fr  G0  H2  I1  J67
+       case 18       Af  B 18777  C36  Do  Ec  Fd  G1  H3  I2  J63
+       case 19       Am  B 18665  C36  Do  Ea  Fr  G1  H4  I1  J62
+       case 20       Af  B------  C40  Dr  Ea  Fd  G1  H5  I4  J66
```

This matrix should be in a file of its own. We are now ready to do some analysis--the subject of the next part.

Date Conversions

Even though T/Maker does not have a special date routine to determine the number of days elapsed between two dates, it is nonetheless possible to calculate an approximation of elapsed time. In the table below the difference between two dates is calculated on the assumption that one month is equal to 1/12th of a year and one day is equal to 1/365th of a year. The elapsed years is then used to calculate the annual growth rate of an investment based on a purchase price and an appraised value.

If the elapsed time between the two dates is not too short, the results will be sufficiently accurate for most needs.

A S S E T S

		--- Purchased --			-- Last Appraisal ---				Elapsed	Annual
	Item	Mn	Dy	Yr	Price	Mn	Dy	Yr	Value	Years	Rate
ex		99	99	99	999,999	99	99	99	999,999	999.9	999.9
uc1		sta	-			stb	+			=	
uc2/		12				12				365	
uc3		-fta		-		+ftb		+		+=	
uc4					-				+	/	grw
+	House	10/	9/	76	75,600	2/	16/	83	145,000	6.4	10.8
+	Car	8/	11/	80	13,400	1/	1/	83	9,000	2.4	-15.3
+	Gold	7/	19/	82	650	1/	21/	83	420	0.5	-57.9
+	Stock	7/	23/	82	1,000	7/	23/	83	1,100	1.0	10.0

. .

MISSING DATA (One example of dealing with it)

A few persons have asked about handling "missing data" in a table. In particular, one person brought up the problem of averaging scores for students over a series of tests, when some of the students miss some of the tests. The teacher wanted to know the average percentage score for each student (i.e. the student's score divided by the possible score for the tests actually taken by the student). This problem is solved on the facing page.

Let's explain it.

Scores are entered (by convention) as "minus one" when a student did not take a test. If a student did take a test, the score is entered (i.e. zero or more). Thus, a distinciton between a score and "no data" can be made. Now, let's look at how the calculations deal with this distinction.

The first calculation, "uc1", simply "stores" the test scores so that the columns can be used for intermediate results. Next, "uc2", adds one to the value for each column. Thus, minus one becomes zero, zero becomes one, etc. The third calculation, "uc3", replaces the value of each column by the "sign" of the number in that column. Zero stays at zero; and any positive number becomes one.

The value in the each column after these calculations will be zero if the student did not take the test and one if he did. "uc4" multiplies each column by the possible score for the corresponding test. The result of this will be zero if the student did not take the test or the possible score for the test if he did. "uc5" simply sums up the columns and places the total into the last column. Thus, the last column will temporarily contain the total possible score for the student.

The original scores are next "fetched" back into the columns by "uc6". The effect of "uc7" is simply to replace negative numbers in the columns by zeros (positives only). Thus, minus one is replaced by a zero. "uc7" sums up the scores, divides by the possible score, and produces the desired result as a percentage.

Finally, "uc9" puts the original values back into the columns.

TEST SCORES

		Test 1	Test 2	Test 3	Test 4	Test 5	Test 6	Average Percent
ex		999	999	999	999	999	999	999.99
uc1		sta	stb	stc	std	ste	stf	
uc2+		1	1	1	1	1	1	
uc3		+sgn	+sgn	+sgn	+sgn	+sgn	+sgn	
uc4*		50	100	75	125	100	75	
uc5		+	+	+	+	+	+	=
uc6		fta	ftb	ftc	ftd	fte	ftf	
uc7		+ply	+ply	+ply	+ply	+ply	+ply	
uc8		+	+	+	+	+	+	/%
uc9		fta	ftb	ftc	ftd	fte	ftf	
+	John	20	-1	-1	-1	40	-1	40.00
+	Mary	-1	80	62	-1	-1	45	74.80
+	Alice	45	70	52	79	-1	52	70.12
+	Bob	10	-1	-1	-1	-1	25	28.00
+	Burt	30	50	-1	74	-1	50	58.29
+	Herbert	25	75	-1	100	50	44	65.33
	.							
	.							
	.							

T/Maker Appointment Calendar

Here is a system written entirely in T/Maker to keep your daily appointment calendar. It also shows a few useful T/Maker programming tricks. Three files are involved. I call them "a:schedule.me", "a:schedule.sho", and "a:date".

"a:schedule.me" actually contains my appointment schedule, one entry per line, in the form:

mmdd one line entry describing appointment

Thus I might have:

```
214     remember Valentine's day
727     Jane's birthday
1021    Daniel's birthday
1205    send Xmas cards early this year!
214     dev. bank report due
215     tax stuff to accountant
401     dinner w Pres
327     dentist 4pm
215     check on typesetting: David 683-9414
218     Peter lunch 1230 Clyde's re comm software
```

Notice that it isn't sorted - it's just there in whatever order things were entered. The only important fact is that there is a numeric date in the first four positions of each line.

"a:date" looks like this:

```
g a:schedule.sho do
date='2/15/83'
```

Any T/Maker program can get the date by simply doing- load a:date. "schedule.sho" does exactly that. Here it is:

```
/200 delete junk rename junk load a:date do
replace / *    * arrange 1 7 14 22 13 13 8 12 end compute clean find -END- insert a:schedule.me 1/1 sort a n 1 4 find =TODAY
ex          9999 ,,,
uc1!        100
uc2         +    +=
uc3         +    +av1
+      TODAY=((date   )
       -END- 9999
```

Each morning, the first thing I do on turning on the computer is to type- **tmaker g a:date e.** Then using the editor I can change the date as necessary. After quitting the editor I enter **save do** which saves today's date and activates a:schedule.sho to sort "TODAY=215" into the information from schedule.me and leave today's schedule at the top of the screen. With the example above, I would be left with:

```
 214    remember Valentine's day
 214    dev. bank report due
-------------------------------------------------------------
! 215     =TODAY                                             !
!215    tax stuff to accountant                             !
!215    check on typesetting: David 683-9414                !
!218    Peter lunch 1230 Clyde's re comm software           !
!327    dentist 4pm                                         !
!401    dinner w Pres                                       !
!727    Jane's birthday                                     !
!1021   Daniel's birthday                                   !
!1205   send Xmas cards early this year!                    !
!9999      -END-                                            !
!                                                           !
!                                                           !
!                                                           !
!                                                           !
!                                                           !
!-----------------------------------------------------------!
```

with the screen placed as shown.

The initial /200 effectively makes the screen go blank during processing. That aids the calmness one needs in the morning. The "junk" stuff makes sure that if I accidentally type a "save" I won't wipe out any useful files. The replace, arrange, and compute convert the date in mm/dd/yy format to mmdd format. The insert of schedule.me is done after the -END- so the TODAY indicator will sort ahead of any entries for today, as well as future entries. 1/1 moves back up to the top of the working file (and incidentally starts normal displaying again). Finally, after sorting on date, the "find" brings the TODAY indicator to the top of the screen.

Tom Moran

A Simple Invoicing System

Many business can get by with a very simple invoicing system, because they just don' issue that many of them. Here is an example of such a system developed with T/Maker.

The system would probably start with a form representing what an invoice ought to look like. An example of such a form is shown below. We'll assume it has the name "INVOICE.FRM". Of course, this invoice could be modified in many ways depending on what is most suitable for the type of business involved.

```
**********************************************************************
 File: INVOICE.FRM
**********************************************************************

.clean
        ------------------------------------------------------------
        ACME COMPANY
        ------------------------------------------------------------
        Invoice Number:
        ------------------------------------------------------------
        Name:



        ------------------------------------------------------------
        Date Shipped:

        ------------------------------------------------------------
        Item                              Quantity   Price          Total
        ------------------------------------------------------------
ex                                          999      999.99      99,999.99
zv
uc1                                          +          *              =
rc2                                         ftz        ftz

+       1.                                  ???      ??????
+       2.                                  ???      ??????
+       3.                                  ???      ??????
+       4.                                  ???      ??????
cc1                                          --
        ------------------------------------------------------------
+       Shipping                                                 ?????????
        ------------------------------------------------------------

=       Total:
        ============================================================

**********************************************************************
```

When a new invoice is needed, this file can be called up on the screen and renamed. For example, we might say:

get invoice.frm rename invoice.4

if the next invoice is to be number four. Once the form becomes the working file, it can be edited to enter the required information. Then, Compute and Print can be used to carry out the calculations and produce a copy on paper. Presumably, the file in the form below is saved on disk (entered data is shown in boldface).

```
**************************************************************************
 File: INVOICE.4
**************************************************************************

.clean
        ------------------------------------------------------------
        ACME COMPANY
        ------------------------------------------------------------
        Invoice Number: 4
        ------------------------------------------------------------
        Name: Bills Plumbing
              235 Fourth Street
              San Francisco, CA 94112

        ------------------------------------------------------------
        Date Shipped: 03/08/83

        ------------------------------------------------------------
        Item                          Quantity   Price          Total
        ------------------------------------------------------------
ex                                       999     999.99     99,999.99
zv
uc1                                        +          *              =
rc2                                      ftz        ftz

+       1. Pipes                          12      14.95         179.40
+       2. Joiners                        15       2.49          37.35
+       3.
+       4.
cc1
        ------------------------------------------ ---------------------
+       Shipping                                                  12.09
        ------------------------------------------------------------

=       Total:                                                   228.84
        ============================================================

**************************************************************************
```

A next objective might be to keep a summary file of what amounts are due. An example table is shown below which divides

the invoices into those which have been paid and those which remain due. When a customer pays a bill, the buffer in the editor can be used to move his line from the "Total Due" category to the "Total Paid" category, and the table can be recomputed to derive the new figures.

When a new invoice is issue, it is necessary to enter a new line into the table. One goal might be to make the entry automatically by using the data in the invoice. Let's see how this could be done.

```
**************************************************************************
 File: SUMMARY
**************************************************************************

        ------------------------------------------------------------
        Accounts Recievable
        ------------------------------------------------------------
        Name                          Invoice    Date Shipped      Amount
                                      Number
        ------------------------------------------------------------
ex                                                              999,999.99
+       Bradley Associates                1      10/10/82        2,549.11
+       General Products Inc.             2      11/11/82        3,876.12
        ------------------------------------------------------------
=       Total Paid                                               6,425.23
        ------------------------------------------------------------
+       Richards Company                  3       1/13/83       12,481.15
        ============================================================
=       Total Due                                               12,481.15
        ------------------------------------------------------------

**************************************************************************
```

First we need to get the necessary data out of the invoice. This can be done by using the Unload command with an approriate mask (see below). If the invoice were the working file and the command unload invoice.msk were given, the working file would be converted to the file shown as "TEMP".

```
**************************************************************************
 File: INVOICE.MSK
**************************************************************************

       Invoice Number:{inv     }

       Name:{name                        }

       Date Shipped:{date              }

=      Total:                                                {amt     }

**************************************************************************


**************************************************************************
 File: TEMP
**************************************************************************

inv = '4'
name = 'Bills Plumbing'
date = '03/08/83'
amt = '228.84'

**************************************************************************
```

The above information could be saved on disk. Now suppose we got back the summary table so that it became the working file. If we advanced the screen so that the top line was the one just above the calculation of the total due, we could insert the mask below into the file and load the data from TEMP, thus creating a new entry in the table.

```
**************************************************************************
 File: SUMMARY.MSK
**************************************************************************

 +      {<name                      }{inv> }   {date    >}     {amt>     }

**************************************************************************
```

The entire command series could itself be kept as a file called "UPDATE." When we were finished computing, printing, and saving an invoice, we would only need say-

insert update do

to automatically update the summary table.

```
**************************************************************************
 File: UPDATE
**************************************************************************

1 unload invoice.ask delete temp rename temp save get summary find ===== insert summary.ask load temp 1 compute save

**************************************************************************
```

If you use this procedure with some slow disk drives you may find it a little cumbersome. Another prossibility is to wait until you are going to disappear for a few minutes and then leave your machine with a number of invoices to do by using a command like the one below.

```
g invoice.11 insert update do g invoice.12 insert update do
```

For experts skilled in the naming conventions used in data files, it is possible to change the invoice somewhat so that the information in it can be loaded without having to use the Unload command and saving the resulting data. The invoice below is a data file as well as an invoice.

```
.clean
        ------------------------------------------------------------
        ACME COMPANY
        ------------------------------------------------------------
        Invoice Number=
        ------------------------------------------------------------
        Name='


        ------------------------------------------------------------
        Date Shipped=
        ------------------------------------------------------------
        Item                          Quantity   Price          Total
        ------------------------------------------------------------
ex                                      999      999.99      99,999.99
zv
uc1                                      +          *              =
rc2                                     ftz        ftz

  +     1.                              ???      ??????
  +     2.                              ???      ??????
  +     3.                              ???      ??????
  +     4.                              ???      ??????
cc1
        ------------------------------------------------------------
  +     Shipping                                          ????????
        ------------------------------------------------------------
+=      Total:
        ============================================================
```

The names of the data items would be:

Invoice Number
Name
Date Shipped
+1 (the word "Total:" would be the data named "+")