THE WOMBAT
EXAMINER

"INCREASES THE CIRCULATION OF ANYONE IN AMERICA"   Vol. 4 No. 3

DECUS

November 1982

Coming Attractions:

One Night Only

WOMBAT MAGIC

IN *

Anaheim

California

...land Hotel

BZL

A REFERENCE TOOL FOR DATATRIEVE

California Fish and Game's Susan Dakuzaku will describe a quick reference guide that she developed using Warnier-Orr techniques.

SIG BUSINESS MEETING

The main topics will cover the SIG's operating principles and approval of changes in SIG leadership. Prospective new members welcome. Bring your ideas.

RECORD DEFINITION WORKSHOP

The indomitable Gary Saxer, AXXA Corp., will star again in this workshop, offering clever and useful record definition techniques. If you didn't laugh last time, try him again. Session scheduled late in the day to let him run overtime in peace.

Also Note:

VAX-11 DBMS Technical Tutorial
VAX-11 DBMS Technical Panel

Wednesday, December 8:

DTR-11 and VAX-11 TECHNICAL TUTORIALS

Optimization techniques and internal design, answers to your questions by DIGITAL's Scott Matsumoto and Dan Diettrich.

DTR DEVELOPMENT AND UTILIZATION PANEL

Users on the line, presenting their Datatrieve application systems. Sid Edelman, Texas Heart Institute, will chair this well-prepared, enthusiastic panel.

WRITING REPORTS WITH VAX-11 DATATRIEVE

Use of VAX-11 special features and techniques in report writing: the CROSS, dividing records into groups, grouping data by date, and more, presented by Henry Morris of DIGITAL.

VAX-11 DTR EDITOR

Wayne Jones of DEC will show off the new EDT-based DTR editor. Don't miss it.

Also Note:

VAX-11 CDD Technical
Practical Usage of VAX-11 CDD
Fourth Generation Languages, Richard Landau of DIGITAL.

Thursday, December 9:

USER PAPERS

We have chosen three excellent formal papers for this Symposia. K. M. Richardson, 3M, will show how DTR is used in data monitoring and analysis of electronic circuit manufacturing systems. Datatrieve supports a channel communication data base for Larry Creel of Los Alamos. E. A. Haser, Westinghouse, will describe how DTR is used in software project management at his shop.

(These people are presenting; why aren't you?)

DTR-11/VAX-11 FUTURES

DIGITAL's Scott Matsumoto and Dan Diettrich will forecast their products' futures. Announcements may be made.

WOMBAT MAGIC

In the twilight hours, Magicians, apprentices, non-aligned hirelings emerge to cast awesome artifices upon the shining walls. Brew included.

Also Note:

USING DATATRIEVE IN OFFICE AUTOMATION
ON-LINE UPDATING WITH VAX-11 DBMS
CANCELLED: CUSTOMIZING VAX-11 DATATRIEVE

Friday December 10:

SIG CLOSING SESSION

Wrap-up: evaluation of current sessions, planning for St. Louis, response to Campground wish list, more.

Solving Equations in Datatrieve
B. Z. Lederman I.T.T. World Communications

This paper highlights some of the methods of solving equations by using the mathamatical, logical and statistical functions available in Datatrieve. This paper will not attempt to teach equation solving, but will highlight the facilities available in Datatrieve, demonstrate some approaches to solving problems, and will point out some of the difficulties or limitations to the process.

Datatrieve has all of the basic requirements for solving mathamatical or logical equations, which are:

(1) Mathamatical operators:
     Addition (+)
     Subtraction (-)
     Multiplication (*)
     Division (/)

(2) The ability to control the flow of calculations by logical (Boolean) operators (IF-THEN-ELSE).

(3) The ability to perform repetitively until a condition is met (FOR and WHILE).

While this may not seem to be a very large repetoire, it is enough to solve almost any equation: it is, in fact, all that any computer has, or what any person would have if the equation were to be solved by hand. Other computer languages have libraries of functions which can be called for convenience, but their basic functions are the same.

In order to illustrate the process, I will set up a sample domain and run through a series of examples. The record definition is:

```
01 SAMPLE-REC.
      03 ITEM PIC 9.
      03 A PIC 999 EDIT-STRING ZZ9.
      03 B PIC 999 EDIT-STRING ZZ9.
      03 C PIC 999 EDIT-STRING ZZ9.
      03 T1 PIC 9999 EDIT-STRING ZZZ9.
      03 T2 PIC 9(6) EDIT-STRING ZZZ,ZZ9.
;
```

The domain is SAMPLE, and is keyed by item. This very simple domain is for demonstration purposes only.

The first example will be to calculate T1 by the formula $T1 = (A+B)*C$. While this could easily be done by making T1 a COMPUTED-BY field, it is not possible to sort on a computed field, but it will be possible to sort on T1. The FOR statement will be used as it is the easiest way to perform the same calculation for every record in a domain or collection. A possible command sequence is:

```
READY SAMPLE MODIFY
FOR SAMPLE MODIFY USING BEGIN
      T1=(A+B)*C
      END
PRINT SAMPLE SORTED BY DESC T1
```

The data in the domain before these commands looks like this:

ITEM	A	B	C	T1	T2
1	3	5	7	0	0
2	7	5	3	0	0
3	2	6	4	0	0
4	7	3	4	0	0

After the commands, it looks like this:

ITEM	A	B	C	T1	T2
1	3	5	7	56	0
4	7	3	4	40	0
2	7	5	3	36	0
3	2	6	4	32	0

This is a rather trivial example. Something which will find greater application is running totals: for this, it is neccessary to store data from one record to another in some sort of variable or field, and this raises the first important point concerning "programming" in Datatrieve, which is that there are no default variables as there are in Basic or Fortran. All fields must be defined in a record or declared, and you must make the field large enough to hold the data planned for it. Starting with the same sample domain, the commands would be:

```
DECLARE RUNNING PIC 9(6).
RUNNING=0
FIND SAMPLE
FOR CURRENT MODIFY USING BEGIN
      T1=(A+B)*C
      END
SORT BY DESC T1
FOR CURRENT MODIFY USING BEGIN
      RUNNING=RUNNING+T1
      T2=RUNNING
      END
```

Since the running total will be in field T2, RUNNING has been declared to be the same size as T2. Notice that RUNNING must be initialized to zero: Datatrieve does not initialize any fields. In this example, the data is placed in the current collection rather than storing the running totals as the collection is being totaled by field T1 rather than by the primary key field of the sample domain. The current collection now looks like this:

computations into smaller segments can save a considerable amount of pool, and is especially useful in Datatrieve-11.

It should be noted that the statement WHILE DIF > 0.01 BEGIN could have been written in many different ways. One could also say WHILE (DIF > 0.01 OR DIF < -0.01) BEGIN, or WHILE DIF BETWEEN -0.01 AND 0.01 BEGIN or any other valid boolean expression. If any of these had been used, the line IF DIF < 0 THEN DIF=DIF*-1 which converts negative values to positive values would not be required.

One more example of this type of data processing will be fitting a trend line to data in a domain. This is the "least squares" method of fitting the best line to a set of data points, and is often used for such things as predicting future growth. The procedure is:

```
DEFINE PROCEDURE TREND
DECLARE SUMX USAGE IS REAL.
DECLARE SUMY USAGE IS REAL.
DECLARE SUMXY USAGE IS REAL.
DECLARE SUMXSQ USAGE IS REAL.
DECLARE SUMYSQ USAGE IS REAL.
DECLARE SLOPE USAGE IS REAL.
DECLARE INTERCEPT USAGE IS REAL.
DECLARE FIT USAGE IS REAL.
DECLARE TEMP USAGE IS REAL.
DECLARE N USAGE IS INTEGER.
N=0
SUMX=0
SUMY=0
SUMXY=0
SUMXSQ=0
SUMYSQ=0
READY SAMPLE
FOR SAMPLE BEGIN
     SUMX = SUMX + ITEM
     SUMY = SUMY + T1
     SUMXY = SUMXY + (ITEM * T1)
     SUMXSQ = SUMXSQ + (ITEM * ITEM)
     SUMYSQ = SUMYSQ + (T1 * T1)
     N=N+1
     END
TEMP = ((SUMX*SUMY/N)-SUMXY)
SLOPE=TEMP/((SUMX*SUMX/N)-SUMXSQ)
INTERCEPT = (SUMY - SLOPE*SUMX)/N
FIT = SLOPE * TEMP / (SUMYSQ - (SUMY*SUMY/N) )
PRINT SLOPE USING ZZZ9.9999, INTERCEPT USING ZZZ9.9999, FIT USING ZZZ9.9
999
FINISH SAMPLE
RELEASE N
RELEASE TEMP
RELEASE FIT
RELEASE INTERCEPT
RELEASE SLOPE
RELEASE SUMYSQ
RELEASE SUMXSQ
RELEASE SUMXY
RELEASE SUMY
RELEASE SUMX
END-PROCEDURE
```

The procedure follows the same rules as before as to declaring all variables and initializing them. The FOR statement is used to process the domain and sum up some values which will be required for the calculation. The question might arise as to why the procedure is summing up the values for X (ITEM) and Y (T1) and counting up the number of items in N when it could simply FIND the domain and then use the SUM and COUNT commands to have Datatrieve do the work. The answer is that the procedure has to go through the domain once anyway to sum the squares of the variables and the products of the two variables, and it is more efficient to also sum the other values at the same time than to have Datatrieve make additional passes through the domain to to the summing and counting, especially if this were to be done on a large domain. It is a good general rule to gather as much data at one time as possible to save time in processing (but don't store values you won't need). This is also shown by the use of an intermediate calculation for the value of TEMP: this expression is used in two other places, and it is more efficent to use four bytes of pool to store the value than to calculate it twice, and it is also faster. The data now in the domain and the answers look like this:

ITEM	A	B	C	T1	T2
1	0	0	0	1200	0
2	0	0	0	1800	0
3	0	0	0	1600	0
4	0	0	0	1900	0
5	0	0	0	1800	0
6	0	0	0	2100	0

```
DTR> :TREND
  SLOPE     INTERCEPT     FIT

137.1429    1253.3334    0.6954
DTR>
```

The statements which were missing from previous examples but are included here are FINISH and RELEASE. As pool is always a scarce resource, it is good practice to free up pool space by closing out domains and releasing space reserved for variables which are no longer used. Although only global variables actually require explicit release, it is best to get into the habit of releasing resources as soon as possible: in this example, if the RELEASE statements were not included, the variables would still be stored in pool after the procedure was finished.

At this point, the reader should have a grasp of what is possible in the way of equation solving in Datatrieve. More complex problems may be approached by breaking them down into smaller sections, each of which should yield to one of

the methods presented. For those who plan to go further with this approach, the following subjects in the Datatrieve manual will be of interest: chapter 5 (generally, and the ABORT, DECLARE, FOR and IF-THEN-ELSE commands specifically), chapter 6 (arithmatic and boolean expressions), chapter 12 (proceedures and indirect command files), appendix B (optimization), and especially the section of chapter 11 dealing with the USAGE clause, which describes the internal format of the different types of numbers. As noted in appendix B, COMP {INTEGER} is the most efficent type of storage; for real numbers REAL and DISPLAY (the default) should be the next most efficient. The author reccomends avoiding COMP-3 {PACKED}, COMP-5 {ZONED}, COMP-6 and DATE.

Readers may be interested in knowing where to find equations in suitable form for solution in Datatrieve (or other computer languages). Books on the particular subject (for example, a book on statistics for standard deviation or trend line fitting) are a good beginning, especially the older books which give instructions for solving the equations by hand, and even better, books which show how to solve the equations on pocket calculators. When such calculators were more expensive than they are now, and most had only four functions (rather than the specialized math or financial calculators now available), a number of books showing how to break down trigonometric functions, financial equations, etc. into a form which could be solved on a four function calculator were published, and these methods should be easily transferred to Datatrieve. They will also give worked examples, so the user can compare the answer obtained in Datatrieve with the answers in the book to determine if the equation has been correctly solved. Another good source is the manuals provided with programmable pocket calculators, which often give the formula and a worked example: the trend line example was obtained in this way. There are also books published for high-school and college math classes containing nothing but formulas, and some have functions expanded into series, which are particularly suitable for solution by computer. Finally, for those wishing to solve trigonometric functions, the Fortran-IV (Fortran-77) manual set contains an appendix describing the methods used to provide those functions and the accuracy obtained. The author would like to thank Mr. E. Sweeney for proofreading this paper, and for his suggestions, and for the comments received when the paper was presented in Atlanta.

Dear Editor,

I have been Datatrieving for almost a year. Some of my little-minded friends say there is no Magic Wombat. My system manager says, "If you see it in the 'Wombat Examiner', it is so." Please tell me the truth, is there a Magic Wombat?

Virginia

Virginia, your little-minded friends are wrong. They have been affected by the rigidity of a traditional data-processing age. They do not believe that which they do not see. They think that no program can be which their little minds cannot comprehend. All minds, Virginia, whether they be adult's or children's (or even Jim Starkey's) are little compared to this great universe of ours. Mankind's intellect is tiny when measured by the intelligence capable of grasping the whole of truth and knowledge and the even greater intelligence which created all.

Yes, Virginia, there is a Magic Wombat. He exists as surely as domains and records and procedures exist, and you know that they exist in abundance and give simplicity, joy and beauty to your data management. How dreary would be the world of DEC software if there were no Magic Wombat! It would be as dreary as if there were no Virginias. We would have no enjoyment except in sense and sight. There would be no magical DTR procedures, no poetry, no romance to enliven this existence.

Not believe in the Magic Wombat! You might as well not believe in VAX and virtual memories, might as well not even believe in software. You might ask your system manager to set software traps throughout your system, but even if you did not trap the Magic Wombat, what would that prove? Although no-one sees the Magic Wombat, that does not mean there is no Magic Wombat. Did you ever hold a software bug in your hand? Of course not, but you know that they exist. No-one can imagine all the wonders there are unseen inside your computer system.

You open the disk drive and see what makes the noise inside, but there is a veil covering the unseen world inside the disk which no amount of strength could open. Only faith, poetry and DTR can move that veil and view the glory beyond. Is it all real? Ah, Virginia, there is nothing in this world more real.

No Magic Wombat! Thank DEC! (Thank some special people at DEC)! He lives and grows stronger. Years from now, Virginia, he will continue to gladden the hearts of those who have kept alive the child, the innocent, wondering child inside us.

Editor.

[by Philip Dickerson (With thanks to Francis P. Church, editor, New York Sun, 1897 and apologies to anyone offended)]

PRINT ALL FATHER ,ALL KIDS WITH (AGE EQ 26) ,SKIP 2 OF FIRST 3 FAMILIES

```
                KID
FATHER   NAME   AGE

JIM


JIM      ELLEN  26


JOHN     JEAN   26
```

REPORTS - DETAIL LINES ARE NOT NECESSARY

The Report Writer may be used to generate reports with summary lines only ,and no detail lines. For example:

```
REPORT YACHTS WITH PRICE GT 0
SET REPORT-NAME = "EXAMPLE OF SUMMARY REPORT"
AT BOTTOM OF REPORT PRINT COL 10 ,"AVERAGE PRICE OF" ,
SPACE 1 ,COUNT USING ZZ9 ,SPACE 2 ,"YACHTS IS" ,SPACE 2 ,
AVERAGE PRICE(-) USING $$$,$$$
END-REPORT
```

EXAMPLE OF SUMMARY REPORT   22-Sep-82

AVERAGE PRICE OF 50 YACHTS IS $25,388

FIELD HEADERS

Sometimes field headers require more space than the actual field. One way to reduce the space required by a field header is to print it vertically as follows:

```
REPORT FIRST 10 YACHTS
SET REPORT-NAME = "DEMONSTRATION OF" /"VERTICAL FIELD HEADERS"
PRINT MANUFACTURER("BUILDER") ,MODEL ,RIG ,LOA("L"/"O"/"A") ,
DISP ,BEAM("BE"/"AM") ,PRICE
SET COLUMNS-PAGE = 50
END-REPORT
```

```
     DEMONSTRATION OF              22-Sep-82
     VERTICAL FIELD HEADERS          Page 1

                        L
                        O          BE
     BUILDER   MODEL   RIG  A  WEIGHT  AM   PRICE

     ALBERG    37      MK   II  KETCH  37    20,000
     ALBIN     79      SLOOP 26  4,200  10  $17,900
     ALBIN     BALLAD  SLOOP 30  7,276  10  $27,500
     ALBIN     VEGA    SLOOP 27  5,070  08  $18,600
     AMERICAN  26      SLOOP 26  4,000  08   $9,895
     AMERICAN  26-MS   MS    26  5,500  08  $18,895
     BAYFIELD  30/32   SLOOP 32  9,500  10  $32,375
     BLOCK I.  40      SLOOP 39 18,500  12
     BOMBAY    CLIPPER SLOOP 31  9,400  11  $23,950
     BUCCANEER 270     SLOOP 27  5,000  08
```

NOW FOR SOME LESS COMMON DETAILS

WHILE (UNDOCUMENTED)

The keyword WHILE exists in DTR ,but is not documented or supported. It is however very useful to allow conditional exits from "Repeat" loops as an example will show :

```
DECLARE YR PIC 99.
DECLARE T-DATE USAGE DATE.
T-DATE = "TODAY"
YR = 70                   ! Set YR to an arbitrary previous year
```

WHILE Statement allows a statement to be repeated as long as a specified condition remains TRUE.

```
WHILE T-DATE NOT CONT YR   ! This condition depends on DTR's
                           ! ability to match segments of Dates
YR = YR + 1                ! Continue to add 1 to YR until
                           ! today's date contains YR


PRINT YR

YR

82
```

The variable YR is now equal to the current year

```
RELEASE YR
RELEASE T-DATE
```

WHILE is extremely useful in procedures for storing data in domains (for a good example see the June 1982 issue of the Wombat Examiner page 4)

LABELS

There is sometimes a need to print more than one record across the page as for example when generating computer-printed labels with more than one column of labels. I believe this is not possible with DTR Report-writer, but it may be done with a few variables and the file that may be created with the OPEN and CLOSE commands. (I have found the ability to create a file which will contain the input and output of DTR to be extremely useful)

```
DEFINE PROCEDURE PRINT-YACHT-LABEL
DECLARE B1 PIC X(10) .        ! Variable to contain Builder's name
DECLARE B2 PIC X(10) .        ! Variable to contain Builder's name
DECLARE M1 PIC X(10) .        ! Variable to contain Model name
DECLARE M2 PIC X(10) .        ! Variable to contain Model name
DECLARE N PIC 9 .             ! Numeric variable for counting
B1 = " " ;B2 = " " ;M1 = " " ;M2 = " ";N = 0
                             ! Initialize variables
OPEN LABEL.LST               ! Open a file to contain screen output
                             !Set up loop to print records across page
FOR FIRST 10 YACHTS BEGIN    ! 10 Yachts chosen for this example
            N = N + 1
                             ! At every third record print across page
                             ! contents of two pairs of variables plus
                             ! the current Builder and Model fields

     IF N = 3 THEN PRINT SKIP 2 ,COL 11 ,B1(-) ,COL 31 ,B2(-) ,
     COL 51 ,BUILDER(-) ,SKIP 1 ,COL 12 ,M1(-) ,
     COL 32 ,M2(-) ,COL 52 ,MODEL(-) THEN N = 0

     B1 = B2      ! Shift contents of Builder and
     B2 = BUILDER! Model fields through the
     M1 = M2      ! two pairs of variables
     M2 = MODEL
END
                             ! At this point there will be
                             ! 0 ,1 or 2 records still not
                             ! printed. The following two
                             ! statements will print the
                             ! remaining records
```

```
IF N = 1 THEN PRINT SKIP 1 ,COL 11 ,B2(-) ,SKIP 1 ,COL 12 ,M2(-)
IF N = 2 THEN PRINT SKIP 1 ,COL 11 ,B1(-) ,COL 31 ,B2(-) ,SKIP 1 ,
COL 12 ,M1(-) ,COL 32 ,M2(-)

CLOSE                        ! Close the screen tracking file
RELEASE N,M2,M1,B2,B1        ! Release variables
END-PROCEDURE

:PRINT-YACHT-LABEL
```

ALBERG 37 MK II	ALBIN 79	ALBIN BALLAD
ALBIN VEGA	AMERICAN 26	AMERICAN 26-MS
BAYFIELD 30/32	BLOCK I. 40	BOMBAY CLIPPER
BUCCANEER 270		

Miscellaneous DTR Date Magic

*******   Important Information   ********

DTR Date fields and variables are stored in a unique manner as follows:

A base date is defined as "17-Nov-1858." A time unit (called a clunk) is defined as 100 nano-seconds thus:

```
1 second  ====  10,000,000 clunks
1 day     ====  86,400 seconds
1 day     ====  864,000,000,000 clunks
```

DTR date is stored as an 8-byte binary number which is equal to the number of clunks from Midnight on the Base Date to Midnight on the Date to be stored.

Thus "18-Nov-1858" is stored as 864,000,000,000 and "19-Nov-1858" is stored as 1,728,000,000,000 etc.

BOMBAY	CLIPPER	31	$23,950
BUCCANEER	270	27	NO PRICE AVAILABLE
BUCCANEER	320	32	NO PRICE AVAILABLE
C&C	CORVETTE	31	NO PRICE AVAILABLE
CABOT	36	36	NO PRICE AVAILABLE
CAL	2-27	27	NO PRICE AVAILABLE
CAL	2-34	33	NO PRICE AVAILABLE
CAL	29	29	NO PRICE AVAILABLE
CAL	3-30	30	NO PRICE AVAILABLE
CAL	35	35	NO PRICE AVAILABLE
CAPE DORY	25	25	$8,995
CAPE DORY	28	28	$21,990
CAPE DORY	TYPHOON	19	$4,295
CAPITAL	NEWPORT	28	NO PRICE AVAILABLE
CARIBBEAN	35	35	$37,850
CHALLENGER	32	32	$31,835
CHALLENGER	35	35	$39,215
CHALLENGER	41	41	$51,228
CHRIS-CRAF	CARIBBEAN	35	$37,850
COLUMBIA	35	35	NO PRICE AVAILABLE
COLUMBIA	41	41	$48,490
COLUMBIA	PAYNE 9.6	32	NO PRICE AVAILABLE

Example 2

Ex. 1 was relatively simple, but the next desire is to print the First 30 Yachts with the LOA field replaced as follows:

```
If LOA LE 29 then          print : "Too short"
LOA EQ 30                  print : "Just under Ideal"
LOA EQ 31                  print : "IDEAL LENGTH"
LOA EQ 32                  print : "Just over Ideal"
LOA GE 33                  print Actual Length
```

(This example uses a table - example 1 did not)

The statements to achieve this result are:

```
#*******          The statements were
#*******          edited out and
#*******          will be revealed
#*******          in a future issue
#*******
#*******          (The report however is unedited)
```

MANUFACTURER	MODEL	LENGTH OVER ALL	PRICE
ALBERG	37 MK II	37	$36,951
ALBIN	79	Too short	$17,900
ALBIN	BALLAD	Just under Ideal	$27,500
ALBIN	VEGA	Too short	$18,600
AMERICAN	26	Too short	$9,895
AMERICAN	26-MS	Too short	$18,895
BAYFIELD	30/32	Just over Ideal	$32,875
BLOCK I.	40	39	
BOMBAY	CLIPPER	IDEAL LENGTH	$23,950
BUCCANEER	270	Too short	
BUCCANEER	320	Just over Ideal	
C&C	CORVETTE	IDEAL LENGTH	
CABOT	36	36	
CAL	2-27	Too short	
CAL	2-34	33	
CAL	29	Too short	
CAL	3-30	Just under Ideal	
CAL	35	35	
CAPE DORY	25	Too short	$8,995
CAPE DORY	28	Too short	$21,990
CAPE DORY	TYPHOON	Too short	$4,295
CAPITAL	NEWPORT	Too short	
CARIBBEAN	35	35	$37,850
CHALLENGER	32	Just over Ideal	$31,835
CHALLENGER	35	35	$39,215
CHALLENGER	41	41	$51,228
CHRIS-CRAF	CARIBBEAN	35	$37,850
COLUMBIA	35	35	
COLUMBIA	41	41	$48,490
COLUMBIA	PAYNE 9.6	Just over Ideal	

(by P. Dickerson)

DATATRIEVE HINTS and KINKS

Gary Saxer
Anaconada-Ericsson
21201 Oxnard Street
Woodland Hills, CA 91367

Forward:

These hints and kinks are things I have found useful. Some of the information is a repeat of that in the DTR manuals. Some is my own way of saying things, and some is only comprehendable by a wombat! I hope that these ramblings will make sense to somebody.

ERRORS

Sometimes I would like to say to DATATRIEVE:

"Expected some help, encountered lousy error message"

or

"That error message is useless, or used out of context"

HINT - When an error message is displayed, (especially the expected/encountered one), focus on the location in the quoted string, and all that is "to the left" of the string. This may include the the previous line or lines.

```
IF WOMBAT = "GREY" THEN BEGIN
    PRINT "This guy is sick"
END
ELSE PRINT "It is all right"
```

The error (?) message will be:
Expected statement, encountered "ELSE"

This occurs because the ELSE clause of DTR is optional, and DTR has determined that the IF statement is finished when it finds the END.

This form of the statement will work:

```
IF WOMBAT = "GREY" THEN BEGIN
    PRINT "This guy is sick"
END ELSE PRINT "It is all right"
```

HINT - Make sure your ELSE clause is on the same line as the last part of your THEN clause. In this case, the END is the last part of the THEN clause. Putting the ELSE on the same line as END is always good.

Expected statement, encountered "READY".

What is this error message? Isn't READY a statement? NO!!! READY is a COMMAND. There is a difference. You may ask: "What is that difference?" Here is the world's best definition:

HINT - A command is one of the words on pages 5-2 and 5-3 of the DATATRIEVE-11 User's Guide (pages 13-2 to 13-4 for VAX-11), which are followed by (C), statements are followed by (S).

HINT - As general help, you can think this way: "Commands can't be used in loops, statements can (except FIND)". This is not perfect, and Anne Duncan won't like it, but it does help when you are too lazy to look at the book (you do look at the book!).

"BLARK" is undefined or used out of context.

Everyone hates this one. "Context" is a fun DTR game where you and DTR fight to see who can outguess the other. The only problem with the game is that, unless you really know what you are doing, (and nobody ever really does), DTR will always win. This particular message is telling you that, although you may know what "BLARK" is, DTR is confused.

HINT - Try to figure out why DTR doesn't know what you are saying.

(1) Find "BLARK" in the statement. (Not always easy if "BLARK" appears several times.)

(2) If you have a CURRENT collection, then either you have not SELECTed a record, or "BLARK" is not a field in the record. (This usually occurs when you misspell a field name: always make sure you have spelled "BLARK" correctly".)

(3) If the spelling looks correct, and the person who entered the line is new to DTR, I'll bet they used the BACKSPACE key do do some deletions! This is a very obscure problem. The line looks correct, and the next time you type it (without any need for corrections) it works! I have found it useful to physically remove the backspace key from the keyboard until new users learn to use the DELETE key.

(4) If you do have a collection, then maybe what you really wanted was to type ALL BLARK. If you are doing something in a collection (usually printing), and you want more than the SELECTed record to be accesed, then you usually need to put the word ALL before the field name(s). Why? If DTR always affected all of the records, then there would need to be a special word for only ONE! It seems more safe to let the user usually specify when only one record is affected (which is the default in DTR) than to assume all.

(5) If you get to here, you are probably doing something a little more advanced than typing "BLARK" incorrectly. By now you should understand that DTR

If the parentheses were left out, DTR would assume that you were interested in the NAME field of OWNERS which either had the OWNER.BUILDER field equal to the Z.BUILDER field or the OWNER.BUILDER field equal to the OWNER.PRICE field (which does not exist!) The error message if DATATRIEVE-11 is very nasty and does not seem to help; DTR is really confused and can't help you much.

HINT: When using EQ, EQUAL, or = in a print-list, always use parentheses around the expression.

VIEWS and LISTS

Have you ever wondered how to TOTAL a field from a list? Why not try something like this:

```
01 VACA.
   10 ACC-TO-DATE USAGE COMP-1 EDIT-STRING IS ZZ9.9999.
   10 VAC-TAKEN  COMPUTED BY (TOTAL TAKEN OF VACA.ADJUSTMENTS WITH
      TYPE = "V","v")
      EDIT-STRING IS ZZZ.99.
   10 BEGIN-BALANCE USAGE COMP-1 EDIT-STRING IS ZZZ.99-.
   10 CURRENT-VAC-BALANCE COMPUTED BY (BEGIN-BALANCE +
      ACC-TO-DATE - VAC-TAKEN)
      EDIT-STRING IS ZZZZ.99-B
      QUERY-NAME IS BALANCE.
   05 ADJ.
   10 NUMB-ADJ USAGE COMP EDIT-STRING IS ZZ QUERY-HEADER IS "NA"
      QUERY-NAME IS NA.
   10 ADJUSTMENTS OCCURS 0 TO 99 TIMES DEPENDING ON NUMB-ADJ.
   15 ADJUST.
   20 ADJ-TYPE PIC X
      QUERY-NAME IS TYPE.
   20 HOURS-TAKEN  USAGE COMP-1 EDIT-STRING IS ZZ9.99
      QUERY-NAME IS TAKEN.
;
```

A report on a domain using this format can have a statement like:

AT BOTTOM OF DEPT PRINT TOTAL VAC-TAKEN

You CANNOT sort by VAC-TAKEN (except with VAX-11).

If you don't like having a field in your record, why not create a COMPUTED-BY field:

DECLARE TOT-AGE COMPUTED BY TOTAL AGE OF KIDS.

Then you could have a REPORT statement like:

AT BOTTOM OF FATHER PRINT TOTAL NUMBER-KIDS USING ZZ9,
   TOTAL (TOT-AGE) ("TOTAL"/"AGE") USING ZZZ9,
   (TOTAL (TOT-AGE))/ (TOTAL NUMBER-KIDS) ("AVER"/"AGE") USING ZZZ9,

AVERAGE (AVERAGE AGE OF KIDS) ("AV"/"AGE") USING ZZZ9

HINT - The whole secret to getting totals from lists is to treat the list as though it was a domain. Keep saying to yourself: "A list can have more than one record or field, am I trying to ask for only one or many?". What if you want only one value? It may be possible to get it. There are two easy ways, depending on whether your value is a number or a string. The first example is for a string. In this example, a record is being stored in the FAILURES domain with a TAG-NUMBER. In another domain (DEPOTS), the TAG-NUMBER is stored along with the PART-TYPE. We would like to not have to ask for PART-TYPE since we already know the TAG-NUMBER: (this is a piece of a procedure)

DECLARE T-P-NO PIC XXX.

```
CE-TAG-NO = *."CE tag number"
T-P-NO = NEW.P-NO
FOR FIRST 1 DEPOTS WITH TAG-NUMBER = NEW.CET
T-P-NO = PART-TYPE !This statement will be executed once
P-NO = T-P-NO
```

Note that T-P-NO is given a known value before the FOR statement. This is in case there is no record with TAG-NUMBER = NEW.CET! The "FIRST 1" clause keeps the amount of access to DEPOTS to a minimum. (Note that this is pretty fast as long as DEPOTS uses TAG-NUMBER as an indexed key.)

Now think of the same example as above, but consider the difference if we were interested in PART-NUMBER instead of PART-TYPE. It could look something like this:

DECLARE T-P-NO PIC 999.

```
CE-TAG-NO = *."CE tag number"
T-P-NO = MAX PART-NUMBER OF FIRST 1 DEPOTS WITH
TAG-NUMBER = NEW.CET
P-NO = T-P-NO
```

In this way, the "list" of DEPOTS with the same tag number (I'll grant that there is only one, but DTR doesn't know that) is accesed and the maximum value (that is the only value) is computed. I have used this trick in many unusual places. It seems that, if you need a value from an RSE, and the syntax of the statement allows a "value expression", you can simply use MIN, MAX, TOTAL, or COUNT, and then have your RSE!

In case some of you are interested, here is the complete entry procedure for STOREing records in the FAILURES domain:

```
DELETE FAILENT;
DEFINE PROCEDURE FAILENT
READY FAILURES SHARED WRITE
SET DICTIONARY CE:DICTIONRY
READY DEPOTS SHARED
SET DICTIONARY DT:DICTIONRY
DECLARE OLD-PART-NO PIC XXXXXXX.
DECLARE T-TECH-NUMB PIC 999.
DECLARE T-DATE USAGE DATE.
```

```
DECLARE T-STATION PIC XXX.
DECLARE T-SN PIC 9(8).
DECLARE T-P-NO PIC XXX.
DECLARE CTRL-G PIC X.
CTRL-G = ""    !There is a control G in the quotes
T-TECH-NUMB = 0
OLD-PART-NO = "0"

REPEAT 1000 BEGIN
        STORE NEW IN FAILURES USING BEGIN
                PRINT " "
                T-TECH-NUMB = **."tech number"
                IF T-TECH-NUMB NOT IN TECH-TBL THEN BEGIN
                    PRINT "Not a valid tech - re-enter"
                    T-TECH-NUMB = **."tech number"
        END
        T-DATE = **."date"
        T-STATION = **."station"
        IF T-STATION NOT IN STA-TBL THEN BEGIN
                PRINT "Station not in station table"
                T-STATION = **."station"
        END
        IF T-TECH-NUMB = 0 THEN ABORT CTRL-G|"You MUST answer Y the
first time."
        TECH-NUMB = T-TECH-NUMB
        DATE = T-DATE
        STATION = T-STATION
        IF OLD-PART-NO EQ "0" THEN BEGIN PART-NO = *."part
                or kit number"
        IF NEW.P-NO = "0" THEN ABORT "You cannot enter 0 the first time"
        END ELSE BEGIN
                PRINT COL 1, "Last part number was:", SPACE 2,  OLD-PART-NO (-)
                PART-NO = *."part or kit number, use 0 for same"
        END
        IF NEW.P-NO = 0 THEN BEGIN
                NEW.PART-NO = OLD-PART-NO
        END ELSE IF NEW.P-MO = " " THEN BEGIN
                PRINT CTRL-G|"Must have month or RET"
                PART-NO = *."ALL of the part or kit number"
        END
        SERIAL-NO = *."serial number"
        PF = *."pass/fail code"
        IF NEW.PF = "P" THEN FAULT-CODE = "---"
        COST-CODE = *."cost code"
        IF NEW.COST-CODE = "0" THEN BEGIN
                NEW.COST-CODE = " "
        END
        IF NEW.PF = "F" THEN BEGIN
                FAULT-CODE = *."fault code"
                IF NEW.FAULT-CODE = "0" THEN BEGIN
                NEW.FAULT-CODE = " "
        END
        IF NEW.FAULT-CODE NOT IN FAULT-TBL THEN BEGIN
                PRINT "This is not a valid fault code -try again"
                FAULT-CODE = *."fault code"
```

```
        END
        REF-DES = *."Reference Designator"
        IF NEW.REF-DES = "0" THEN BEGIN
                NEW.REF-DES = "   "
        END
        IF NEW.REF-DES NE " " THEN BEGIN
                MFG-CODE = *."Manufacturer Code/Date"
        END
        END
        IF (NEW.PART-NO NE OLD-PART-NO) THEN BEGIN
                TEST-TIME = *."Total Test Time"
        END
        IF NEW.P-MO = "RET" THEN BEGIN
                CE-TAG-NO = *."CE tag number"
                T-P-NO = NEW.P-NO
        FOR DEPOTS WITH TAG-NUMBER = NEW.CET T-P-NO = PART-TYPE
        P-NO = T-P-NO
        END
        OLD-PART-NO = NEW.PART-NO
        DATE-STAMP = "TODAY"
        END
END
FINISH DEPOTS
READY FAILURES SHARED
END-PROCEDURE
```

There are several tricks in this procedure, see if you can find these:

(1) The DEPOTS domain and record definition are in another dictionary, after it is READYed, the current dictionary is reset.

(2) Entering the value "0" results in fields defined as PIC X() being blank filled. (The zero key is on the keypad and it is faster for a data entry person to hit it rather than change to the other keys for the space bar.)

(3) By using the context variable (wow those things keep showing up!) called NEW, a value which has just been entered may be examined.

(4) The record has a DATE-STAMP. It is automatically entered into the USAGE DATE field by using the string "TODAY". This special string has a value of the current system date. When some verification is performed, it can be done on a day-by-day basis.

(5) The format of the raw data has the tendency to have many of the same part numbers in a row. The procedure "remembers" the last part number so the data entry person need not type it over and over.

(6) Those failures which are "returns" (P-MO = "RET"), are assumed to have a CE-TAG-NUMBER, and this field is only asked for when necessary.

(7) When a part finally passes, it can't have a failure code, so it's not requested.

(8) Many of the fields are checked against tables, this helps to ensure the integrity of the data entered.

(9) The first several fields remain constant for many records, (this information appears at the top of the page on the data sheet and is assumed to be the same for all of the detail lines), they are only requested once. This requires the entry person to type CTRL-Z when one of these fields changes; but it is easier to type CTRL-Z and then :FAILENT every once in a while than to ask if it has changed every time!

```
!                    NAME-RANGE-LIST

DELETE NAME-RANGE-LIST;
DEFINE PROCEDURE NAME-RANGE-LIST
DECLARE FIRST-LETTER                    PIC X.
DECLARE LAST-LETTER                     PIC X.
DECLARE FIRST-TWO-LETTERS               PIC XX.
DECLARE LAST-TWO-LETTERS                PIC XX.

FIRST-LETTER = *."the first letter of the LAST-NAME range..."
LAST-LETTER = *."the last letter of the LAST-NAME range..."

FIRST-TWO-LETTERS = FIRST-LETTER || "A"
LAST-TWO-LETTERS = LAST-LETTER || "z"

READY DOMAIN
FIND DOMAIN WITH LAST-NAME BETWEEN
        FIRST-TWO-LETTERS AND LAST-TWO-LETTERS
SORT BY LAST-NAME
FOR CURRENT
    PRINT "<GENDER-SALUTATION>" | GENDER || " " | LAST-NAME,
    SKIP 1,"<NAME>" | FULL-NAME,
    SKIP 1,"<ADDRESS1>" | ADDRESS-1,
    SKIP 1,"<ADDRESS2>" | ADDRESS-2,
    SKIP 1,"<CITY-STATE>" | CITY || ", " | STATE | " " | ZIP-CODE,
    SKIP 1,"<>" ON SY:[35,1]WORD99.TMP
FINISH


PRINT "Now type TE WORD99.TMP <RETURN> and then EX<ESC><ESC>"
PRINT "followed by the (WD) option of WORD11 to convert (WORD99.TMP)"
EXIT
END-PROCEDURE
================================================================

DTR :NAME-RANGE-LST
Enter the first letter of the LAST-NAME range...: W
Enter the last letter of the LAST-NAME range...: W

================================================================
```

Results:

```
<GENDER-SALUTATION>Mr. Wombaten
<NAME>W.B. Wombaten, President
<ADDRESS1>Austrail Motel, Inc.
<ADDRESS2>455 South Hairy Way
<CITY-STATE>Camdec, LD 81666
<>
<GENDER-SALUTATION>Mr. Weightlessness
<NAME>Federico H. Weightlessness
<ADDRESS1>Lowdown's Hotel
<ADDRESS2>999 Fourth Way
<CITY-STATE>Old River, NY 17785
<>
<GENDER-SALUTATION>Mr. Worldbfree
<NAME>Btsec Worldbfree
<ADDRESS1>c/o Roilic Inn
<ADDRESS2>455 Teenager Avenue
<CITY-STATE>Lastville, FR 74821
<>
```

```
DDDDDDDDDDDDDDDD    TTTTTTTTTTTTTTTT    RRRRRRRRRRRRRRRR    >>>>>
DDDDDDDDDDDDDDDDD   TTTTTTTTTTTTTTTT    RRRRRRRRRRRRRRRRR   >>>>>>
DDDDDDDDDDDDDDDDD   TTTTTTTTTTTTTTTTT   RRRRRRRRRRRRRRRRR    >>>>>>
DDDDD       DDDDD        TTTTT          RRRRR       RRRRR     >>>>>>
DDDDD       DDDDD        TTTTT          RRRRR       RRRRR      >>>>>>
DDDDD       DDDDD        TTTTT          RRRRR       RRRRR       >>>>>>
DDDDD       DDDDD        TTTTT          RRRRRRRRRRRRRRRRR        >>>>>>
DDDDD       DDDDD        TTTTT          RRRRRRRRRRRRRRRR          >>>>>>
DDDDD       DDDDD        TTTTT          RRRRRRRRRRRRRRR          >>>>>>
DDDDD       DDDDD        TTTTT          RRRRR     RRRRR         >>>>>>
DDDDD       DDDDD        TTTTT          RRRRR     RRRRR        >>>>>>
DDDDD       DDDDD        TTTTT          RRRRR     RRRRR       >>>>>>
DDDDDDDDDDDDDDDDD        TTTTT          RRRRR     RRRRR      >>>>>>
DDDDDDDDDDDDDDDDD        TTTTT          RRRRR     RRRRR     >>>>>>
DDDDDDDDDDDDDDDD         TTTTT          RRRRR     RRRRR    >>>>>
```

This document includes all of the information that is currently active in [99,6]'s DATATRIEVE dictionary. It is provided to the user for documentation of their DATATRIEVE system. As changes to, additions to, and deletions of definitions in the dictionary occur, the user may request an update of this document from Information Systems. The definitions are in alphabetical order...

```
DDDDDDDDDDDDDDDD    FFFFFFFFFFFFFFFF    NNNNN       NNNNN    >>>>>
DDDDDDDDDDDDDDDDD   FFFFFFFFFFFFFFFF    NNNNNNN      NNNNN    >>>>>>
DDDDDDDDDDDDDDDDD   FFFFFFFFFFFFFFFF    NNNNNNNN     NNNNN     >>>>>>
DDDDD       DDDDD        FFFFF          NNNNNNNNN    NNNNN      >>>>>>
DDDDD       DDDDD        FFFFF          NNNNNNNNNN   NNNNN       >>>>>>
DDDDD       DDDDD        FFFFF          NNNNN NNNN   NNNNN        >>>>>>
DDDDD       DDDDD        FFFFFFFFFFF    NNNNN NNNN   NNNNN         >>>>>>
DDDDD       DDDDD        FFFFFFFFFFF    NNNNN NNNNN  NNNNN          >>>>>>
DDDDD       DDDDD        FFFFFFFFFFF    NNNNN  NNNN  NNNNN         >>>>>>
DDDDD       DDDDD        FFFFF          NNNNN  NNNN  NNNNN        >>>>>>
DDDDD       DDDDD        FFFFF          NNNNN   NNNNNNNNN        >>>>>>
DDDDD       DDDDD        FFFFF          NNNNN   NNNNNNNNN       >>>>>>
DDDDDDDDDDDDDDDDD        FFFFF          NNNNN    NNNNNNNN      >>>>>>
DDDDDDDDDDDDDDDDD        FFFFF          NNNNN    NNNNNNN      >>>>>>
DDDDDDDDDDDDDDDD         FFFFF          NNNNN     NNNNNN     >>>>>
```

```
!!!!!!!!!
!!!!!!!!!        The following are the production DOMAINS...
!!!!!!!!!

!                       THE-DOMAIN

DELETE THE-DOMAIN;
DEFINE DOMAIN THE-DOMAIN
USING THE-RECORD ON DEVICE:[99,6]FILE.IDX
;
!!!!!!!!!
!!!!!!!!! The total number of DOMAINS is 1...
!!!!!!!!!


!!!!!!!!!
!!!!!!!!!        The following are the production RECORDS...
!!!!!!!!!

!                       THE-RECORD

DELETE THE-RECORD;
DEFINE RECORD THE-RECORD
ALLOCATION IS LEFT-RIGHT
01  THE-RECORD.
02  LOCATION                  PIC 9(4).
02  LOCATION-NAME             PIC X(30).
02  STATE                     PIC XX.
02  ROOMS                     PIC 9(4).
02  ROOMS-SWITCH              PIC X
VALID IF ROOMS-SWITCH EQUAL "O", "C", "D".
;

!!!!!!!!!
!!!!!!!!! The total number of RECORDS is 1...
!!!!!!!!!


!!!!!!!!!
!!!!!!!!!
!!!!!!!!!        The following are the production PROCEDURES...
!!!!!!!!!

!                       BUILD-THE-DOMAIN

DELETE BUILD-THE-DOMAIN;
DEFINE PROCEDURE BUILD-THE-DOMAIN
DEFINE FILE THE-DOMAIN ALLOCATION = 2000,
KEY = LOCATION,
KEY = LOCATION-NAME,
KEY = ROOMS,
KEY = ROOMS-SWITCH
END-PROCEDURE
```

Operating System	Computer	Name	Telephone
		Kelly, Joe WAYMON GORDON Worchestel Road North Grafton Mass. 01536	(617) 839-4441 X5480
		Saad, David UNITED TECHNOLOGIES Micro Electronics Center 1365 Garden of the Gods Road Colorado Springs, Colorado, 80907	(303) 594-8098
		Swanger, James G. D. Searle & Co. P. O. BOX 5110 Chicago Ill. 60680	(312) 982-7430
		Saxer, Gary AXXA CORP 21201 OXNARD ST. Woodland Hills, Ca. 91367	(213) 992-8110
		Eade, Darrell J. NAVAL UNDERSEA WATER E. S. Heyport Washington 98345	(206) 396-2501
		Wool, Chris E. I. DuPONT Engineering Dept. Willington Delaware 19898	(302) 366-4610
		Stephenson, Glen KAISER HEALTH PLAN 4747 Sunset Blv, 1st Floor Los Angeles Ca. 90027	(213) 667-8435

Fall '82 DECUS Symposium

AIRPORT TRANSPORTATION

Luxe Livery Service, Inc. has offered to provide our attendees Airport transportation to and from the Los Angeles Airport at a discounted fee. The total cost for our attendees will be fifteen dollars ($15.00) each way excluding driver gratuity.

You can arrange to be met at the airport outside your airline by a uniformed driver who knows your name and where you are going. The driver will load your baggage and take you directly to your hotel.

FROM LOS ANGELES AIRPORT

After claiming your baggage please proceed to the center island, across from the baggage claim area. The van service is only permitted to park to load passengers. If you do not see Luxe Livery's green van in your area, watch for them in traffic - they may be circling the airport. To be recognized by the driver detach and wave the green portion of your transpass as he approaches. In the event you do not make contact with your driver within 10 minutes, please phone (800) 422-4267 or (714) 558-1413 and ask to speak with the dispatcher.

TO LOS ANGELES AIRPORT

Call (714) 558-1413 after 7:00 p.m. the evening before service to confirm your reservation and pick-up time. Meet the van at the hotel's main entrance. Be prompt! The drivers are not permitted to wait more than 10 minutes. Remember, Airport check-in lines are usually quite long.

SPECIAL NOTES:

1. Acquire a transpass from the DECUS office. A special mailing will be done for all preregistered attendees and DECUS Leadership. Anyone else wishing to recieve the transpass must call the DECUS Office at (617) 467-4875.

2. Each attendee is required to have their own transpass. Therefore, when requesting transpasses from the DECUS Office please state the correct number you need.

3. Phone Luxe Livery Service and make an advanced reservation. The phone number for outside California is (800) 854-8171 and within California (800) 422-4267.

4. The transpass must be presented to the driver at the time of transfer to be eligible for the discounted rate.

5. The transpass has no value in itself, cannot be bought or sold, and must be used in conjunction with CASH ONLY at the time of transfer.

6. Please pass this information on to your SIG/LUG members so they can also benefit from this offering.

DECUS

DIGITAL EQUIPMENT COMPUTER USERS SOCIETY
ONE IRON WAY, MRO2-1/C11
MARLBORO, MASSACHUSETTS 01752

INSTALLATION

MOVING OR REPLACING A DELEGATE?

Please notify us immediately to guarantee continuing receipt of DECUS literature. Allow up to six weeks for change to take effect.

( )   Change of Address
( )   Delegate Replacement

DECUS Membership No.: _____

Name: _____

Company: _____

Address: _____

_____

State/Country: _____

Zip/Postal Code: _____

Mail to: DECUS - ATT: Membership
One Iron Way, MRO2-1/C11
Marlboro, Massachusetts 01752 USA

Affix mailing label here. If label is not available, print old address here. Include name of installation, company, university, etc.