

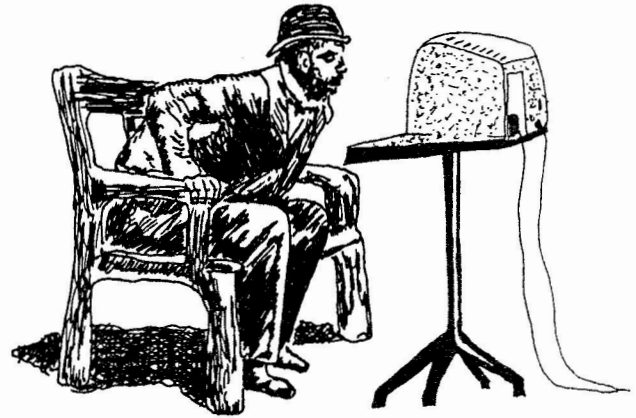
# The Wombat EXAMINER



*"Increases the Circulation of Anyone in America"*

March 1983  
Volume 5 Number 1

In this Issue:



The **Adventure**  
of the  
**Missing Context**

Copyright © Digital Equipment Corporation 1983  
All Rights Reserved

It is assumed that all articles submitted to the editor of this newsletter are with the authors' permission to publish in any DECUS publication. The articles are the responsibility of the authors and, therefore, DECUS, Digital Equipment Corporation, and the editor assume no responsibility or liability for articles or information appearing in the document. The views herein expressed are

#### Experiences in Developing In-house Datatrieve Training

In this session David Doxey of DIGITAL will discuss a training program that has been developed to quickly train over 400 MIS professionals.

#### Applying Datatrieve to a Non-data Base Problem

In this session James Swanger of G.D. Searle & Company will discuss a method developed using Datatrieve to generate a listing of unique bibliographic citations from input files containing any number of duplicate citations.

#### A Tutorial on Datatrieve Advanced Features

In this session Poonam Bahl, Kiran Shah, and R. Srinivasan of Hinditron Computer Systems & Consultants Private Ltd. (Bombay, India) will show how to produce user-friendly Datatrieve documentation.

---

Tuesday, May 24

---

#### Datatrieve in the Police Department

In this session Art Gann of the Evansville Police Department will discuss the implementation of computer applications with Datatrieve by police personnel who had no prior data processing experience or training.

#### Maintaining, Planning and Scheduling with Datatrieve

In this session Bob Lott of duPont will present the implementation of a Datatrieve based system for the planning and scheduling of maintenance work in a chemical plant.

#### Application Design Workshop

In this session Joan Hilton of G.D. Searle & Company and Cynthia Mealy of Boeing Commercial Airplane Company will describe the way in which a typical application might be analyzed and implemented in order to provide guidance for new users.

#### VAX-11 DATATRIEVE Technical Tutorial

This tutorial is designed to help VAX-11 DATATRIEVE users enhance their use of Datatrieve in writing applications. This session will focus on the more difficult to understand features including the use of distributed VAX-11 DATATRIEVE.

#### Presenting DATATRIEVE-11 Version 3

In this session the DIGITAL Datatrieve Development Group will discuss the new features and availability of DATATRIEVE-11 Version 3. Emphasis will be placed on the new distributed capabilities of accessing data using both DATATRIEVE-11 and VAX-11 DATATRIEVE.

On the Education of new Datatrieve Users

In this session Steve Pacheco of Athena Systems will discuss the problems of educating new users of Datatrieve.

Integration of the Command Definition Utility with VAX-11 DATATRIEVE

In this session, William Kator and Gerard Czadowski of Borg-Warner Research Center will present a number of applications which use VAX-11 DATATRIEVE and the Command Definition Utility integrated through FORTRAN programming.

---

Wednesday, May 25

---

Record Definition Workshop

In this workshop Bart Lederman of ITT World Communications will expand upon the material in the manuals to show the relative merits of different data descriptions and their uses. He will also show many useful "tricks" developed by users to meet different application needs.

Development and Utilization Panel

This panel will discuss the organization of their systems and the adaptation of Datatrieve to their applications with emphasis on the functionality of the language and the creativity of the developer.

Datatrieve in the Office

In this session the topic of Datatrieve in the Office Environment will be discussed.

DATATRIEVE-11 Version 3 Field Test Panel

A panel of users who have field tested DATATRIEVE-11 Version 3 will discuss their experiences with the software and the transition from the previous release.

Using Datatrieve for a Control System Data Base

In this session Nancy Greene from Los Alamos National Laboratory will discuss the data bases involved with the control of the Proton Storage Ring at Los Alamos.

Datatrieve SIG Business Meeting

This session is devoted to addressing items of business relating to the Datatrieve SIG which are of general interest to members of the SIG.

Domain Tables:  
Stored in a file  
IN is ANY  
VIA is FROM

RMS Locking:  
File level locking  
Timed wait for records  
Unlock after every get

DBMS Locking:  
Record level locking  
True concurrent update  
Read locks on collections

Reflexive Joins:  
FOR case works fine  
FIND case fails

FOR A IN YACHTS CROSS B IN  
YACHTS OVER BUILDER  
PRINT A.MODEL, B.MODEL

READY YACHTS AS FOOBAR  
FIND YACHTS CROSS FOOBAR OVER BUILDER  
PRINT ALL YACHTS.MODEL, FOOBAR.MODEL

Outer Join:  
What is it?  
Can't do it in DTR  
Can do each piece

FOR A CROSS B OVER FRIBBLE

FOR A WITH NOT ANY B WITH  
A.FRIBBLE = B.FRIBBLE

FOR B WITH NOT ANY A WITH  
A.FRIBBLE = B.FRIBBLE

Join Optimization:  
Use keys!  
Keyed domains last  
Domain with fewest records first

User Functions:  
If you can't do it in DTR...  
Must be position independent  
Use stack or VM for storage

Plot Language Restriction:  
Why don't we publish the graphics  
language?  
Not the right product  
Not designed for end user

## CALL INTERFACE

Anticipated Uses:  
Application programs  
(DTR as access method)  
Canned packages  
(Mimic terminal interface)  
Intelligent data manager  
(Generate requests on the fly)  
Remote Datatrieve server

What is it good for?  
Productivity  
Data Independence  
Database Independence  
Efficiency  
Distributed performance

Stall Points:  
Basic concept of the inteface  
DTR runs until it stalls  
Stall tells you what to do

Stall Points:  
Need a command  
Need a prompt  
Have a print line  
Have a message / condition  
Have a record for host program  
Need a record from host program

Ports:  
Communication mechanism  
An in-memory record  
DECLARE or DEFINE and READY  
Act like domains in DTR language  
STORE of port causes "port get"  
Access to port causes "port put"

Programming Hints:  
Use REPEAT of FOR <port> loops  
Use prompts  
Synchronize with DISPLAY  
Make ports specific to task



```

DECLARE PORT FOO
  01 GROUP.
    03 VALUE1 REAL.
    03 VALUE2 REAL.
;
FOR YACHTS STORE FOO USING
  BEGIN
    VALUE1 = BEAM;
    VALUE2 = LOA;
  END;
DISPLAY "FOO";

```

Programming Hints:

- Very flexible interface
- Application may not need it all
- Build a layer for what is needed

The layer should:

- Check stallpoints
- Check DTR calls for errors
- Handle messages and conditions
- Worry about DTR conditions

Efficiency

- Initialize at the start
- Shut down at the end
- Never put DTR\$COMMAND in a loop
- The more work per  
DTR\$COMMAND the better

**DISTRIBUTED ACCESS**

Efficient access to remote data:

- Define a domain that points to remote
- READY a domain AT remote
- Query decomposition
- Maximizes parallel processing
- Minimizes network traffic

Restrictions:

- No WHILE, IN or ANY
- No CONTAINING with a prompt
- No CROSS of remote domain
- Use unique field names for each domain
- No reprompt for validation errors
- No STORE or MODIFY of hierarchies

Programming Hints:

- Develop application locally
- Use local loopback for testing
- Location independent procedures

## A REFERENCE TOOL FOR DATATRIEVE

Susan K. Dakuzaku  
California Department of Fish & Game  
Rancho Cordova, California

### Abstract

This paper presents a hierarchical approach to a Datatrieve session showing the relationship between commands, statements and what you wish to do. This approach makes use of a structured reference guide developed using the Warnier-Orr methodology. This reference guide can prove a valuable training tool in Datatrieve workshops.

### Introduction:

When I was learning to use Datatrieve the only reference I had was the User's Guide. Though complete in its presentation, it is not organized ("structured") in terms of application making it awkward to use as a working reference without an already working knowledge of Datatrieve. So during this time I developed a series of diagrams documenting a Datatrieve session using the Warnier-Orr methodology. I found these diagrams an excellent reference guide for those learning Datatrieve and incorporated them into classes I have conducted.

This paper will cover a brief introduction to the Warnier-Orr diagram followed by the Datatrieve diagrams themselves.

Note: These diagrams were developed for Datatrieve-11.

### The Warnier-Orr Diagram

Warnier diagrams first appeared in "The Logical Construction of Programs" by Jean-Dominique Warnier. They were then modified by Ken Orr in "Structured Systems Development" resulting in the Warnier-Orr diagram.

One of the ways it is used is to represent a process. In this case it shows the flow of control of that process. We will now look at how a process is represented with a Warnier-Orr diagram.

### Hierarchy

Hierarchy is represented by the brace. Sequence is simplified within the brace as elements are read from top to bottom. In Diagram 1 a process is broken down into its more elementary parts. The "begin" and "end" are used to designate steps necessary to begin and end a particular process.

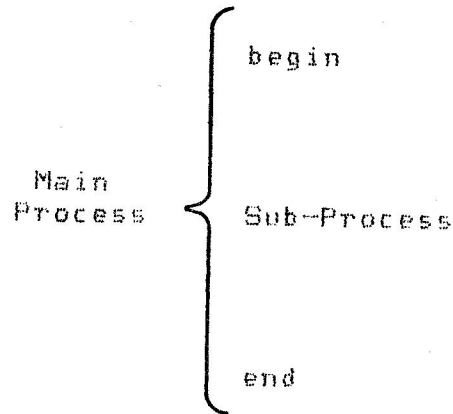


Diagram 1

### More on Hierarchy

Any part of the diagram may be broken down further by use of additional braces to show as much detail as desired. Hierarchy is determined by reading the braces from left to right. In diagram 2 we now see that to begin the Main Process, we execute an Initialization process (still very general but as I said, we can use additional braces if we wish to show more detail). Then we execute the Sub-Process which is shown to consist of 3 distinct Steps. To end the Main Process, we execute the Termination process.

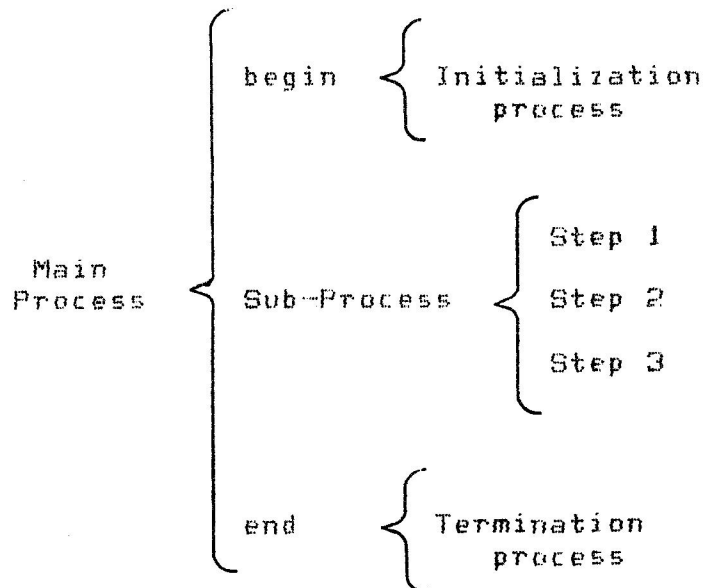


Diagram 2

### Repetition

Repetition is represented by specifying the parentheses below the element the number of times the element is to be repeated. The normal case of unity (1) is

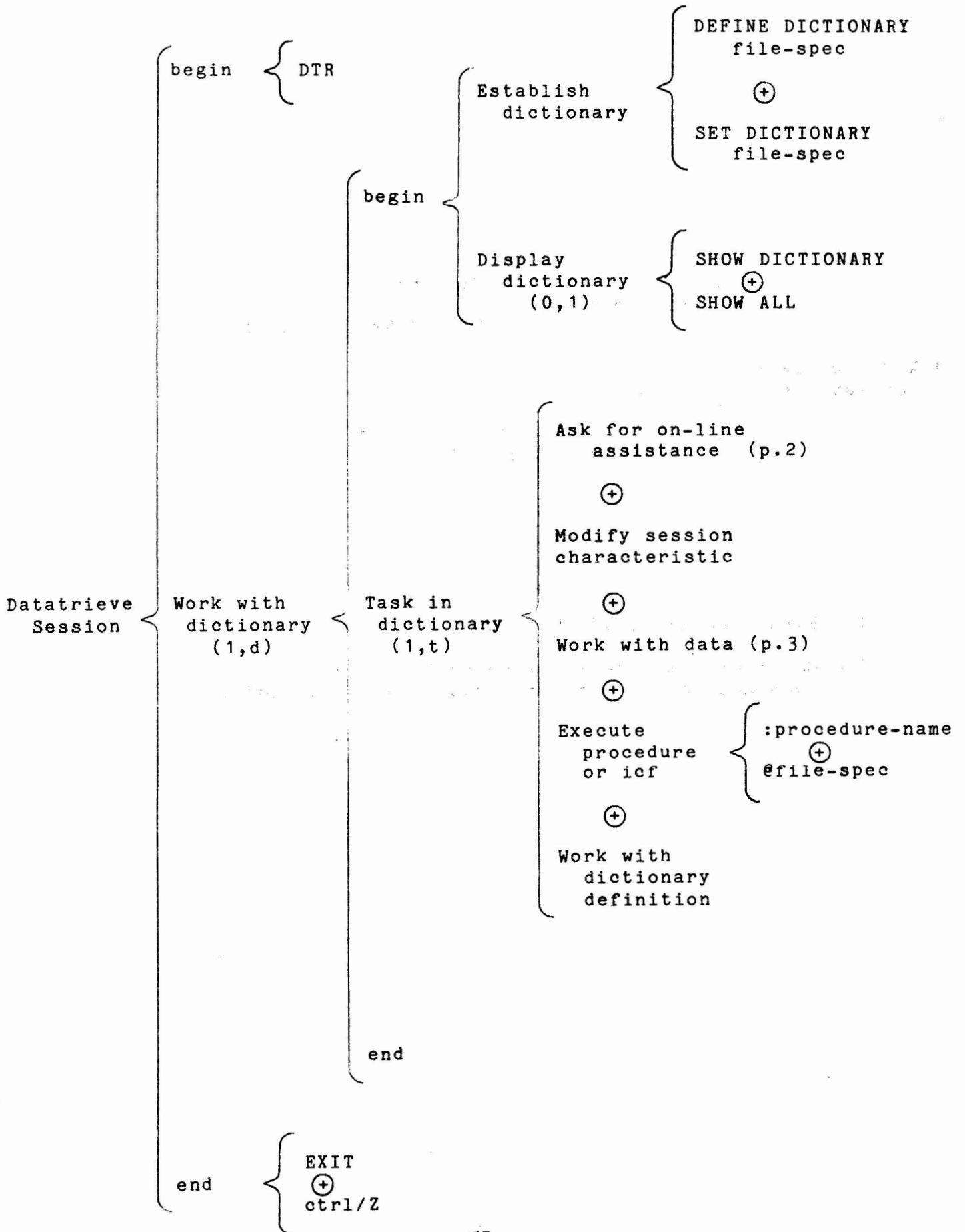
Suggested Reading:

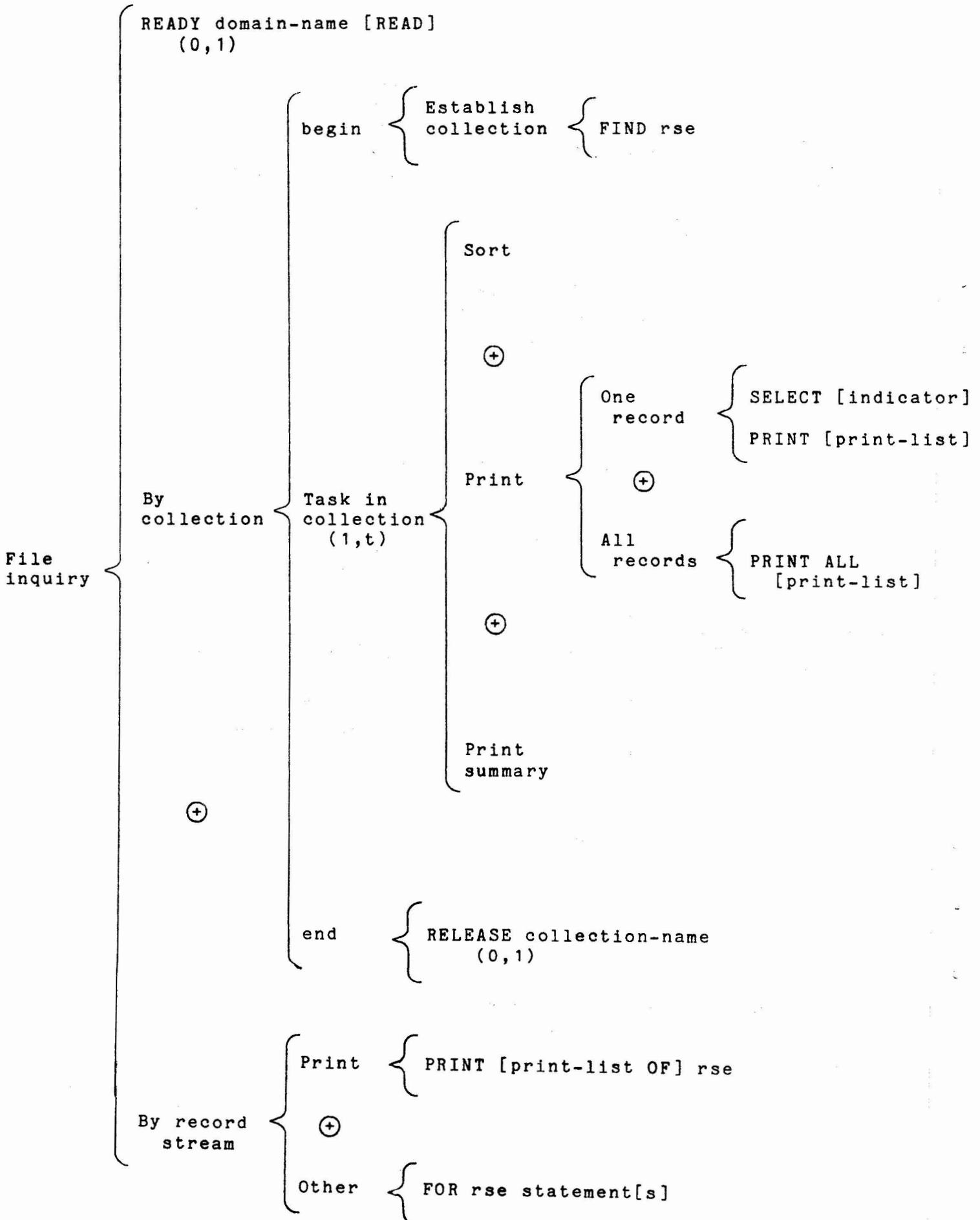
Orr, Ken; "Structured Requirements Definition;" Yourdon Press; New York; 1977.

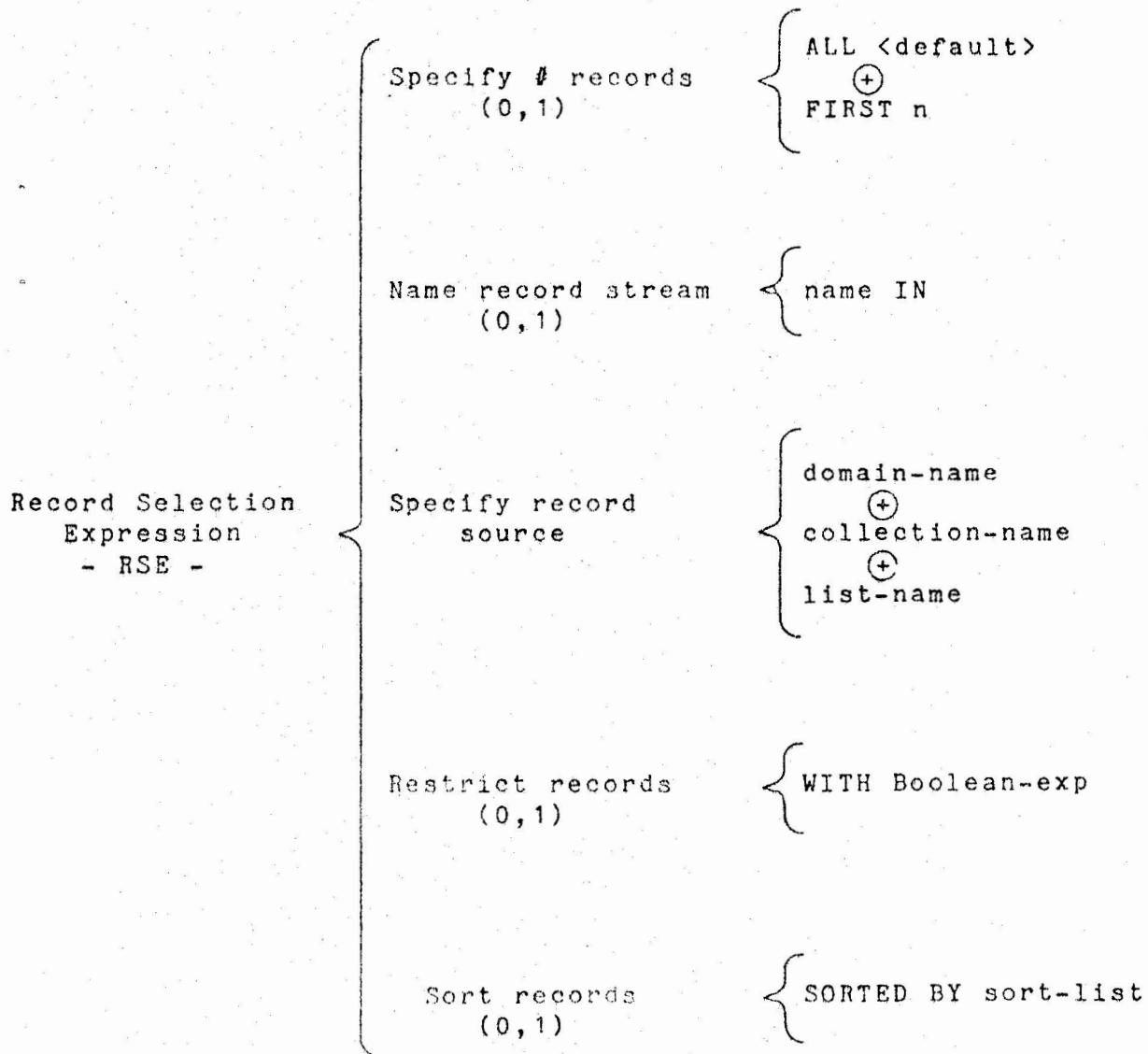
Orr, Ken; "Structured Systems Development;" Kenn Orr and Associates, Inc.; Topeka, Kansas; 1981.

Acknowledgement:

I wish to thank the staff of the Data Services and Biometrics Units of the California Department of Fish and Game for their valuable suggestions and support.







## 2.1 Defining the NEW\_HIRE Record

The figure below indicates the structure of the NEW\_HIRE record. DATATRIEVE can relate records from two domains faster if the common field is the primary key. ID, the key field, is surrounded by asterisks. You also derive performance benefits if there are no duplicate values for the primary key. Since every employee ID is unique, this condition is also satisfied.

Figure 1: Field Structure of NEW\_HIRE\_REC

```

+-----+-----+-----+-----+-----+-----+
|                                     01 NEW_HIRE_REC                                     |
+-----+-----+-----+-----+-----+-----+
* 10 ID * 10 NAME | 10 DIV | 10 LOC | 10 START_ | 10 BIRTH_ |
*          *      |         |         |         DATE |         DATE |
+-----+-----+-----+-----+-----+-----+

```

This is the full record definition:

```

RECORD NEW_HIRE_REC USING
01 NEW_HIRE_REC.
  10 ID      PIC IS 9(5)
    EDIT_STRING IS 9(5).
  10 EMPLOYEE_NAME      PIC IS X(20).
  10 COMPANY.
    15 DIVISION PIC X(10)
      QUERY_NAME IS DIV.
    15 LOCATION PIC X(12)
      QUERY_NAME IS LOC.
  10 START_DATE          USAGE IS DATE
    DEFAULT VALUE IS "TODAY"
    EDIT_STRING IS NN/DD/YY.
  10 BIRTH_DATE          USAGE IS DATE
    EDIT_STRING IS NN/DD/YY.
;

```

The field START\_DATE has a default value of "TODAY". If you store a record on the employee's first day of service, DATATRIEVE automatically stores the current system date as the value for START\_DATE. There is no need to store values for an employee's age or months employed. VAX-11 DATATRIEVE can compute these values based on the field values for BIRTH\_DATE and START\_DATE.



### 3.0 Producing a Control Group Report

The NEW\_HIRE domain contains the data for the first required report - all company employees according to division and company location. So that the report can fit on one page, I've limited it to employees in the "MARKETING" and "SOFTWARE" divisions. The steps listed below hold true for most reports that summarize information for groups of records. The first two steps are:

- (1) Identify and sort the data.
- (2) Divide detail lines into groups.

A REPORT statement operates much like the PRINT statement. But the REPORT statement has a significant advantage over the PRINT statement. Once the records are sorted, you can group similar records together with AT TOP or AT BOTTOM OF field-name statements. For example, you can enter the statement: AT TOP OF DIVISION PRINT DIVISION. DATATRIEVE processes one record at a time. Every time DATATRIEVE encounters a new value for the DIVISION field, it prints the division. In this case DATATRIEVE comes across a new value for DIVISION only twice - for "MARKETING" and for "SOFTWARE".

You can also include AT BOTTOM OF DIVISION PRINT followed by a list of print items such as COUNT. This instructs DATATRIEVE to display the number or (COUNT) of records in the previous group whenever it encounters a new value for DIVISION.

- (3) Print the detail lines.

There is one detail line for each record identified in the REPORT statement. The format and content of the detail line is specified in the PRINT statement.

- (4) Summarize data on each DIVISION.

Use the AT BOTTOM OF field-name statement.

- (5) Summarize data for the entire report.

Use the AT BOTTOM OF REPORT statement to display summary data for all of the records DATATRIEVE has processed.

```
DTR> SHOW NEW_HIRE_REPORT<RET>
PROCEDURE NEW_HIRE_REPORT
REPORT NEW_HIRE WITH DIVISION = "SOFTWARE", ----->(1)
    "MARKETING" SORTED BY DIVISION, LOCATION
SET COLUMNS_PAGE = 70
SET REPORT_NAME = "EMPLOYEE REPORT"
AT TOP OF DIVISION PRINT DIVISION ----->(2)
AT TOP OF LOCATION PRINT LOCATION ----->(2)
```

DATATRIEVE supports date arithmetic. Follow these steps:

- (6) Declare a variable for the current date. (If you include this assignment in your DTR\$STARTUP file, this variable will always have the value of the current system date.)

```
DECLARE T DATE DEFAULT VALUE IS "TODAY"
```

- (7) Define a COMPUTED BY field to calculate the age. (You can add the virtual field AGE to the record definition or you can declare a variable AGE in your report procedure.)

```
10 AGE COMPUTED BY (T - BIRTH_DATE)/365.25  
    EDIT_STRING IS Z9.
```

#### 4.1 An Employee Age Range Report

You can now sort NEW\_HIRE by the AGE field to group employees with the same age. The trouble is that employees with values for AGE of 44.1 and 44.2 will not be grouped together. To solve this problem, you can apply a DATATRIEVE function, FN\$FLOOR, to AGE. FN\$FLOOR drops off the decimal and leaves only the integer value. Now all employee records with the same age in years can be grouped together.

If you want to group employees in their forties, employees in their fifties, and so on, you can divide the AGE value by ten and then apply FN\$FLOOR. This is the key to producing the employee age report. Follow these steps:

- (8) Define a variable to compute the decade of an age. (This associates the decade of the age with each age record, allowing you to group employee records by the decade of the age. FN\$FLOOR is a truncating function that takes off the decimal part of the value.)

```
DECLARE AGE_RANGE COMPUTED BY FN$FLOOR (AGE/10) EDIT_STRING IS 90.
```

- (9) Sort records by AGE.

- (10) Group records according to AGE\_RANGE.

```
DTR> SHOW EMPLOYEE_AGE<RET>  
PROCEDURE EMPLOYEE_AGE  
DECLARE AGE_RANGE COMPUTED BY ----->(1)  
    FN$FLOOR (AGE/10) EDIT_STRING IS 90.  
REPORT NEW_HIRE WITH DIVISION = "SOFTWARE", ----->(2)  
    "MARKETING" SORTED BY DIVISION, BIRTHDATE  
SET COLUMNS_PAGE = 70  
SET REPORT_NAME = "AGE OF EMPLOYEES"  
AT TOP OF DIVISION PRINT DIVISION
```

AT TOP OF AGE\_RANGE PRINT AGE\_RANGE ("EMPLOYEES"/"OVER") ----->(3)  
 PRINT EMPLOYEE\_NAME, BIRTH\_DATE  
 AT BOTTOM OF DIVISION PRINT SKIP, COL 18,  
 "AVERAGE AGE OF EMPLOYEES IN" |||DIVISION|":", SPACE,  
 AVERAGE AGE USING 99, SKIP  
 END\_REPORT  
 END\_PROCEDURE  
 DTR> :EMPLOYEE\_AGE<RET>

AGE OF EMPLOYEES

3-Sep-1982  
 Page 1

DIVISION	EMPLOYEES OVER	EMPLOYEE NAME	BIRTH DATE
MARKETING	20	DAN ROBERTS	3/21/59
		LOUISE DEPALMA	11/20/54
		LYDIA HARRISON	8/29/54
	30	GAIL CASSIDY	8/20/47
		DAVID LITELLA	10/30/46
		JIM MEADER	9/26/44
		BILL SWAY	12/30/42
	40	THOMAS SCHWEIK	10/10/39
		ANTHONY IACOBONE	9/20/39
	60	CASS TERRY	6/28/20
		STAN WITTGEN	5/29/19
	AVERAGE AGE OF EMPLOYEES IN MARKETING :		
SOFTWARE	20	JOANNE FREIBURG	9/28/57
		RANDY PODERESIAN	6/20/57
	30	HANK MORRISON	6/29/47
		JEFF TASHKENT	9/20/43
		SY KELLER	9/01/43
		BRUNO DONCHIKOV	1/02/43

company's calendar, you can set up a monthly table with 12 entries as shown below, or a table with 365 entries. Then use FN\$MONTH or FN\$JULIAN before looking up a value via the table.

```
DTR> DEFINE TABLE QUARTER_TABLE
DFN>     QUERY_HEADER IS "QTR"
DFN>     EDIT_STRING IS 9
DFN>     1 : 3
DFN>     2 : 3
DFN>     3 : 3
DFN>     4 : 4
DFN>     5 : 4
DFN>     6 : 4
DFN>     7 : 1
DFN>     8 : 1
DFN>     9 : 1
DFN>    10 : 2
DFN>    11 : 2
DFN>    12 : 2
DFN> END_TABLE
```

To find the quarter for a date, look up the value via the dictionary table. For example:

```
DTR> PRINT T, FN$MONTH (T) VIA QUARTER_TABLE<RET>
```

```
      T      QTR
10-Sep-1982  1
```

```
DTR>
```

#### 4.4.1 Defining a Field for the Quarter

You can define a virtual field in the record to do the table lookup. Then you can sort the records by this virtual field, and develop a report that displays quarterly statistics. Here is a sample field definition:

```
05 QTR COMPUTED BY
   FN$MONTH (SALES_DATE) VIA QUARTER_TABLE
   EDIT_STRING IS 9.
```

## 5.0 Defining a Related Record

To produce the two sales reports, you need to define another record. This record must relate to NEW\_HIRE over the ID field. You need to answer these questions before defining the record and its file:

- Flat or hierarchical record?
- Primary key?
  - Simple or compound?
  - Order of Fields within the Compound key?

Assume that you intend to store one record a month for each salesperson. You could have a list field that repeated 12 times, one for each month of the year, with list items for the sales date and sales amount. But hierarchies are difficult to deal with sometimes. Alternatively, you could define a flat record and store a record for each salesperson for each month of sales. If you use a group field (SALE\_DATA) that encompasses ID and SALES\_DATE as the primary key, you do not need duplicate keys. You also have guaranteed that a data entry clerk will not inadvertently store duplicate records for an employee's monthly sales.

You could include sales records for all years in the same domain. For the sales reports, you could access only those records in which the SALES\_DATE value falls in a specified range. But you may want to store previous years records in an old domain, that you keep on tape for archival purposes. This decision can be made depending what additional reports may be needed, for example those comparing one year's sales with another year. For this example, the domain and file contains data only for the current year.

The order of the fields within the group field SALE\_DATA is also important. DATATRIEVE can only do a fast keyed access of records for queries that use the first field of the group field. Since ID will be used most often in accessing records and relating sales records with NEW\_HIRE records, you should list it first in the record definition.

With these factors in mind, a good organization of the sales record is:

Figure 2: Structure of SALES\_B2\_REC

```
+-----+
|          01 SALES_B2_REC          |
+-----+
| *      05 SALE_DATA      *      05 AMOUNT      |
+-----+
| * 10 ID * 10 SALES_ |
| *          *      DATE          |
+-----+
```

2. Set up heading for each location.
3. Group the records and calculate summary data for each employee.
4. Calculate summary data for each location.
5. Calculate summary data for all employees.

```

REPORT SALES_B2 CROSS NEW_HIRE OVER ID SORTED BY LOCATION, ID ---->(1)
SET COLUMNS_PAGE = 70
SET REPORT_NAME = "EMPLOYEE SALES REPORT"
AT TOP OF LOCATION PRINT COL 1, LOCATION, SKIP ----->(2)
AT BOTTOM OF ID PRINT COL 13, ID, COL 20, EMPLOYEE_NAME, ----->(3)
  COL 41, COUNT ("M"/"O"/"S") USING Z9,
  COL 46, TOTAL AMOUNT USING $$$, $$$,99,
  COL 60, AVERAGE AMOUNT USING $$$, $$$,99
AT BOTTOM OF LOCATION PRINT SKIP, COL 25, LOCATION|||"TOTALS:", -->(4)
  COL 45, TOTAL AMOUNT (-) USING $$$$, $$$,99,
  COL 60, AVERAGE AMOUNT (-) USING $$$, $$$,99, SKIP
AT BOTTOM OF REPORT PRINT SKIP, COL 30, "SALES FORCE:", ----->(5)
  COL 45, TOTAL AMOUNT (-) USING $$$$, $$$,99,
  COL 60, AVERAGE AMOUNT (-) USING $$$, $$$,99
END_REPORT

```

EMPLOYEE SALES REPORT

30-Sep-1982

Page 1

LOCATION	ID	EMPLOYEE NAME	M O S	TOTAL AMOUNT	AVERAGE AMOUNT
<b>MILWAUKEE</b>					
	11111	ANNE DINNAN	3	\$7,289.20	\$2,429.73
	22222	NANCY ROTHBLATT	3	\$19,293.93	\$6,431.31
	34567	LYDIA BARNET	3	\$7,926.31	\$2,642.10
	44444	WAYNE SMITH	3	\$30,053.24	\$10,017.75
	88888	JOSEPH FREDERICK	3	\$15,250.00	\$5,083.33
		<b>MILWAUKEE TOTALS:</b>		<b>\$79,812.68</b>	<b>\$5,320.85</b>
<b>PORTLAND</b>					
	23456	DAN DERRICK	3	\$34,943.45	\$11,647.82
	45678	HENRY MAILER	3	\$30,499.97	\$10,166.66
	77777	SEYMOUR KIMMELMAN	3	\$22,343.29	\$7,447.76
	99999	RICK LANGHART	3	\$15,249.97	\$5,083.32
		<b>PORTLAND TOTALS:</b>		<b>\$103,036.68</b>	<b>\$8,586.39</b>

ST LOUIS

12345	WILLIAM SULLIVAN	3	\$26,452.62	\$8,817.54
33333	LINDA REINE	3	\$26,023.27	\$8,674.42
55555	JAMES STORER	3	\$78,921.86	\$26,307.29
56789	DENNIS MCADOO	3	\$37,654.14	\$12,551.38
66666	SANDY LEVINE	3	\$30,500.03	\$10,166.68

ST LOUIS TOTALS:        \$199,551.92        \$13,303.46

SALES FORCE:        \$382,401.28        \$9,104.79

DTR>

### 6.0 Including Several Types of Detail Lines in One Report

The fourth required report displays sales commission information for every employee in the sales force. The company has established a commission schedule for the sales division.

MONTHS EMPLOYED	AMOUNT SOLD	COMMISSION PCT	RATING
GT 8	GT 10000	12%	ABOVE QUOTA
GT 8	LE 10000	7%	BELOW QUOTA
LE 8	GT 5000	10%	ABOVE QUOTA
LE 8	LE 5000	5%	BELOW QUOTA

To report a salesperson's commission, you must first determine the commission percentage. To determine the percentage, you need to test field values for MONTHS\_EMP and AMOUNT. This means that the report will have several types of detail lines depending on the outcome of the tests. Some detail lines are "12%" lines, others are "5%" lines, and so on.

If the derived values depended on only one field, you could probably define a table to associate a value for every field value. But this technique is not feasible when a detail line value depends on more than one field. When you have this type of reporting need, you can follow these general steps. Since this method involves defining and storing new records, it is not recommended for use with very large data files.

The first steps require you to determine the fields needed in the final report, and then to define a temporary record and an associated domain with that structure. For those fields

```

"SALES COMMISSION FOR MONTH ENDING"|||SLS_DATE,
SKIP 2, COLUMN_HEADER
AT TOP OF COMM_PCT PRINT COMM_PCT, RATING
PRINT SM_NAME ("NAME"), MON_EMP, AMT, COMMISSION
AT BOTTOM OF COMM_PCT PRINT SKIP 2, COL 10,
"TOTAL AT"|||COMM_PCT|"%:", SPACE, COUNT (-) USING Z9,
COL 40, "TOTAL COMMISSION:",
TOTAL COMMISSION USING $$$,$$$99, SKIP 2
AT BOTTOM OF REPORT PRINT COL 10,
".....",
SKIP, COL 10, "TOTAL SALESMEN:", SPACE, COUNT (-) USING Z9,
COL 35, "GRAND TOTAL COMMISSION:",
TOTAL COMMISSION USING $$$$,$$$99
END_REPORT

```

!Delete the temporary domain and record

```

DELETE TEMP,TEMP_REC;
END_PROCEDURE

```

```

DTR> :SALES_COMM<RET>
[Record is 50 bytes long.]
Enter month in numbers: 2<RET>

```

SALES COMMISSION FOR MONTH ENDING 28-Feb-1982

COMM PCT	RATING	NAME	MON EMP	AMT	COMMISSION
5%	BELOW QUOTA				
		ANNE DINNAN	6	\$2,150.91	\$107.55
		LYDIA BARNET	4	\$2,338.91	\$116.95
		JOSEPH FREDERICK	7	\$4,500.00	\$225.00
		RICK LANGHART	6	\$4,499.99	\$225.00
		TOTAL AT 5%:	4		
				TOTAL COMMISSION:	\$674.49
7%	BELOW QUOTA				
		WILLIAM SULLIVAN	11	\$7,805.69	\$546.40
		HENRY MAILER	9	\$8,999.99	\$630.00
		LINDA REINE	9	\$7,679.00	\$537.53
		SANDY LEVINE	10	\$9,000.01	\$630.00



TOTAL AT 7%: 4 TOTAL COMMISSION: \$2,343.93

10% ABOVE QUOTA

WAYNE SMITH	8	\$8,868.17	\$886.82
NANCY ROTHBLATT	5	\$5,693.29	\$569.33
SEYMOUR KIMMELMAN	7	\$6,593.10	\$659.31

TOTAL AT 10%: 3 TOTAL COMMISSION: \$2,115.46

12% ABOVE QUOTA

JAMES STORER	16	\$23,288.42	\$2,794.61
DAN DERRICK	10	\$10,311.18	\$1,237.34
DENNIS MCADOO	14	\$11,111.06	\$1,333.33

TOTAL AT 12%: 3 TOTAL COMMISSION: \$5,365.28

\*\*\*\*\*

TOTAL SALESMEN: 14 GRAND TOTAL COMMISSION: \$10,499.15

DTR>

## 7.0 Budget and Expense Reports

There is more than one way to include data from two or more domains in one report. The last two examples used the CROSS clause to form one record stream that consisted of expanded records from two domains that shared a common field value. But you can also produce a report based on two separate record streams, by using inner print lists in the PRINT and AT statements.

This technique allows you to produce a report comparing a budget with actual expenses. You achieve this result with the CROSS clause.

It is common practice in both business and personal finance to develop a budget. When an expense is incurred, it is charged to a particular category of the budget. At periodic intervals, you may wish to compare each category of the budget with the actual expenditures. A report of the budget should indicate all the expenses for each category, and whether you are over budget or under budget in each category.

To compare budgeted expenses with actual expenses, you need to relate two separate record sources. One domain contains a record for each category of the budget. A second domain is used to store records every time an expenditure is made.

DTR> :CHECK\_RPT<RET>

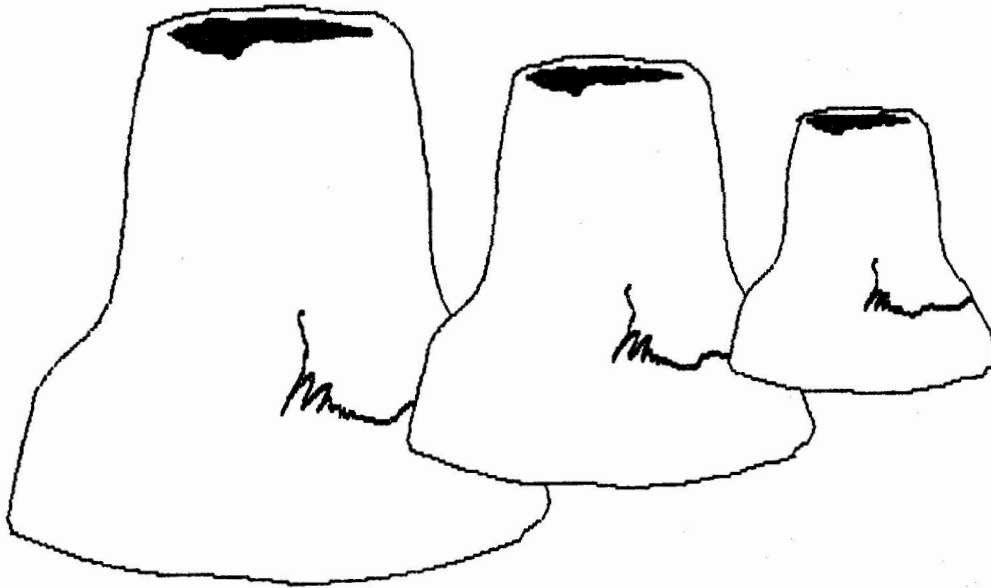
20-Sep-1982

Page 1

CATEGORY	DESC	CHECK DATE	AMOUNT	BUDGET	DIFFERENCE
1	EDUCATION	15-Jun-1982	\$123.00		
		Category 1 Totals:	\$123.00	\$500.00	\$377.00+
2	RECREATION	15-May-1982	\$134.00		
		Category 2 Totals:	\$134.00	\$25.00	\$109.00-
3	CLOTHING	15-Oct-1982	\$243.00		
		Category 3 Totals:	\$243.00	\$250.00	\$7.00+
4	FOOD	15-Jan-1982	\$126.00		
		15-Aug-1982	\$234.00		
		17-May-1982	\$239.00		
		16-May-1982	\$543.00		
		Category 4 Totals:	\$1,142.00	\$150.00	\$992.00-
5	MORTGAGE				
		Category 5 Totals:	\$ .00	\$550.00	\$550.00+
		TOTALS:	\$1,642.00	\$1,475.00	\$167.00-

DTR>

**Drawing Pictures With Datatrieve  
Wayne Jones, Graybeard  
Supervisor of Software Documentation  
DISG Documentation  
Digital Equipment Corporation**



This Datatrieve procedure draws a picture of a vase by California potter, Robert Turner. I used it to design part of a cover for THE STUDIO POTTER. It generates ReGIS code that can control the graphics display of a VT125 terminal. The data points were stored and kept in a Datatrieve domain.

Because it contains the <ESC> character, you must define the procedure with an editor such as VAX-11 EDT (or the VAX-11 Datatrieve Editor): press the ESC key twice to insert the <ESC> character into the procedure. You must also use such an editor to read the procedure after it is stored in the data dictionary: use the Extract command to create a file containing the definition and read the file with the editor.

```

PRINT COL 1, "P["|(FN$NINT(X * A) + C)|", "|||(FN$NINT(Y * B) + D)|"]",
SKIP, COL 1, "C(B,W(I1))", SKIP, COL 1, "["
!
!GET POINTS FOR OUTLINE
!
FOR P1 WITH SEQ BT 3 AND 91
PRINT COL 1, "["|(FN$NINT(X * A) + C)|", "|||(FN$NINT(Y * B) + D)|"]"
!
!RETURN CURSOR TO STARTING POINT
!
FOR P1 WITH SEQ = 1
PRINT COL 1, "["|(FN$NINT(X * A) + C)|", "|||(FN$NINT(Y * B) + D)|"]"
!
!END OUTLINE
!
PRINT COL 1, "[", SKIP, COL 1, "(E)", SKIP 2
!
!SET CURSOR FOR START OF HOLE
!
FOR P1 WITH SEQ = 92
PRINT COL 1, "P["|(FN$NINT(X * A) + C)|", "|||(FN$NINT(Y * B) + D)|"]"
!
!CURVE COMMAND TO CONTROL SHADING OF HOLE
!
FOR P1 WITH SEQ = 92
PRINT "C(B,W(S1,E,S["|||(FN$NINT(Y * B) + D)|"])", SKIP, "[", SKIP
!
!GET POINTS FOR THE HOLE
!
FOR P1 WITH SEQ BT 93 AND 111
PRINT COL 1, "["|(FN$NINT(X * A) + C)|", "|||(FN$NINT(Y * B) + D)|"]"
!
!END HOLE
!
PRINT COL 1, "[", SKIP, COL 1, "(E)", SKIP 2
!
!START FIRST SQUIGGLE
!
FOR P1 WITH SEQ = 112
PRINT COL 1, "P["|(FN$NINT(X * A) + C)|", "|||(FN$NINT(Y * B) + D)|"]",
SKIP, COL 1, "C(S,W(S0,E))", SKIP, COL 1, "["
!
!GET POINTS FOR FIRST SQUIGGLE
!
FOR P1 WITH SEQ BT 113 AND 143
PRINT COL 1, "["|(FN$NINT(X * A) + C)|", "|||(FN$NINT(Y * B) + D)|"]"
!
!END FIRST SQUIGGLE
!
PRINT COL 1, "[", SKIP, COL 1, "(E)", SKIP 2
!
!START SECOND SQUIGGLE
!

```

```

FOR P1 WITH SEQ = 112
PRINT COL 1,
  "P["|(FN$NINT(X * A) + C)|", "|||(FN$NINT(Y * B) + (D - 2))|"]",
  SKIP, COL 1, "C(S,W(S0,E))", SKIP, COL 1, "["
!
!GET POINTS FOR SECOND SQUIGGLE
!
  FOR P1 WITH SEQ BT 113 AND 143
  PRINT COL 1,
    "["|(FN$NINT(X * A) + C)|", "|||(FN$NINT(Y * B) + (D - 2))|"]"
!
!END SECOND SQUIGGLE
!
  PRINT COL 1, "[", SKIP, COL 1, "(E)", SKIP 2
!
!EXIT REGIS
!
  PRINT COL 1, "<ESC>"
CLOSE
END-PROCEDURE

```

The results of this procedure are shown on the next page.

AMERICAN	26	26	\$9,895
AMERICAN	26-MS	26	\$18,895
BAYFIELD	30/32	32	\$32,875
BLOCK I.	40	39	NO PRICE AVAILABLE
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

### Important Notes:

The first statement "Report ... X ...Yachts" defines a context "X" to be used in the report statements.

The expression "Builder = X.Builder and Model = X.Model" defines a unique record in Yachts (the same record as the one being printed).

If the Price of the above record is 0 then "No Price Available" will be printed, if the Price is not 0 then the Price will be printed.

The importance of the parentheses in "(Price = 0)" has been explained in a previous issue.

The RSE used in the PRINT statement above must refer to a UNIQUE record in Yachts for the report to work correctly.

If this technique is used with large files it is important that the RSE in the Print statement use indexed key fields or the Report will take an excessive amount of time to generate.

**Example 2.** Example 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)

```

DELETE LOA-TABLE;
DEFINE TABLE LOA-TABLE
30:"Just under Ideal",
31:"IDEAL LENGTH",
32:"Just over Ideal"
END-TABLE

```

The statements to achieve this result are:

```

REPORT FIRST 30 Y IN YACHTS
PRINT BUILDER ,MODEL ,
ALL COL 32 ,"Too short" ,COL 50 ,PRICE OF YACHTS WITH
BUILDER EQ Y.BUILDER AND MODEL EQ Y.MODEL AND ( LOA LE 29 ) ,
ALL COL 32 ,LOA VIA LOA-TABLE USING X(16) ,COL 50 ,PRICE OF YACHTS WITH
BUILDER EQ Y.BUILDER AND MODEL EQ Y.MODEL AND ( LOA BT 30 AND 32 ) ,
ALL COL 32 ,LOA ,COL 50 ,PRICE OF YACHTS WITH
BUILDER EQ Y.BUILDER AND MODEL EQ Y.MODEL AND LOA GE 33
SET REPORT-NAME = "YACHTS WITH LOA CATEGORIZED"
END-REPORT

```

YACHTS WITH LOA CATEGORIZED

22-Feb-63

Page 1

MANUFACTURER	MODEL	ALL	LENGTH OVER 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	

All of these options are defined in the file QD.MAC which is part of the DTR installation package and should be defined by editing this file before DTR is task-built. Please consult your System Manager if any of the above options are not set as you would prefer.

### Backup of DTR files

If the DEC utility BRU is used to backup disk files on the system there are 2 important points to be aware of:

- (1) BRU has a switch for daily backup to determine whether files created today or those modified today should be copied for backup. If an editor such as EDT is used to modify a file, in most cases a new version of the file is created. However if DTR is used to MODIFY, STORE or ERASE data in a file, a new version of the file is NOT created, but the existing version is modified. If the system only copies files created (not modified) today then any changes made to DTR data files will NOT be backed up. Ask your System Manager to please use the "Files Modified" switch on BRU.
- (2) BRU has consistently (even after several patches) had problems backing up and subsequently restoring large multi-header files. For several reasons large files (especially if indexed) created and added to using DTR are almost certainly of the type that BRU will not restore correctly. Please discuss this problem with your System Manager and if there any doubts, ask him to restore a large DTR file that was previously backed-up and then examine the file for corruption!

### Use of IFL

The DEC utility RMSIFL is used to populate existing empty indexed files in an optimised manner. IFL has at least 2 very important uses for DTR data files (I will not explain the details here):

- (1) After an indexed file (and any file with more than probably a couple of hundred records should usually be indexed) has been through a "reasonable" amount of STORE and ERASE using DTR, its storage may be optimised and subsequent access to the file made more efficient using IFL as follows:
  - Using DTR, DEFINE FILE with index keys and ALLOCATION defined.
  - At the operating system level: IFL file;new-version=file;old-version
  - After checking the new version of the file with DTR, delete the old.
- (2) Adding a new field to an indexed file may be done with DTR, but it is easier to use IFL as follows (assuming the new field may be added at the very end of the record which is usually true since the fields may be printed in any order desired):
  - Edit the record definition to add the new field at the end.
  - Using DTR, define a new file (indexed) for the domain.
  - At the operating system level: IFL newfile/PD=oldfile
  - will copy the old records (padding them with blanks to the new record length) into the new file and optimise the storage at the same time.



## Conditional Printing Using Print All All Statements

**Alexander J. Annala, Data Base Specialist  
The Salk Institute for Biological Studies  
P.O. Box 85800, San Diego, California 92138  
(619) 453-4100 x 104 (Direct) x 441 (messages)**

The EXAMPLE.COM file and sample listing displayed below demonstrate a technique for selectively including and/or excluding individual print list elements based upon boolean conditionals. The technique is simple yet elegant.

First, we need a single element list. The list may be a domain or a collection which contains only one record. If the domain/collection name is SINGLE then:

- (A) PRINT ALL "XXX" OF SINGLE will display "XXX"
- (B) PRINT ALL "XXX" OF SINGLE WITH (1 EQ 1) will display "XXX"
- (C) PRINT ALL "XXX" OF SINGLE WITH (1 EQ 0) will display nothing

Why? In all three examples, the string "XXX" is displayed for each element of the RSE (record selection expression). In example (A) the RSE consists of all records in the SINGLE domain/collection. Because the SINGLE domain/collection contains one and only one record, one and only one line will appear containing the string "XXX." In example (B) the RSE consists of all records in the SINGLE domain/collection where the boolean conditional (1 EQ 1) is true. Because the boolean conditional (1 EQ 1) is always true, no records are excluded from the RSE, and one and only one line of output will appear. In example (C) the RSE consists of all records in the SINGLE domain/collection where the conditional (1 EQ 0) is true. Because the boolean conditional (1 EQ 0) is always false no records are contained in the RSE, and no output will appear.

A field name/variable name may be included on the RIGHT SIDE of the boolean conditional. Assume STR is a string variable containing "TRUE". Then:

- (D) PRINT ALL "XXX" OF SINGLE WITH ("TRUE" EQ STR) will display "XXX"
- (E) PRINT ALL "XXX" OF SINGLE WITH ("TRUE" NE STR) will display nothing

If the field name/variable name is included on the LEFT SIDE of the boolean expression Datatrieve will generate an error message.

The following EXAMPLE.COM file and sample output listing contains two parts. The first part establishes a permanent disk file containing a single record which in turn contains a single element list. The second part is a procedure which uses that disk file to solve the conditional print problem posed by Philip Dickerson in the Wombat Examiner 4:3 (November 1982), pp. 26-28. In short, Mr. Dickerson challenged his audience to write a simple procedure to print the manufacturer of each yacht, the model, a value from the following table:

```
LOA LE 29 "Too short"  
LOA EQ 30 "Just under ideal"
```

---

## Wombat Magic

---

Preface:

I was sitting in my office in the hills above the San Francisco Bay, watching the hawks circle Strawberry Canyon, and wondering about ways in which to preface the transcript of the Magic session.

Discretion being the safer part of valor, I decided to simply say that at each DECUS, time is set aside for a strange ritual called the Datatrieve Magic Session. During the four hours preceding midnight, various strangers throw themselves to the mercy of a large audience to share interesting subversions. Things being what they are, the combination of spirits, the late hour, and the DECUS phenomenon known as *cerebrum saturatum*, the magical descriptions tend toward the higher end of the wizardly scale.

In order that everyone have the benefit of these acts of sleight of hand the magic was recorded and translated. The result is the set of pages you see before you.

When making your way through them, you must keep in mind that the direct transcription of the spoken word is frequently found to be less communicative than the written word; this is why writers invented "dialogue." Also, words like "here," "this one," and "at the top" usually refer to the overhead. You'll have to position yourself.

I have blended in the various overheads but have not edited any of the verbiage; the act of editing, in this instance, comes under the context of creating documentation, which is better left to Digital.

The entire evening is not included in this issue, but will be completed in the next. Two of the expositions have been expanded and appear elsewhere in this issue.

The Editor

---

---

What we're going to be doing tonight, is some wizardry, and magical things that will amaze and mystify everyone. A long time ago, Jim Starky said Datatrieve was strictly a query report language and would never really amount to much more than that. I remember Gary (Saxer) telling him that he had written a complete financial system and Jim said, "you're crazy, it can't be done," and went "ha ha ha ha." I knew they'd do it someday.

Well, we've proved in the last four years that you can do things with Datatrieve

that are impossible. This is where we show some of these impossibilities off. We'd like to show you some of the stuff that we've done in the past years because we have so many newcomers here and we want to show off some of the past magic. When we talk about magic it doesn't have to be something earth-shattering or startling. It can be something very, very simple that we use every day that's just kind of wierd or out of the ordinary, and we'd like to see that. We've got some transparencies up here for those of you who haven't prepared their magic. We'd like you to grab a pen and jot it out and then we'll give everybody a chance at the screen to explain what it is and what it does.

I think a couple of years ago, one of the best ones was a kid that was running a bookie operation using Datatrieve. There were two cops in the room and they wanted copies, his name and address. I don't know whatever came of that. They aren't here this time, but then neither is he. I don't know if that's significant.

At the end of the evening we're going to pick the super wizard of the evening and this framed picture will go to the super wizard of the evening. Now, this is the original silkscreen layout that our shirts were made from, so its going to be a real collector's item, so we're going to give that out. Now, a little later on, I know everybody is thirsty and there's not much water here, so we've made some arrangements, the Magic Wombat in the sky is going to bring us some goodies, so be patient and we should have about twenty cases of goodies coming in any time.

(Presented by Bill Noyce, Digital)

"This one isn't mine, I can't take credit for it. It's actually due to Ann Harrison, who couldn't be here. But there are some things that everybody you know says you can't do with Datatrieve. One of them is make the Report Writer put out a two column list. Here we have a header at the top of the page, the date and page number, two columns, in this case it's a mailing list, and another header down at the bottom of the page.<sup>1</sup>

WOMBAT MAGIC - a two column list

1-DEC-1982  
Page 1

Alan Anderson	AL01-7/J05	Ed Jkymans	ED01-7/J05
Melinda Benult	ME01-7/J05	Fred Kamrs	FR01-7/J05
Hug Boodat	HU01-3/J05	Jeff Lumomatso	JE01-5/J05
Jim Berdane	JH01-1/J05	<del>Scott Mannc</del>	SC01-7/J05
Tim Carbot	TO01-7/J05	Dennis Nitzmier	DE01-7/J05
<del>John Donbar</del>	<del>JOO1-3/J05</del>	<del>Mill Oarker</del>	MA01-7/J05
Neal Donber	NE01-8/J05	Peter Pims	PE01-7/J05

<sup>1</sup>The names and mail stops are fictitious; the list has been shortened because we'd run out of space if we didn't.

copy the input domain into the report. And this actually, strictly speaking, should have a sorted by clause on it, but it works, it comes out sorted because the other thing was a keyed file, and we put name in the name field and the mail stop in the mail stop field, and you can see that we're using running count, so this exact version would only run on VAX, but I think with the PDP11 you could initialize a variable and keep a count yourself, so this can be made to run on the 11 as well. And, this is computing page number that this record is going to end up on, takes a running count and divides it by twice the page length because we get two columns in every page. O.K., now comes the magic. You go through this domain and I'm gonna do interesting stuff for every B that is going to end up being a second column of some line. So, I say for B in temp, which is going to examine all the records, but I'll immediately ignore about half of them, because I say for A in temp with record count equals B.record count minus page length. So, for the first forty of those there aren't any matching A's so it skips right out, for the second forty, it finds the A that ought to be the front of the line where this B is the back of it, and then it modifies the A record to put the B information in there, and after it's done it erases the B record. So, now I have, if you think about it you can figure out how the first page gets done, I run through 40 records, up to forty records, that are gonna be on the second column and move them to the right first column and throw the records away, and it works out that since I've erased those, the second page works out just swell. Then all I have to do is report on this thing. At top of report I print the header, at the bottom of the page I print a footer, and I print my columns however I feel like it, and once you've done the magic the report's pretty easy."

---

---

"I'm Al Watson and I am an occasional Datatrieve user. So when I usually get into it it is mostly to do some screwy hack rather than anything ordinary. We had an application where we were storing options for users of a system in a file, in a list; the file for each user of the system had the user name, account number, and a bunch of other things, and then, at the last there was an option table, or item table which occurred a number of times depending upon a variable. We developed some procedures to add new options to that list, to delete them from that list, and we had another program in, I think, C, which wanted to see that list in sorted order. We thought, gee it would be nice if Datatrieve would sort it for us. So, I jumped in and developed a way of sorting the items in a list at the end of a record in place. And it may not be magic to those who use Datatrieve a lot, but it's magic to me because I still don't understand why it works. I just sat at the terminal all night playing with it until finally it happened, and then I said that's it and I left it that way.

```
DEFINE RECORD USER-REC USING
01 USER-REC.
  05 LAST-NAME PIC X(30).
  05 -
  .
  .
  05 NUMBER-OF-ITEMS PIC 9(3).
  05 ITEM-TABLE OCCURS 0 TO 100 TIMES DEPENDING ON NUMBER-OF-ITEMS.
```

## 10 ITEM PIC X(3)

The record definition up at the top is basically irrelevant in the sort part of it, except for the user's last name, and then at the end of the record the number of items and the item table occurring zero to one hundred times depending upon the number of items, and the item was just a three character option name. The procedure to sort it:

```
DEFINE PROCEDURE SORTLIST
DECLARE USERNAME PIC X(30).
USERNAME = *,"user's name"
READY ACCOUNT-USER MODIFY
FIND ACCOUNT-USER WITH LAST-NAME = USERNAME
SELECT
FIND SRT IN ITEM-TABLE SORTED BY ITEM
FOR ACCOUNT-USER WITH LAST-NAME = USERNAME
MODIFY USING ITEM-TABLE = SRT
FINISH
END-PROCEDURE
```

I declared a temporary variable "username" and then asked for that name, readied the domain, and then I found the record by the username and selected it. Now, the next part was, I don't know whether that's a view, or what it is, I defined SRT in item table, sorted by item. It worked. Does anybody know what that is and can explain what I did? I don't know. Alright, what it does is it reads the list and sorts it and puts it in some temporary thing called SRT, which I just named there, and then you say for account\_user with username equal username, the last name equal username, so it's again finding the record with the for statement, modify using item table equals SRT, and it just transfers the whole sorted list into the list in the record, finish, and then procedure."

---

---

(Fred Van Itallie)

I'm a chemist for Dupont. In fact, I share an office with the gentleman who was up a few moments ago, to whom I would like to wish a very Happy Birthday! I, like he, get somewhat frustrated at times when I think Datatrieve ought to be able to do things things that, gee, I thought of that, it certainly hasn't thought of that. And I confess that at least for the time-being, I'm constrained by Datatrieve 11, don't have a VAX. Now anybody who has Datatrieve 11 knows that there are certain facilities that you have with it, like GUIDE mode. How many people use GUIDE mode? That's three more than I expected. How many people use ADT? Well, that's about six more than I expected. Alright, how many people use edit? How many people use EDT otherwise? How would you like to have EDT with Datatrieve 11? Me too! I use some other languages I confess, I don't want to mention them, and found that there's a very nice thing that you can do with everything, except Datatrieve. And that's when you want to step out of a language and do something like edit, you just hit control C, and by golly, here you are at DCL and you type in your stuff and go along."

10 ADDR FROM CUSTS.

PRINT PC OF PART-CUST  
(works)

PRINT SER, NAME, ADDR OF PART-CUST

Field "name" is undefined or used out of context.

To fix this, add ICUST to view:

10 ICUST FROM CUSTS.

This is a fun one. O.K., this is a really, kind of a messy hack, I guess if you're really desperate for it, you can use it, but some people want to be able to refer to the inner parts of the print list, like you want the third kid or the fourth kid or the fifth kid, and you can't do that, you can't say enter report print first kid of families. O.K., so the only way to do this is every time you do a store routine what you've gotta is go ahead and add a little index, see, kid number and then kid name and kid age, initialize them as kid 1, kid 2, kid 3, and kid 4. Then you can say print father all kid name of kids with kid number equal three. This is of kind of messy but I guess some people are desperate to do it this way. That's it."

10 KIDS OCCURS 0 TO 5 TIMES DEPENDING ON NUMBER-OF-KIDS  
15 EACH-KID.  
20 KID-NUM PIC 9.  
20 KID-NAME PIC X(8).  
20 AGE PIC 99.

Initialize kid-num with an index number (use 1,2,3, etc.)

FOR FAMILIES WITH . . . .  
PRINT FATHER, ALL KID-NAME OF KIDS WITH (KID-NUM = 3),  
ALL KID-NAME OF KIDS WITH (KID-NUM = 2)  
ALL KID-NAME OF KIDS WITH (KID-NUM = 1)

FATHER	KID-NAME	KID-NAME	KID-NAME
NICE GUY	KID3	KID2	KID1
BAD GUY	BKID3	BKID2	BKID1

---

(Bill Noyce, Digital Equipment Corporation)

"Alright. Sometime today, I don't remember if it was morning or afternoon, Dan stood up and said one of the things that he's thinking of adding in the distant future to VAX Datatrieve is exponentiation, A raised to the B power. You can get that today, although it's not so pretty on VAX, by defining a user defined function. And you can either use the one in the library that raises a number to a power, or you can build it out of exponentials and logs. And that's pretty straightforward. But, I guess there's still some people that have PDP 11's. In the last Wombat Examiner, for those of you who are Sig members, Bart had a bunch of different functions, a bunch of different Datatrieve procedures to compute exotic functions. But, I guess he left this one out. It's not real hard.

Exponentiation  $A^{**}B$

On VAX FN\$POWER(A,B) or  
FN\$EXP(B\*FN\$LOG(A))

---

On PDP-11, if B is an integer:

```
DECLARE X USAGE REAL.  
DECLARE Y PIC 9(4) USAGE COMP.  
DECLARE RESULT USAGE REAL.  
X = A  
Y = B  
RESULT = 1  
  
WHILE Y NE 0  
  BEGIN  
    IF (Y/2) * 2 NE Y THEN RESULT = RESULT * X  
    Y = Y/2  
    X = X * X  
  END  
  
PRINT RESULT
```

We're gonna declare some temporary variables here, X is a real number that's gonna accumulate powers of A, and Y is an integer which is going to keep track of the power that I'm raising it to. And result is going to accumulate the result. So, we start off with X equal to A, Y equal to B, the result equal to 1. And what I want to compute, remember, is A raised to the B power. O.K., A raised to the power of zero, of course, is 1. So, if Y ever gets to be equal to zero, I'm done. If Y is not equal to 0, then I have to do some multiplying, and what I'm going to do is take whatever's in X and multiply it into my result to make the result bigger. What I'm going to do is, if Y is an odd number then I'm gonna multiply X into the result. If Y is an even number I'm gonna skip that step, and then I'm gonna shift Y to the right, consider it as a binary number, and square X, so in a sense the power of X is shifting to the left, so Y and X together represent the number that I want it raised to the power of, and every time a bit falls off to the right, I'm gonna multiply in an X to my result. And that's all there is to it."

DATATRIEVE SIG  
OPERATING PROCEDURES  
30 NOVEMBER 1982

ARTICLE I - NAME

- 1.0. The name of the organization is the Datatrieve Special Interest Group.

ARTICLE II - PURPOSE AND AFFILIATION

- 2.0. The Datatrieve special interest group is established as a special user group under the bylaws of the DECUS/U.S. Chapter.
- 2.1. The SIG is established, as empowered by the DECUS Bylaws, to serve members having a common interest to:
- 2.1.1. Promote the interchange of information concerning the utilization of software and computers manufactured and marketed by the Digital Equipment Corporation (DIGITAL).
- 2.1.2. Provide channels to facilitate the exchange of information pertaining to Datatrieve and its applications among SIG members.
- 2.1.3. Provide feedback to DIGITAL on equipment, software, and other user needs as they arise.

ARTICLE III - MEMBERSHIP

- 3.0. SIG membership shall be open to any DECUS Installation or Associate Member worldwide.
- 3.1. SIG membership shall be established with a formal application to the DECUS/U.S. Chapter.
- 3.2. The Steering Committee shall have the right to revoke SIG membership for cause.

ARTICLE IV - SPECIAL INTEREST GROUP STRUCTURE

- 4.0. The overall structure of the SIG shall be:
- 4.0.1. Steering Committee
- The Steering Committee is responsible for governing the activities and direction of the SIG, and shall be composed of both voting and non-voting members.



4.0.2 Chairman

The Chairman shall be chief executive and operational officer of the SIG.

ARTICLE V - STEERING COMMITTEE

5.0. The Steering Committee shall function as the governing body of the SIG.

5.1. Responsibilities

5.1.1 The Steering Committee shall establish procedures and policies for orderly operation and development of the group.

5.1.2. The Steering Committee shall appoint working groups and sub-committees on an ad-hoc or permanent basis, as required.

5.1.3. The Steering Committee shall represent SIG members with respect to DECUS and DIGITAL.

5.1.4. The Steering Committee shall effect liaison with the DECUS Symposium, Standards, Publications, and Library Committees.

5.1.5. The Steering Committee shall effect liaison with any other DECUS generic groups with respect to Datatrieve and related data management topics.

5.2. Composition

5.2.1. The Steering Committee shall consist of the persons appointed to the following positions:

5.2.1.1. Chairman Duties described in Article 6.1.

5.2.1.2. Past Chairman

The Immediate Past Chairman shall hold a position on the Steering Committee. His role is advisory to the Chairman, and he shall perform such duties as arranged with the Chairman.

5.2.1.3. Symposium Coordinator

Duties described in Article 6.2.

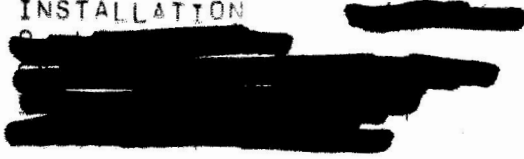


DECUS

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

BULK RATE  
U.S. POSTAGE  
PAID  
PERMIT NO. 129  
NORTHBORO, MA  
01532

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.

RECEIVED

APR 14 1983

ACCOUNTING DEPT.