

1961-1986



Leading
Toward
The
Future

DECUS U.S. CHAPTER SIGs NEWSLETTERS

ARTIFICIAL INTELLIGENCE SIG	AI
BUSINESS APPLICATIONS SIG	BA
COMMERCIAL LANGUAGES SIG	CL
DAARC SIG	DAR
DATA MANAGEMENT SIG	DMS
DATATRIEVE/4GL SIG	DTR
EDUSIG	EDU
GRAPHICS SIG	GRA
HARDWARE MICRO SIG	HMS
IAS SIG	IAS
LANGUAGES AND TOOLS SIG	L&T
LARGE SYSTEMS SIG	LS
MUMPS SIG	MMP
NETWORKS SIG	NTW
OFFICE AUTOMATION SIG	OA
PERSONAL COMPUTER SIG	PC
RSTS SIG	RST
RSX SIG	RSX
RT SIG	RT
SITE MANAGEMENT & TRAINING SIG	SIT
UNISIG	UNI
VAX SYSTEMS SIG	VAX
LIBRARY INFORMATION SECTION	LIB
"HOW TO" SUBMIT AN ARTICLE . . . GROUP FORMS	HOW
QUESTIONNAIRE SECTION	QU

Printed in the U.S.A.

"The Following are Trademarks of Digital Equipment Corporation"

ALL-IN-1	J-11	RSX-11 M-PLUS
A-to-Z	LN03	RT-11
CTS	MicroVAX	RX-02 (et al)
DATATRIEVE	OS/8	TK50
DATATRIEVE-11 (et al)	PDP-11	UNIBUS
DDCMP	PDP-11/24 (et al)	VAX
DEC	PDT	VAX-11/730 (et al)
DECalc	PRO	VAX DATATRIEVE
DECgraph	PRO/DATATRIEVE	VAX/VMS
DECnet	Q-bus	VMS
DECslide	Rdb/VMS	VT200
DECtape	ReGIS	WPS
DECUS	RMS	WPS-PLUS
IAS (et al)	RSX	WPS-PLUS/VMS (et al)

Copyright© DECUS and Digital Equipment Corporation 1986
All Rights Reserved

The information in this document is subject to change without notice and should not be construed as a commitment by Digital Equipment Corporation or DECUS. Digital Equipment Corporation and DECUS assume no responsibility for any errors that may appear in this document.

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 of liability for articles or information appearing in the document. The views herein expressed are those of the authors and do not necessarily express the views of DECUS or Digital Equipment Corporation.

IBM, IBM360 are trademarks of International Business Machines; MS-DOS, is a trademark of Microsoft Corporation; Ada is a trademark of Department of Defense; TSX-PLUS is a trademark of Computer Systems, Inc.

Production Staff:

Mark Grundler: Communications Committee Chair
Palmer College

Don Golden: SIG Publications Chair
Shell Oil

Judy Arsenault: Publications Manager
DECUS

Judy Tessier: Phototypesetter/Graphics Designer
DECUS

Circulation: 9436

GENERAL TABLE OF CONTENTS

SECTIONS	PAGE NO.
ARTIFICIAL INTELLIGENCE SIG	
. Steering Committee Listing	AI-i
BUSINESS APPLICATIONS SIG	
. Steering Committee Listing	BA-i
COMMERCIAL LANGUAGES SIG	
. Steering Committee Listing	CL-i
DATA ACQUISITION ANALYSIS, RESEARCH AND CONTROL SIG	
. Steering Committee Listing	DAR-i
DATA MANAGEMENT SIG	
. Steering Committee Listing	DMS-i
. VAX Rdb/VMS Guidelines & Standards – Part II	DMS-1
DATATRIEVE/4GL SIG	
. Steering Committee Listing	DTR-i
. Chairman's Corner	DTR-2
. Integrating DATATRIEVE with DECalc and DECGraph	DTR-3
. Commonly Asked DATATRIEVE Questions and Answers	DTR-15
. Preparing DATATRIEVE Plots for Printing	DTR-23
. Ask The Wombat Wizard	QU-1
. DATATRIEVE/4GL SIG Product Improvement Request Submission Form	QU-3
EDUSIG	
. Steering Committee Listing	EDU-i
GRAPHICS APPLICATIONS SIG	
. Steering Committee Listing	GRA-i
HARDWARE MICRO SIG	
. Steering Committee Listing	HMS-i
IAS SIG	
. Steering Committee Listing	IAS-i
. Contribution Guidelines	IAS-ii
. From The Editor's Terminal	IAS-1
. 10 Years Ago This Month	IAS-2
. Macro-16 to Macro-32/ A Tale of Two Systems	IAS-10
. IAS WHIMS	QU-5
LANGUAGES AND TOOLS SIG	
. Steering Committee Listing	L&T-i
LARGE SYSTEMS SIG	
. Steering Committee Listing	LS-i
MUMPS SIG	
. Steering Committee Listing	MMP-i
NETWORKS SIG	
. Steering Committee Listing	NTW-2
. DATAGRAM	QU-7

OFFICE AUTOMATION SIG

. Steering Committee Listing OA-i
. In This Issue OA-1-1
. From The Editor OA-1-2
. Fixing ALL-IN-1 Bugs OA-2-1
. 12 Bit News OA-3-1
. WPS Notes OA-3-2

PERSONAL COMPUTER SIG

. Steering Committee Listing PC-i

RSTS SIG

. Steering Committee Listing RST-i

RSX SIG

. Steering Committee Listing RSX-i
. The Editor's Corner RSX-1
. The Bag of Tricks: MACRO-11 RSX-4
. RMDEMO Enhancements RSX-6
. Rebuilding Device Drivers RSX-9
. Free Software RSX-13
. Files-11 On Disk Structure Specification RSX-14

RT SIG

. Steering Committee Listing RT-i
. Notes From The RT-11 World RT-1
. Notes From The 1986 DECUS Australia Symposium RT-4

SITE MANAGEMENT AND TRAINING SIG

. Steering Committee Listing SIT-i

UNISIG

. Steering Committee Listing UNI-i

VAX SYSTEMS SIG

. Editor's Workfile VAX-3
. INPUT/OUTPUT VAX-4
. Registration For Non-DEC VMS Products VAX-24
. Tape Management System Needs VAX-28
. VAX System SIG Committee List VAX-46
. INPUT/OUTPUT Submission Form QU-9
. System Improvement Request Submission Form QU-11
. Non-DEC VMS Product Registration Form QU-13
. VAX Tape Drive Questionnaire QU-15

LIBRARY

. New Library Programs Available LIB-1
. Revisions To Library Programs LIB-6
. DECUS Program Library Changes LIB-9

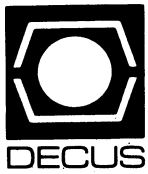
HOW TO SUBMIT TO A SPECIFIC SECTION OF THE NEWSLETTER

. Submitting Articles To The HMS SIG Newsletter HOW-i
. DECUS Subscription Service Form HOW-3
. DECUS U.S. Chapter Application For Membership HOW-5

QUESTIONNAIRE SECTION

. Ask the Wombat Wizard	QU-1
. DATATRIEVE/4GL SIG Product Improvement Request Submission Form	QU-3
. IAS WHIMS	QU-5
. DATAGRAM	QU-7
. INPUT/OUTPUT Submission Form	QU-9
. System Improvement Request Submission Form	QU-11
. Non-DEC VMS Product Registration Form	QU-13
. VAX Tape Drive Questionnaire	QU-15

**ARTIFICIAL
INTELLIGENCE**



ARTIFICIAL INTELLIGENCE SIG

Cheryl Jalbert
Chair

J C C
128 West Broadway
Granville, Ohio 43023
(614)587-0157

Terry Shannon
Newsletter Editor
Digital Review
Prudential Tower
800 Boylston Street Suite 1390
Boston, MA 02199
(617)375-4321

Pam Vavra
Symposia Representative

Hughes Aircraft EDSG
PO Box 902 E52/D220
El Segundo, CA 90245-0902
(213)616-7071

Art Beane
Counterpart

Sally Townsend
Store Representative

Jim Sims
Public Domain Software TF Chair

Bill Lennon
Reporter to the Update.Daily

David Slater

George Winkler

Jeff Fox

Don Rosenthal
Vice Chair
OPSS Working Group Chair
Space Telescope Science Institute
Homewood Campus
Baltimore, MD 21218
(301)338-4844

Becky Wise
Newsletter Task Force Chair
Administrative Assistant
Amdalh CSD
2200 North Greenville Avenue
Richardson, TX 75081
(214)699-9500 X 272

Chris Goddard
Membership Coordinator
Suite Coordinator
Simfact Associates
9210 Skypark Court
San Diego, CA 92123
(619)565-1865

Tom Viana
PSS Representative

Dennis Clark
Site Coordinator, Nashville

George Humfeld
Session Notes Editor

David Frydenlund
Assistant Session Notes Editor

John Williamson

Wayne Graves

Matt Mathews



DECUS

BUSINESS APPLICATIONS

BA

BUSINESS APPLICATIONS SIC STEERING COMMITTEE

Chairman

Stuart Lewis
Douglas Furniture
Bedford Park, IL

CL SIG Liaison

Becky Burkes
Financial Insurance Consultants
Covington, LA

Symposium Coordinator

Steve Simek
IRT Corporation
San Diego, CA

DMS SIG Liaison

Joe Sciuto
Army Research Institute
Alexandria, VA

Asst. Symposium Coordinator

Bobbie Wiley
CEI Perry Nuclear Power Plant
Euclid, OH

Members-at-Large

Robert D. Lazenby
Dixie Beer Dist, Inc.
Louisville, KY

LRP and Marketing Coordinator

Arnold I. Epstein
D-M Computer Consultants
Rolling Meadows, IL

Robert Kayne
Gallaudet College
Washington, DC

Ray Evanson
Paragon Data Systems
Winona, MN

Marketing Asst

George Dyer
Gallaudet College
Washington, DC

Digital Counterparts

Sue Yarger
Digital Equipment Corporation
Merrimack, NH

Communications Representative

Steven Lacativa
Price Waterhouse
New York City, NY

Ray Arsenault
Digital Equipment Corporation
Merrimack, NH

Newsletter Editor

Thomas Byrne
L Karp and Sons
Elk Grove Village, IL

SIC Mentor

Bill Brindley
Networks SIG Chair

Session Notes Editor

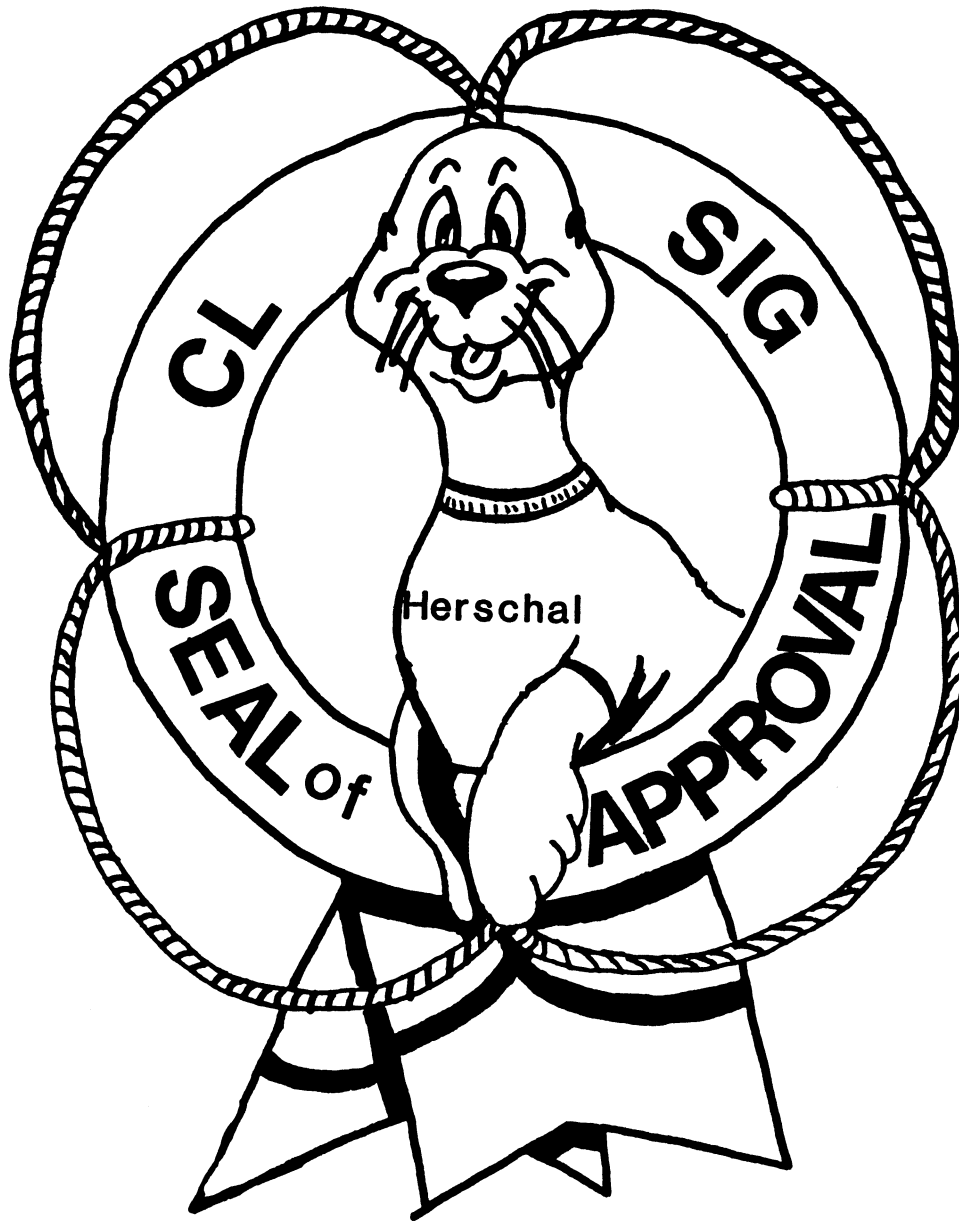
Raymond Swartz
Goodyear Tire and Rubber Co.
Akron, OH

SIC Review Committee

Larry Jasmann
Leslie Maltz
Ted Bear
Jeff Killeen

Library Representative

David Hittner
Projects Unlimited
Dayton, OH



COBOL

BASIC

DIBOL

RPG

COMMERCIAL LANGUAGES SIG

Chairperson

Dena Shelton
Systems Industries
Milpitas, CA

Symposium Coordinator

Ray Strackbein
Palm Desert, CA

Library Coordinator

Philip Hunt
System Industries
Milpitas, CA

Session Note Editor

Bob Van Keuren
Userware International, Inc.
Escondido, CA

Newsletter Editor

Ted A. Bear
Ramtek
Santa Clara, CA

Ass't Newsletter Editors

Beverly Welborne
LaPorte, IN
Kevin Cullen
VITA-Mix Corp.
Holmstead Falls, OH
Daniel Cook
Userware International, Inc.
Escondido, CA

BASIC Working Group Members

Mark Hartman
Jadtec Computer Group
Orange, CA
Rocky Hayden
UserWare International Inc.
Escondido, CA
Bill Tabor
Computer Productss
Pompano Beach, FL

Ted A. Bear
Ramtek
Santa Clara, CA

COBOL Working Group Members

Keith Batzel
Crowe, Chizek & Co.
South Bend, IN
Mary Anne Feerick
RDBS Inc.
Kernersville, NC
Bill Leroy
The Software House, Inc.
Atlanta, GA
Herbert J. Matthews IV
ManTech International Corp.
Alexandria, VA
Jim Welborne
Crowe Chizek & Co.
South Bend, IN
Jim Wilson
Pizer Inc. QC Div.
Terre Haute, IN

DIBOL Working Group Members

Neil Baldrige
CompuShare
Lubbock, TX
Becky Burkes-Ham
Financial Insurance Consultant
Covington, LA
Colin Chambers
Software Ireland Rep. Inc.
Portola Valley, CA
Mark Derrick
WAAY-TV
Huntsville, AL
Gary A.P. Kohls
Milwaukee, WI
Ken Lidster
Disc
Sacramento, CA
Kenneth M. Schilling
MCBA
Montrose, CA

Marty Schultz
Omtool Inc.
Tewksbury, MA
Marty Zergiebel
The Software Gallery
Brookfield, CT

RPG Working Group Members

Keith Batzel
Crowe Chizek & Co.
South Bend, IN

Digital Counterparts

Tom Harris
Nashua, NH
Jim Totten
Nashua, NH
Joe Mulvey
Nashua, NH
Shirley Ann Stern
Nashua, NH

Standards Representatives

BASIC
Dan Esbensen
Touch Technologies, Inc.
Escondido, CA
COBOL
Bruce Gaarder
Macalester College
St Paul, MN
DIBOL
Eli Szklanka
TEC
Newton, MA



D A A R C

DAR

DAARC

Chairman

James Deck
Inland Steel Research Lab
East Chicago, IN

Hardware & Interfacing

Peter Clout
Los Alamos National Lab
Los Alamos, NM

Symposium Coordinator

Mack Overton
FDA
Chicago, IL

Math Statistics & Analysis

Herbert J. Gould
C.C.F.A. University of Illinois Med Center
Chicago, IL

Newsletter Editor

Ellen Reilly
William H. Rorer
Ft. Washington, PA

Process Control - Industrial Automation

Bill Tippie
Kinetic Systems Corp.
Lockport, IL

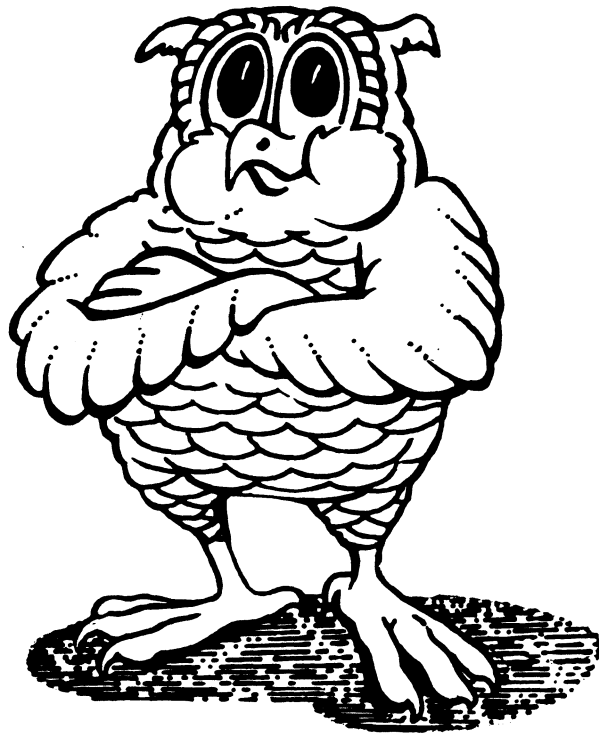
DEC Counterpart

Nancy Kilty
Digital Equipment Corporation
Marlboro, MA

RS-1

George Winkler
CPC International
Argo IL

INFO-the DMS SIG NEWSLETTER



JOIN THE WISE

Data Management Systems SIG Steering Committee
September 1, 1986

SIG Chairman:
Joseph F. Sciuto
Army Research Institute
Alexandria, VA
(202) 274-9420

Comptroller:
Alan Schultz
Land Bank National DP Center
Omaha, NE
(402) 397-5040

Symposium Coordinator:
Keith Hare
JCC

Granville, OH
(614) 587-0157

Symposium Coordinator:
Barbara Mann
TRW
Redondo Beach, CA
(213) 532-2211

Communications Committee Representative/
Newsletter Editor:
J. G. Russell Poisson
SEED Software Corporation
Alexandria, VA
(800) 428-9400

Session Note Editor:
Mark Morgan
Farm Credit Banks
Springfield, MA
(413) 732-9721

Membership Coordinator:
VACANT

MIS Working Group Coordinator (Past SIG Chairman):
Steve Pacheco
Ship Analytics
North Stonington, CT
(203) 535-3092

MIS Working Group (Past SIG Chairman):
Sandy Krueger
Key Financial Systems, Inc.
Pine Brook, NJ
(201) 299-6600

Working Group Coordinator/
Database Working Group:
Jim Perkins
PSC, Inc.
Shelburne, VT
(802) 863-8825

Forms Working Group:
Debbie Kennedy
Land Bank National DP Center
Omaha, NE
(402) 397-5040

Non-Digital Working Group:
Doug Dickey
GTE Government Systems
Rockville, MD
(301) 294-8400

RMS Working Group Coordinator:
Allen Jay Bennett
Lear Siegler Apistan
Grand Rapids, MI
(616) 451-6429

Pre-Symposium Seminar Coordinator/
Black Book:
David B. Turner
Korn/Kerry International
Los Angeles, CA
(408) 945-9600

ANSI Standards Coordinator:
Herman "Spence" Spencer
Army Research Institute
Alexandria, VA
(202) 294-9420

Member-At-Large:
Larry W. Hicks
Relational Database Services
Kernersville, NC
(919) 996-4882

Member-At-Large:
Richard Arndt
Cognos Corporation
Houston, TX
(713) 690-1105

AI SIG Liaison:
David Slater
Institute for Defense Anal
Alexandria, VA
(703) 845-2200

Datatrieve Liaison:
John Schutt
J. R. Simplot Company
Boise, ID
(208) 336-2110

DEC Counterpart:
Wendy Herman
Digital Equipment Corp
Nashua, NH
(603) 881-2494

Contributions

Contributions to the newsletter can be sent to either of the following addresses:

Editor, DMS SIG Newsletter	Russ Poisson
c/o DECUS U.S. Chapter	DMS SIG Newsletter Editor
219 Boston Post Road, BP02	SEED Software Corporation
Marlboro, MA 01752	2121 Eisenhower Avenue
	Alexandria, VA 22314

Letters and articles for publication are requested from members of the SIG. They may include helpful hints, inquiries to other users, reports on SIG business, summaries of SPR's submitted to DEC, etc. Machine readable input is highly desirable.

Submitters should keep in mind the DECUS policy on commercialism.

VAX Rdb/VMS

Guidelines and Standards Part II:

Database Design and Definition

by: Anthony Caserta, Jr.
Bankers Trust Company
One Bankers Trust Plaza
New York, N.Y. 10015

August 25, 1986

Rdb Standards and Guidelines

Table of Contents

1.	Introduction.....	3
2.	Standards.....	4
2.1.	Intermediate Fields.....	4
2.2.	/AUDIT Qualifier.....	5
2.3.	Invoking the Database.....	6
2.4.	Transactions.....	7
2.5.	Structured Code.....	8
2.6.	ON_ERROR Handling.....	8
2.7.	CALLABLE RDO.....	9
3.	Guidelines.....	10
3.1.	DML's under RDO.....	10
3.2.	RSE and Crosses.....	10
3.3.	More Lock Optimization.....	10
3.4.	Rdb and TDMS.....	10
3.5.	RDOINI: RDO Initiation File.....	10
3.6.	Batch Update.....	10
3.7.	Index Limitation.....	11
3.8.	CROSS vs Nested Loop.....	11
3.9.	Snapshot File Suspend.....	11
3.10.	Access from a Remote Node.....	12

1. Introduction

This paper is the second of three dealing with the use of VAX Rdb/VMS. It will deal with Rdb Program Design and Programming standards and guidelines. The next will deal with guidelines and standards for Database Administration.

I distinguish guidelines from standards in that guidelines are general considerations, trade-offs, or recommendations, while standards are rules, which when followed will accrue benefits of Rdb performance or application maintainability.

This paper is directed the programmer who has no DBMS experience and who is faced with the need to design and write an Rdb program.

I advocate the use of these standards for the use of Rdb as a database management tool in an integrated database environment. That is, an integrated tool as distinguished from its use as a local file access method or in a prototype mode. Use of the standards and guidelines are likely to add a NEGLIGIBLE amount of effort to the development process. But this slight incremental effort will help ease the maintenance backlog, promote data security, maintain data integrity and promote the efficient utilization of system resources.

2. Standards

2.1. Intermediate Fields

Use definitions from the CDD for ALL fields. Without using common data definitions for all fields, even intermediate (Working Storage type) fields, Rdb has no way of promoting Data Independence. This can be done by copying the definition of fields defined globally to Rdb (and which are defined in the CDD) into the appropriate data structure.

That is, say a field is defined to Rdb and since the invoke used the pathname, to the CDD to. This field definition appears to CDD as if it were a record structure. When defining an intermediate data structure to the CDD, use the copy statement to copy the definition into the definition of the intermediate structure area.

But this will only go so far in promoting data independence. Ideally, when the definition of this global field was changed, somehow the definition of the data structure which copied the definition would change. This is not the case. When the Rdb definition of this field changes, Rdb can change the definition in the CDD for the globally defined field (provided the changing transaction is performed within an INVOKE which uses pathname). But it will not change the definition of the data structure which copied the definition.

This is a case where the history list is useful. If the copy clause in the structure definition statement used the /AUDIT qualifier then a check of the history list would reveal which structures used this definition.

This leads to the next standard.

2.2. /AUDIT Qualifier

The /AUDIT qualifier allows you to insert an entry into the history list of any record definition. DML statements, most VAX programming languages, and VAX Information Architecture products accept the /AUDIT qualifier. Use the /AUDIT qualifier for:

DML statements such as BACKUP, COPY, CREATE, DELETE/PROTECTION, EXTRACT, MEMO, RESTORE, and SET PROTECTION

compiles of programs, and

with VIA products.

Note though that using the /AUDIT qualifier each time you compile a program or use a VIS product during debugging may result in an excessively long history list for the record in question. Either limit the use of the /AUDIT qualifier, or use the DML MEMO/AUDIT statement. The MEMO statement's sole purpose is the insert a history list entry.

2.3. Invoking the Database

2.3.1. By Filename

Invoke the database by filename only in application programs, unless in the rare instance when the program will change database definitions.

2.3.2. Database Handles

The database handle is used to distinguish between more than one active database in certain statements and clauses. Even if you do not expect your application to use more than one database, always use a handle for the database. This positions you for the day when you will access multiple Rdb databases in your application.

When different program modules access the same database note that it is required that all the modules that invoke the database use the same database handle and the same file name.

2.4. Transactions

2.4.1. Explicitly coded START_TRANSACTIONS

Explicit coding of a START_TRANSACTION makes the program easier to understand and helps the optimizer.

First, always issue a START_TRANSACTION statement with an explicit transaction mode (READ_ONLY, READ_WRITE). Relying on Rdb defaults is unsafe and does not promote understanding when reading the code.

Second, when issuing a START_TRANSACTION, use the reserving clause and specify a lock type (READ, WRITE) and a share mode (SHARED, PROTECTED, EXCLUSIVE).

Similar rules apply to explicitly issuing ROLLBACK, COMMIT, and FINISH statements.

2.4.2. Transaction Scope

In order to avoid holding locks across terminal I/O, release the database before output of a screen to a terminal (particularly when presenting a screen populated with data for update); that is, when retrieving data for update, do not start a transaction in read-for-update mode then output a screen WAITING for operator response, without closing (committing) the transaction.

One method is to:

```
start a transaction read-only,  
retrieve the desired data resource,  
put it to some temporary storage,  
end the transaction,
```

```
put out the screen,  
wait for operator response,  
take in operator entered data,
```

```
start a database transaction read-write,  
read the same desired data,  
check it against temporary storage,  
  if the database data is the same as the temporary  
    storage data, apply the operator's changes  
    (if any) and commit the transaction,  
  if the database data is NOT the same as the temporary  
    storage data, let the operator know that the  
    data has changed since the request was  
    issued.
```

Another method is to:

start a transaction read-only,
use the lock manager on the desired data resource,
retrieve the desired data,
end the transaction,

put out the screen,
wait for operator response,
take in operator entered data,

start a database transaction read-write,
read the same desired data,
apply the operator's changes
commit the transaction,

In either case, keep the transactions as short as possible so as to minimize the database activity and consequently optimize concurrent access to the database. That is to say, shorter transactions promote speedier recovery and minimize the amount of data which will have to be re-entered in the event of a system crash.

2.5. Structured Code

Rdb commands embedded in application programs will be well structured and appropriately indented to promote readability. As well, every line of DML code will be prefaced by "&RDB&" to promote readability.

2.6. ON_ERROR Handling

Associate an ON_ERROR statement with all Data Manipulation Language statements. This provides paybacks in both application testing and when the application is in production.

In conjunction with this, create an application-wide standard routine to handle these unanticipated errors. This routine will be included in every program which uses the database.

As well, do not ignore explicitly handling an unanticipated null record stream resulting from an RSE, since a null response is not considered an error.

2.7. CALLABLE RDO

It is undesirable to use "CALLABLE RDO" in application programs since it is highly inefficient. CALLABLE RDO can be used only when

Employing a programming language for which there is no precompiler or

Or for special purposes such as the creation of DML statements on-the-fly.

13. Guidelines

3.1. DML's under RDO

Develop and test data manipulation language statements under RDO, where appropriate, before inserting them into program code. This helps in speeding the development of the program since the DML has already been tested.

3.2. RSE and Crosses

In order to promote efficiency, when coding a Record Selection Expression, code the selection before coding the cross. This way only the selected tuples are crossed.

3.3. More Lock Optimization

In reserving relations at START TRANSACTION time, pay attention to relations which are mentioned in related constraints. Explicitly reserve these too since they would be locked anyway, and explicit locks are more efficient.

3.4. Rdb and TDMS

Rdb edit criteria enforced by either "VALID IF" or "CONSTRAINTS" are evaluated at either update or commit time. If it is necessary to edit before this time, i.e. at screen read time, then this edit criteria may need to be duplicated in TDMS. In such a case, TDMS will be used to edit for characteristics such as numeric, left justified, etc; while "VALID IF" and "CONSTRAINT" features will be used to edit for range values, referential integrity, and such.

3.5. RDOINI: RDO Initiation File

Development and Production logical name RDOINI pointing to a general initiation file so that when RDO is run, RDO looks for a standard initiation file. This file can invoke the appropriate database(s) and display a message indicating which databases are available.

3.6. Batch Update

Since "Batch Update" in V2 does not use journaling, use it to speed batch update processing. Journaling is not always desirable to ignore, but where there is a narrow window and where the updating process is stable (i.e. not likely to crash) then avoiding journal processing will speed the process.

3.7. Index Limitation

When a transaction creates an index, it cannot be used by another transaction until the creating transaction issues a commit and the second transaction issues a START TRANSACTION.

3.8. CROSS vs Nested Loop

CROSS performs better than a nested FOR loop. CROSS takes advantage of the Rdb optimizer, while a nested FOR loop gets optimized one loop at a time.

3.9. Snapshot File Suspend

For add intensive applications the application may want to suspend the use of the snapshot file in order to improve response time.

VAX Rdb/VMS Guidelines and Standards....
for Program Design and Programming

3.10. Access from a Remote Node

A process can access an Rdb/VMS database from a remote node in a network using a full file specification in the INVOKE DATABASE statement.

Assume the process is logged into a node REM4 and the Rdb/VMS database INVENTORY is located on the network node CENT in the DISK1:[NEWYORK.USGOVI] directory. The following INVOKE DATABASE statement gives the process access to the INVENTORY database.

```
RDO> INVOKE DATABASE FILENAME  
cont> 'CENT"OBRIEN EIRE":DISK1:[NEWYORK.USGOVI]INVENTORY'
```

In the above example, CENT is the remote name, OBRIEN is the account in which the database is defined, and EIRE is the password for that account.

If the invoke statement does not include the password in the file specification, the process logs in to the CENT node as DECNET with DECNET's UIC and the access rights assigned to it in the access control list. No disk specification is required; DECNET runs the login process for the user name OBRIEN and uses Obrien's disk automatically.

To improve performance over the network, modify the login command file on the remote node to allow faster processing. For example, if logical names need to be defined for the database, do so at the beginning of LOGIN.COM. Then include a DCL command such as:

```
$ IF 'F$MODE()' .EQS. "NETWORK" THEN $EXIT
```

There is no need for the job to know the other DCL symbols and unrelated logical names.

The Wombat

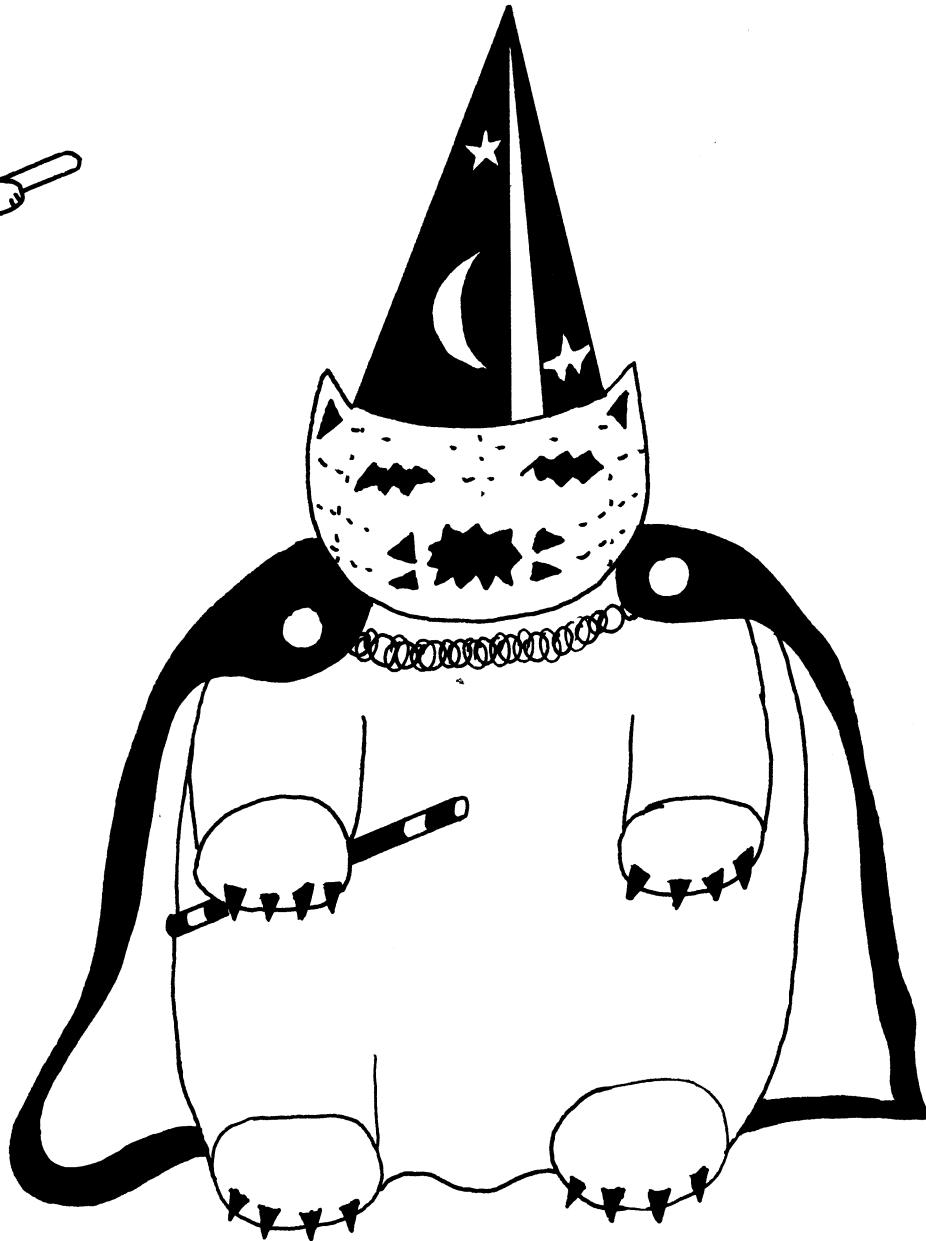
EXAMINER

and 4GL
Dispatch

"Increases the Circulation of Anyone in America"

Volume 8

Number 3



DTR

DTR/4GL Special Interest Group - Officers**DTR/4GL Masters List****SIG Chairman**

Joe H. Gallagher
 Research Medical Center
 2316 East Meyer Blvd
 Kansas City, MO 64132
 816-276-4235

Past SIG Chairman - Vol. Coord.

Larry Jasmann
 U. S. Coast Guard
 10067 Marshall Pond Rd.
 Burke, VA 22015
 202-426-2344

Symposia Comm. Rep.

Chris Wool
 E.I. duPont Eng. Dept.
 Louviers Bldg.
 Wilmington, DE 19898
 302-366-4610

Asst. Volunt. Coord.

Susan Krentz
 NKF Engineering.
 12200 Sunrise Valey Dr.
 Reston, VA 22091
 703-620-0900

Pre-Symposia Seminars

Dana Schwartz

 15719 Millbrook Lane
 Laurel, MD 20707
 301-859-6277

Campground

Bert Roseberry
 Commandant (G-APA-1)
 2100 2nd Street, S.W.
 Washington, D.C.
 202-267-2629

DEC Counterpart - Datatrieve

Andy Schneider
 Digital Equipment Corp.
 110 Spitbrook ZK02-2/N59
 Naushua, NH 03062

Editor & Comm. Rep.

Donald E. Stern, Jr.
 Warner Lambert Company
 10 Webster Road
 Milford, CT. 06460
 203-783-0238

Production Editor

Steve Cordiviola
 Kentucky Geological Survey
 311 Breckinridge Hall
 Lexington, KY 40506
 606-257-5863

Symposia Asst. & New Prod. Coord.

Diane Pinney
 Computer Sciences Corp.
 443 Inyokern Road
 Ridgecrest, CA 93555
 619-446-6585x284

Library Comm. Rep & Artist

Bart Z. Lederman
 I.T.T. World Communications
 67 Broad St. (28th Floor)
 New York, NY 10004
 212-607-2657

Session Notes

Wanda Anderson
 SRI International MS:PN341
 333 Ravenswood Avenue
 Menlo Park, CA 94025
 415-859-2577

WW Editor & PIR Coordinator

Philip A. Naecker
 Consulting Engineer
 3011 N. Mount Curve Ave
 Altadena, CA 91001
 818-791-0945

DEC Counterpart - RALLY, TEAMDATA

Basil Harris Jr.
 Digital Equipment Corp.
 110 Spitbrook ZK02-2K29
 Nashua, NH 03062-2698

VAX-11 Datatrieve -

Joe H. Gallagher
 4GL Solutions
 10308 Metcalf Suite 109
 Overland Park, KS 66212
 913-894-9550

Bart Z. Lederman
 I.T.T. World Communications
 67 Broad Street (28th Floor)
 New York, NY 10004
 212-607-2657

Joe Kelly
 WymAn Gordon Co.
 Worchester Road
 North Grafton, MA 01536
 617-839-4411x5480

Larry Jasmann
 U. S. Coast Guard
 10067 Marshall Pond Rd.
 Burke, VA 20015
 202-426-2344

James Swanger
 G. D. Searle & Co.
 4901 Searle Parkway
 Skokie, IL 60077
 312-982-7430

Pro-Datatrieve -

Joe H. Gallagher
 4GL Solutions
 10308 Metcalf Suite 109
 Overland Park, KS 66212
 913-894-9550

Datatrieve-11 (RSX) -

Bart Z. Lederman
 I.T.T. World Communications
 67 Broad Street (28th Floor)
 New York, NY 10004
 212-607-2657

Chris Wool
 E. I. duPont, Engineering Dept.
 Louviers Building
 Wilmington, DE 19898
 302-366-4610

Dick Azzi
 Motorola SPS
 Box 2953 MS: P116
 Pheonix, AZ 85062
 602-244-4316

Chris Heinz
 American Board of Family Practice
 2228 Younge Drive
 Lexington, KY 40505
 606-269-5626

Bert Roseberry
 Commandant (G-APA-1)
 2100 Second Street, S.W.
 Washington, D.C.
 202-267-2629

Donald E. Stern, Jr.
 Warner Lambert Company
 10 Webster Road
 Milford, CT 06460
 203-783-0238

Bart Z. Lederman
 I.T.T. World Communications
 67 Broad Street (28th Floor)
 New York, NY 10004
 212-607-2657

Contributions

Submissions to this newsletter are constantly sought. A submission can be an article, a letter to the Wombat Wizard, a technical tip, or anything of interest to people using or considering the use of Datatrieve or any 4GL product. Submissions on magnetic media are preferred but almost any type will be considered.

Contributions for the newsletter can be sent to either of the following addresses:

Editor, DATATRIEVE Newsletter
c/o DECUS U.S. Chapter
219 Boston Post Road, BP02
Marlboro, MA 01752

Donald E. Stern, Jr
Warner Lambert Company
10 Webster Road
Milford, CT 06460

Table of Contents

DECUS U.S. Chapter SIG Newsletters, Volume 2, No. 3, Nov. 1986
Wombat Examiner and 4GL Dispatch, Volume 8, No. 3

DTR-2	-	Chairman's Corner
DTR-3	-	Integrating Datatrieve with DECalc and DECGraph
DTR-15	-	Commonly Asked Datatrieve Questions and Answers
DTR-23	-	Preparing Datatrieve Plots for Printing

Chairman's Corner

Joe H. Gallagher, Ph. D., 4GL Solutions, Overland Park, KS

It is with a great deal of pride and pleasure that I include my congratulations with the praise and congratulations of the whole DECUS leadership to the staff of The Wombat Examiner and 4GL Dispatch. On October 7, 1986, at President's Night during the San Francisco DECUS Symposia, a special award was given to our newsletter staff in recognition of their outstanding efforts toward the creation and productions of the newsletter. Don Stern, managing editor; Steve Cordiola, production editor; Bart Lederman, artist; and Phil Naecker, Wombat Wizard column editor are most deserving of our praise and thanks.

Unless you have actually produced a newsletter under the difficult circumstances that exist (no guaranteed source of material, relentless and consuming publications deadlines, little or no direct support, and an absorptions of their physical and emotional energies which rightly belong to their families or employers), you can not appreciate the monumental effort that is expended each month to create a quality newsletter.

Our current newsletter staff are richly deserving of our praise. But they follow in a tradition of quality newsletter production for the DATATRIEVE SIG. Kathy Tamer was the first newsletter editor when the SIG was formed with Chuck Watson as SIG Chair. Five newsletters were published between April '79 and February '81. Each of the newsletters (except the first unnumbered issue) contained a picture or art work on the cover. And each had the now familiar "The Wombat Examiner" banner with the slogan "Increases the Circulation of Anyone in America." With Volume 4, Issue 1, Cynthia Curry and Virginia Sventek began editing the newsletter. During their editorship, the quality of the newsletter was dramatically increased. Their second issue (Volume 4, Issue 2, June '82) was completely typeset (with a typesetting machine); this may have been the first of any of the DECUS newsletter which was typeset. That issue also contained our first PIR (product improvement request with Bart Lederman as the coordinator) and the first DATATRIEVE Masters List (which contained the names of Larry Jasmann, who would soon take over the SIG Chair from Chuck Watson). Volume 4, Issue 3 in November '82 contained the first cover done by Bart Lederman. With Volume 5, Issue 1 in March '83, Cynthia Curry was now Cynthia Curry Mealey and became the Publications Coordinator (what now corresponds to the Communication Committee Representative). Virginia Sventek continued on as editor for three more issues until January '84. Those issues, Number 1, 2, and 3 of Volume 5, were completely or almost completely typeset and were monuments to the editor's hard work and the amount of quality information that appeared within the pages of the newsletter. On the cover of the first of Virginia's newsletters appeared Bart's artistic rendition of Wombat Holmes and Watson (perhaps Bart's best work that we have seen so far) and in the last of Virginia's newsletters appeared Dan Dietterich's THE_THREE_LITTLE_WOMBATS (perhaps the cleverest newsletter article that has every appeared in any DECUS publication).

In the spring of '84, I took over from Virginia and produced four issues (one of which was a double issue) between the fall of '84 and summer of '85. During the spring and summer of '85, it was decided to combine all DECUS newsletters into a single monthly publication. September of '85 was the first issue of the DECUS U. S. Chapter SIGs Newsletters (Volume 1, Number 1 for the combined newsletter and Volume 7, Number 1 for the Wombat Examiner). It was obvious to anyone who had been a newsletter editor that a monthly newsletter could not be done by a single volunteer. With that issue Don Stern and Steve Cordivola agreed to be Associate Editors of the Wombat Examiner. Steve became Production Editor almost immediately (with the November issue which was actually done in September) and Don took over as Managing Editor (in December '85 for the February issue when I became SIG Chair Elect). Phil Naecker's prolific contributions to the PIR process and the Wombat Wizard column began in June '86. And the newsletter changed its name in March '86 to "The Wombat Examiner and 4GL Dispatch" to reflect the change in the SIG to DATATRIEVE/Fourth Generation Languages SIG.

While all newsletter editors of the DECUS SIGs are doing tremendous service to the Society, the accomplishments of Don, Steve, Bart, and Phil are particularly noteworthy. Since the beginning of the combined newsletter more issues have been published than were published before (14 before and 15 in the combined). The DTR or DTR/4GL newsletter has been published every month since the newsletters were combined. The Wombat Examiner and 4GL Dispatch and the Pageswapper are the only newsletters which have not missed a month. Original art work by Bart Lederman which is unique and has never been repeated has appeared on every of the newsletters since November '82. Bart has contributed more original articles to the DECUS newsletters (ours, the RSX Multi-Tasker, and others) than anyone else in the SIG (and perhaps any one in the whole Society). Don, Steve, and Phil have worked hard to keep the quality of the newsletter high and the content interesting and stimulating. The number of pages published in our newsletter is actually increasing; the number of original articles remains high; and the technical publication quality is the leader of the Society.

The efforts of Don, Steve, Bart, and Phil deserve our recognition. Thanks for a difficult job, very well done!

Integrating Datatrieve with DECalc and DECgraph

Donald E. Stern, Jr. - Warner Lambert Co., Milford, CT

This is based upon a session first given at the Spring 1986 DECUS U.S. Chapter Symposium held in Dallas, Texas. Issues relating to the integration of Datatrieve with Digital's spreadsheet and graphics products are discussed. The examples cited were generated using VAX/VMS V4.4, VAX-11 Datatrieve V3.3, DECalc V2.0, and DECgraph V1.5. (NOTE: V2.0 of DECalc will NOT properly link with V3.3 of Datatrieve. This is a known bug. DECalc was linked with V3.2 of Datatrieve for the examples cited.)

1.0 DECalc

DECalc is Digital Equipment Corporation's offering of an electronic spreadsheet product. It runs under the VAX/VMS operating system and is optionally linked with VAX-11 Datatrieve in order to take advantage of DTR's exceptional capability to access data. Using the DTR interface, data can be extracted from RMS files and/or Rdb or DBMS databases for use within DECalc. This enables DECalc to share data with other applications. A user can then take advantage of the various capabilities of an electronic spreadsheet such as the ability to arrange data in either columns or rows and to specify complex formulae. In addition, there are numerous built-in functions of DECalc many of which are not available in Datatrieve. (Note: Datatrieve, however, can be extended beyond its "as delivered" capabilities by the creation of User Defined Functions (UDF's).) Examples of the DECalc functions include such financial functions as present value and future value, to name a couple. The product DECalc-PLUS expands the available built-in functions to include many complex statistical functions such as analysis of variance, Chi-square, multiple regression, and many others.

1.1 Transferring Data to DECalc

Two primary ways to transfer data from a Datatrieve domain to a DECalc spreadsheet will be discussed. One takes advantage of callable Datatrieve linked with the DECalc image. The second method discussed involves the creation of a DECalc load file using a Datatrieve procedure.

In order to demonstrate the ability to transfer data from Datatrieve to DECalc, a Datatrieve domain and record were defined and a corresponding RMS file was created and loaded with sample data. For demonstration purposes a very simple database was defined. The database consists of two domains, DEMO and ACCOUNTS. The domain DEMO contains data relating to spending against a given account number during a given year. The domain ACCOUNTS associates an account number with an account description. The domain and record definitions are given below.

```
DEFINE DOMAIN DEMO USING DEMO_REC ON DEMO.DAT;
```

```
DEFINE RECORD DEMO_REC USING
01 DEMO.
   03 ACCOUNT PIC 9(4).
   03 AMOUNT_SPENT USAGE REAL
       EDIT_STRING IS $(3)9.99.
   03 YEAR PIC 9(4).
```

```
;
```

```
DEFINE DOMAIN ACCOUNTS USING ACCOUNTS_REC ON ACCOUNTS.DAT;
```

```
DEFINE RECORD ACCOUNTS_REC USING
01 ACCOUNTS.
   03 ACCOUNT PIC 9(4).
   03 ACCOUNT_NAME PIC X(15).
```

```
;
```

There is a one-to-many relationship between the ACCOUNTS domain and the DEMO domain. A domain table was defined to provide the translation of the ACCOUNT field.

```
DEFINE TABLE ACCOUNT_DESCRIPTION_TABLE FROM ACCOUNTS
ACCOUNT : ACCOUNT_NAME
END_TABLE;
```

The specific data used for this demonstration is listed as follows:

```
DTR> READY DEMO
DTR> PRINT DEMO
```

ACCOUNT	AMOUNT SPENT	YEAR
1234	\$5500.00	1985
1234	\$3400.00	1986
1234	\$3808.00	1987
1300	\$1400.00	1987
1357	\$6666.00	1985
1357	\$6719.00	1986
1357	\$7525.28	1987
1479	\$2095.00	1985
1479	\$2596.00	1986
1479	\$2907.52	1987
1775	\$2356.00	1986
1775	\$2198.00	1985
1775	\$2638.72	1987
2437	\$1956.00	1985
2437	\$1900.00	1986
2437	\$2128.00	1987
3434	\$4456.00	1985
3434	\$9000.00	1986
3434	*****	1987
4233	\$5890.00	1985
4233	\$5700.00	1986
4233	\$6384.00	1987
5101	\$2000.00	1985
5101	\$3698.00	1986
5101	\$4141.76	1987
9876	\$1234.00	1985
9876	\$3000.00	1986
9876	\$3360.00	1987
9999	\$4213.00	1985
9999	\$5579.00	1986
9999	\$6248.48	1987

1.1.1 Interactively Transferring Data from Datatrieve

One can access Datatrieve from DECalc by specifying the External option followed by the Database option (X D). At this point, DECalc looks for the name of a file which contains the database definition. The default file name is DECALC.DTR. One can now specify either the Retrieve or the Specify option. The Retrieve option implies that a database definition file had previously been created using the Specify option. Instead of specifying Retrieve or Specify, it is possible to specify Interactive. Selecting the Interactive option, one is placed at the DTR> prompt where one can define and/or manipulate data. One returns to DECalc by either typing EXIT, 'Z', or DECalc at the DTR> prompt. (One notes that DECALC has been programmed as a User Defined Keyword (UDK) using callable Datatrieve.) Upon re-entering DECalc one is placed in Specify mode.

The Specify option implies that a database definition is to be created or that an existing definition is to be modified. A database definition consists of a CDD path name, a Datatrieve domain name, a record selection expression (RSE), and/or a list of field names corresponding to fields in the domain. The definition also contains information regarding the starting box, in which to load data into the spreadsheet, whether or not to display headings, and whether the data is to be displayed in row-major or column-major order. Upon completing the data definition phase, DECalc attempts to access the Datatrieve data and load it into the spreadsheet as indicated.

The following database definition file was used to retrieve spending data against accounts during the year 1986.

```
Dictionary:      cdd$default.decus
Domain:         demo
RSE:           demo with year=1986
```

```
Field [optional]:account,amount_spent
Row,Column [R/C]:C
Headings [Y/N]: Y
Starting Box:   C3
```

This specific database definition, when executed, will use callable Datatrieve to a.) set the set the dictionary to CDD\$DEFAULT.DECUS, b.) READY DEMO, c.) FIND DEMO WITH YEAR=1986, and d.) load the ACCOUNT and AMOUNT SPENT fields of the current collection into the spreadsheet in column-major order starting in location C3 using the field name as headings. One can repeat this sequence many times, using different selection criteria, to transfer data from Datatrieve to the spreadsheet. It is up to the user, however, to maintain the necessary spatial relationships between data retrieved via more than one data definition file. For example, if one were to specify the data definition below after the one given above, data for spending data 1987 would be loaded next to the 1986 data. Synchronization between the data, however, would be lost since the account number 1300 exists for 1987 and not for 1986; the 1986 and 1987 data would be offset by one row.

```

Dictionary:   cdd$default.decus
Domain:      demo
RSE:         demo with year=1987

```

```

Field [optional]: amount_spent
Row,Column [R/C]: C
Headings [Y/N]: Y
Starting Box:   E3

```

Once the required data has been loaded into the spreadsheet, data can be moved, functions defined, calculated fields defined, and additional data manually entered using the standard spreadsheet commands.

When creating the data definition, be sure that the domain specified is case identical to that named in the RSE. The DECalc program apparently performs a check for equality between the two items and does not convert to uppercase before making the comparison. In addition, when specifying fields to be transferred the specification should be

```

FIELD1.FIELD2
      not
FIELD1.<SPACE>FIELD2

```

or DECalc will not parse the expression correctly. Furthermore, if any of the fields contain null or space data, DECalc will stop with a fatal error. These limitations are DECalc specific and can be avoided by the use of the technique described in the following section.

1.1.2 Transferring Data from Datatrieve Under Program Control

DECalc permits one to execute a pre-defined set of commands to manipulate the spreadsheet. These commands are stored in an ordinary RMS Sequential file which can be created by your favorite editor or by Datatrieve. It is the second option which will be used here to transfer data from Datatrieve to a DECalc spreadsheet.

A dictionary table was created in the CDD\$DEFAULT.DECUS subdictionary to provide a numerical translation for DECalc columns.

```

DEFINE TABLE DECALC_COLUMNS
1:A
2:B
3:C
4:D
.
.
.
20:T
.
.
.
END_TABLE

```

The following procedure was created in order to demonstrate how Datatrieve can create a DECalc command file with which data can be loaded in a known controlled fashion.

```

DEFINE PROCEDURE DECALC_TEST1
!
!Declare some variables
!
DECLARE RN USAGE INTEGER VALID IF RN BT 1 AND 1000.
DECLARE CN USAGE INTEGER VALID IF CN IN DECALC_COLUMNS.
DECLARE COL COMPUTED BY CN VIA DECALC_COLUMNS.
DECLARE ACCOUNT_NAME_BOX COMPUTED BY "||COL||3.
DECLARE ACCOUNT_NUMBER_BOX COMPUTED BY "||COL||4.
DECLARE YEAR_BOX COMPUTED BY "||"||"A"||RN.
DECLARE BOX COMPUTED BY "||COL||RN.
!
!Data starts in box B6
CN = 2
RN = 6
!
READY DEMO
!
!All output goes to an RMS file
!
ON DECALC_TEST1.DTR BEGIN
!
PRINT "\GFV$"           !Values represent money
PRINT "\GFLC"           !Center Labels
PRINT "\GC15"           !Columns 15 wide
PRINT "|A3 Year"
!
!Get a sorted list of unique account numbers
FOR A IN DEMO REDUCED TO ACCOUNT SORTED BY ACCOUNT BEGIN
PRINT ACCOUNT_NAME_BOX||ACCOUNT VIA ! Print the
! account name
ACCOUNT_DESCRIPTION_TABLE
PRINT ACCOUNT_NUMBER_BOX||'|"'||ACCOUNT! and number
! labels
!
!Get a sorted list of unique years
FOR B IN DEMO REDUCED TO YEAR SORTED BY YEAR BEGIN
!
PRINT YEAR_BOX||'|"'||YEAR !Print a year label
PRINT BOX (-) !Data for this year begins
! here
!
! If there is no data for this account and year use 0
! else use real data.
IF COUNT OF DEMO WITH ACCOUNT=A.ACCOUNT AND
YEAR=B.YEAR = 0 THEN PRINT 0 ELSE

```

```

                FOR DEMO WITH ACCOUNT=A.ACCOUNT AND
                YEAR=B.YEAR
                PRINT AMOUNT_SPENT (-) USING Z(6)9.99
!
                RN=RN+1          !Next year"s data goes in row RN
                END
                RN = 6          !First row
                CN = CN + 1
END
PRINT "|A1"                    !Return to top left corner
END
FINISH;
EXIT;
END_PROCEDURE;

```

This procedure can be executed in a number of ways. One can run Datatrieve (\$MC DTR32) and execute the procedure or one can use the Interactive option from within DECalc and execute it. When the procedure is executed the following is loaded into the file named DECALC_TEST1.DTR

```

\GFV$
\GFLC
\GC15
|A3 Year
|B3 Maintenance
|B4 "1234
|A6 "1985
|B6
5500.00
|A7 "1986
|B7
3400.00
|A8 "1987
|B8
3808.00
|C3 Legal Fees
|C4 "1300
|A6 "1985
|C6
0
|A7 "1986
|C7
0
.
.
|L3 Mortgage
|L4 "9999
|A6 "1985

```

```

|L6
4213.00
|A7 "1986
|L7
5579.00
|A8 "1987
|L8
6248.48
|A1

```

In order to instruct DECalc to use this file, one must select the \S C option and supply the file specification DECALC_TEST1.DTR to the prompt. At this point, DECalc will execute the command and load the data as indicated below.

Year	Maintenance 1234	Legal Fees 1300	. . .	Mortgage 9999
1985	\$5,500.00		. . .	\$4,213.00
1986	\$3,400.00		. . .	\$5,579.00
1987	\$3,808.00	\$1,400.00	. . .	\$6,248.48

1.2 DECalc Summary

It has been shown that it is relatively easy to load a DECalc spreadsheet with data accessed using Datatrieve. While the examples given used RMS data, it would be as easy to load data stored in a DBMS or Rdb database. DECalc users who are familiar with Datatrieve will be able to access the data interactively. For those users who do not know how to access data with Datatrieve, a Datatrieve procedure can be created which will create a DECalc load file for these users. In this manner, even complex data transfers can be effected with no special knowledge.

2.0 DECgraph

Graphical representation of data permits the compression of large amounts of data into a form from which trends and relationships are easily spotted. Recognizing this fact, Digital has included some very useful, but limited, graphics capability in the Datatrieve software. While extremely useful, one does not have to work with existing Datatrieve plots very long before wanting to extend the capabilities beyond what comes delivered with Datatrieve. There have been several articles appearing in this newsletter dealing with ways in which to modify or add to the Datatrieve plotting library. (See, for example, Joe Gallagher's article in the September 1985 issue.) The purpose of this discussion is not to add to the library of unsupported "user-written" plots but to explore the use of Digital's DECgraph product in conjunction with Datatrieve.

DECgraph is Digital's business graphics package for use under VAX/VMS. It provides the user with a great deal more flexibility to produce graphic output in the form desired. Invoked by the DCL command

\$ GRAPH

DECgraph is a menu/icon driven package which permits users to define data for graphs, the form of the graph (scattergraph, stacked bar, pie chart, multi-line, etc.), the colors used (or cross-hatch patterns for printing), axis and field labels, and title information. Like DECcalc, it can be optionally linked with Datatrieve at installation time.

2.1 Interactively Transferring Data to DECgraph

The interactive transfer data from Datatrieve to DECgraph, is a five step process. First, one must start the DECgraph program by either the command GRAPH (as shown above) or the following command.

\$ GRAPH/NOLOGO

This command is more efficient since it eliminates the need to graphically paint a VAX-11 DECgraph message which appears using the first command. In fact, many users define a symbol in their LOGIN.COM files which equates the symbol GRAPH to GRAPH/NOLOGO. When the program initialization completes five options are made available to the user: EXIT, DATA, DESIGN, DISPLAY, and HELP.

The second step is to select the DATA icon. At this point, DECgraph provides an additional four options to the user. The options are: EXIT, KEYBOARD, LOAD, and DTR. The EXIT option backs up to the previous level. Selecting the KEYBOARD option permits the user to manually enter data, the LOAD option loads data from a special file and will be discussed later, and the DTR option informs DECgraph that data from Datatrieve will be accessed.

Third, one selects the DTR icon and is placed at the DTR> prompt. The user's task is to form a current collection containing data to be plotted. Upon typing EXIT, the user is returned to DECgraph and a screen very similar to the one displayed if the KEYBOARD option had been selected.

The fourth step fill in the title, subtitle, and axis label information. The ability to specify title information represents a significant advantage over the plots supplied with V3.3 of Datatrieve.

Fifth, one indicates which fields are to be used from the Datatrieve current collection and names the fields. One cannot use Datatrieve query name when naming the fields to DECgraph. At this point the appropriate data are loaded into DECgraph. One can select the EXIT icon at this point or manually change or add to the data to be plotted. When complete the EXIT icon is selected and DECgraph prompts the user for a filename to be used to store the data. Two files will share this filename. One will have the extension .GRD and will contain the data to be plotted. The other will have the extension .GRI and will contain information relating to the title, labels, etc.

At this point, one can proceed to the graph design phase and create the final form of the graph to be displayed or printed.

Figures 1 and 2 help to illustrate some of the plotting enhancements which DECgraph provides over Datatrieve. Figure one was created using the following Datatrieve command.

DTR> Plot value_pie account.amount_spent of demo with year=1986

AMOUNT_SPENT BY ACCOUNT

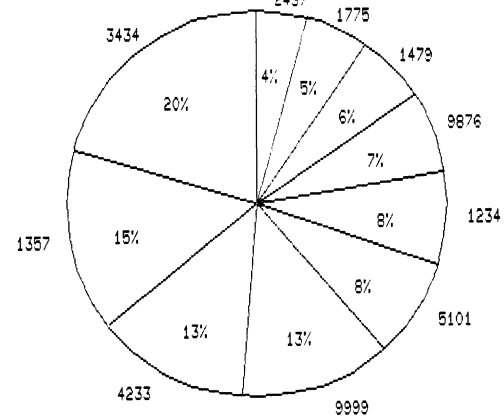


Figure 1 - Pie Chart Created Using Datatrieve Graphics

Spending Analysis by Account Number for 1986

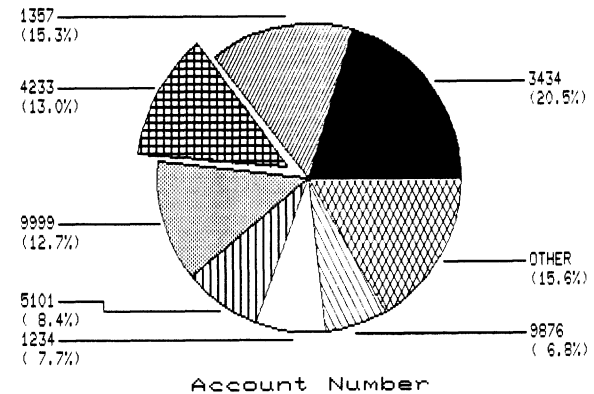


Figure 2 - Pie Chart Created Using DECgraph

Figure two was created using the same data accessed with Datatrieve but plotted using DECgraph.

2.2 Transferring Data to DECgraph Under Program Control

DECgraph recognizes a special file type which has the default extension .GRL. This special load file contains data normally stored in .GRD and .GRI files. A load file is a sequential RMS file and can be created using a text editor or with Datatrieve. If one uses Datatrieve to create the load file then complex data selection criteria can be applied.

The following Datatrieve procedure was created in order to demonstrate one method for creating a DECgraph load file.

```

DEFINE PROCEDURE LOAD_FILE
READY DEMO
DECLARE SPENT USAGE REAL.
ON *. "Device or Filename" BEGIN
  FOR Y IN DEMO REDUCED TO YEARS BEGIN
    FOR A IN DEMO REDUCED TO ACCOUNT_NUMBER BEGIN
      PRINT "TITLE Using Datatrieve with DECgraph Demo"
      PRINT "SUBTITLE Data for All Years "
      PRINT "HORIZONTAL_LABEL Account Number"
      PRINT "VERTICAL_LABEL Amount Spent"
      PRINT "X_DATA_TYPE TEXT"
      PRINT "Y_UNITS ONES"
      SPENT = TOTAL AMOUNT_SPENT OF DEMO WITH YEAR=Y.YEAR AND
        ACCOUNT_NUMBER=A.ACCOUNT_NUMBER
      IF SPENT NE 0 THEN
        PRINT " " | ACCOUNT_NUMBER | " " |
          " " | FORMAT(AMOUNT_SPENT) USING Z(4)9.99
      END
    END
  END
END
END
END
END_PROCEDURE

```

Figure 3 illustrates a graph which is created using this load file.

The load file can be used interactively or non-interactively. To use it interactively, one would choose the "LOAD" icon from the data entry menu. DECgraph would then read in the load file. At this point, one could add to, subtract from, or modify the loaded data. DECgraph then creates GRD and GRI files to be used in graph creation.

To use the load file in a non-interactive mode the following command can be used at the DCL prompt.

```
$ GRAPH/NOINT/LOAD load_file_name [description_file_name]
```

The graph description file name, optionally specified, provides DECgraph with information regarding the type of graph (PIE, BAR, LINE, etc.), the colors to be used, etc. If no file is specified, a file called GRAPH\$LIBRARY:SAMPLE.GRG is used; this file is a multi-bar graph. Upon issuing a command to graph with a load file (as above), DECgraph will create a graph data file (GRD) and a graph information file (GRI) on your behalf.

Using Datatrieve with DECgraph Demo
Data for All Years

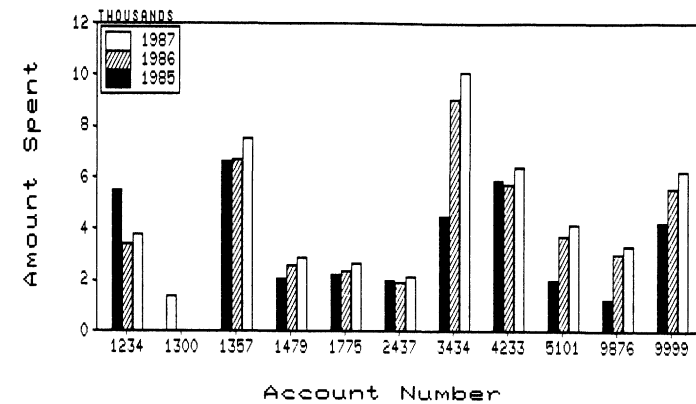


Figure 3 - Graph Created Using a DECgraph Load File

Additional qualifiers on the GRAPH command include the /SIXEL qualifier which instructs DECgraph to produce a SIXEL graphics file which can be printed on an LN03 printer, for example.

2.3 DECgraph Summary

For users who know how to use Datatrieve, data can be retrieved using the interactive DTR mode from DECgraph. For those users who are not familiar with Datatrieve or if complex data retrievals are required, a Datatrieve procedure can be programmed which will produce a DECgraph load file containing the data to be plotted.

DECgraph provides the user with significantly more control over the final appearance of graphs than Datatrieve. In addition, DECgraph can produce files which can be printed directly on system printers as opposed to local attached printers. Data stored in RMS files, RDB, or DBMS can be retrieved and plotted using the Datatrieve interface to DECgraph.

Commonly Asked DATATRIEVE Questions and Answers

(From the Spring 1986 DECUS U.S. Chapter Symposium)
Session Chair: Larry Jasmann, U.S. Coast Guard, Burke VA
Transcribed by: B. Z. Lederman, Brooklyn, NY

Panelists:

Joe H. Gallagher, Research Medical Center, Kansas City, MO
Andy Schneider, DEC Developer, VAX-DATATRIEVE
Dick Azzi, Motorola, Phoenix, AZ
Chris Wool, E.I. DuPont, Wilmington, DE
B. Z. Lederman, Brooklyn, N.Y.

This is a transcription of a panel presentation which answers some of the most common questions asked about DATATRIEVE. Some of the material has been reordered when that would group logical subjects together. The transcription may paraphrase some questions or answers for clarity, and the transcriber apologizes in advance for any misspelled names. Throughout this paper DTR is an abbreviation for DATATRIEVE.

Why is DATATRIEVE so slow?

(Larry:) DTR has a lot of power, and does a lot of things, but it also "sits on top of" one of three other products: RMS, DBMS, or Rdb. If you use DTR in such a way as to cause, for example, RMS to do a sequential search of a file containing 20,000 records, you should not be surprised if it takes a long time to respond with an answer. If you are a programmer [in a traditional language] you probably wouldn't do such a silly thing when writing a program, but when you are using DTR interactively on a large file it's really easy to do this. When you get beyond 1000 records, the amount of time to access a file sequentially skyrockets compared with keyed retrieval. (Joe:) The point to be made is that DTR is only as fast as what it sits on. It's because DTR hides some of the details of what is going on below it that many times it's possible to do something that seems perfectly reasonable to you, but is very slow because the file design is not appropriate for that function, and performance suffers considerably. (Larry:) A corollary: it's not altogether clear, unless you've studied it, which Datatrieve constructs will cause sequential searches, and this is something you need to know well if you have a big file. (Dick:) When I have an application which seems slow, my first answer to this normally is to try hitting the NO SCROLL key again. That happens quite often and that will slow your application. Seriously though, along with the sequential portion, the number of keys [in an indexed file] has a direct bearing on how fast the application will be: this doesn't matter much on a read, but on a write, the more keys there are the slower it will be. (Joe:) if you are going to retrieve records on both keys, the time needed to store the extra key will be well worth it in the retrieval. However, if you are not going to retrieve records on that key, you are going to pay an overhead price. It's important to choose the keys carefully, to see only those which will be used for retrieval. These underlying factors in file design determine how fast the application will be in

Datatrieve. (Andy:) One of the advantages to using keys is that RMS will do sorting for you. When you create a primary key and DTR says "give me the records" RMS gives them back in sorted order. If you have an application and you have a primary key, and you enter a command which sorts on that key, why would it take so long? DTR isn't super smart: if you explicitly order DTR to sort the data, it isn't able to tell that it's already sorted. So: don't go out and automatically sort everything, figure out what are primary keys, and don't sort fields that are already sorted. How do you know when your retrieval is using keys? One way is to do this: instead of just running the DTR image, run it with DEBUG.

RUN/DEBUG SYS\$SYSTEM:DTR32

You get some VAX DEBUG headers and messages, and then the prompt: simply say "GO", and you will be in DTR. What you are doing is initializing the DEBUGGER, and then you will be in DATATRIEVE. What happens is that when you perform an RSE, if a key is being used you will receive an informational message on your terminal for every key being used in your RSE or Boolean or whatever. If you were assuming that three keys are being used, this way DTR will tell you if those keys are actually being used or not. If it isn't using it, then perhaps there is a flaw in your design, and you can go back and work on it. This is a debugging technique to see if what you are doing is what you thought you were doing.

(Larry:) Another thing you need to know is FIND and FOR statements. If you do a FIND and create a current collection, subsequent operations on that collection are going to be done sequentially. This is usually the minimum amount of time you will save with a FOR, but there are other savings that are also obtained. You should remember that any operation on the collection is not keyed, even if it looks as if it was. The only time a FIND is better than a FOR is if you have a very large domain, and you can do a FIND to collect a relatively small number of records, and are going to do several operations that small collection. For example, if you have 10,000 records and want to work on a subset of 50 or 60 records, it makes sense to use a FIND, otherwise not. (Dick:) While we are talking about FINDs, it's important to remember that when you do a SORT, even to do a PRINT (for example, FIRST 5 — SORTED BY field), that DTR is going to do the SORT first. If you can do a FIND and reduce the number of records you are going to be using, and then SORT that small number, you will save a lot of time because DTR will not have to sort the whole file.

(Andy:) another bottleneck is access to the dictionary (the CDD). It's crucial, especially at initial access time, that the dictionary not be "top heavy". A lot of people make the mistake of putting everything into CDD\$TOP, and then when you want to ready a domain the amount of time that it takes for CDD to access the pieces in the dictionary is extremely high. If you lot of stuff in CDD\$TOP and not much in subdirectories, create a good tree structure and move stuff down the tree.

(Joe:) There is one area I run into that most users don't run into. From a scientific and medical standpoint, we have some users who do calculations in DTR rather than some other language like FORTRAN, so in fact they are doing a lot of calculations in DTR. In many cases they created complex procedures where all of the temporary variables are declared something like PIC 999V999 (string variables). If for some reason you have

to do heavy calculations in DTR you gain a substantive return by converting those variables to COMP variables (REAL, INTEGER) because DTR does a fair amount of conversion, and these are CPU intensive activities where numbers in one format has to be converted to other formats with a lot of checks. If you are going to do a lot of calculations (such as a data base of scientific data) you get a performance improvement by making the variables the appropriate data type.

(Joe:) If you think you are running slow, and you don't know if it's you or other programs [on the system] there is a pair of functions within VAX-DTR which will allow you to initialize a timer and then show the amount of elapsed time between the initialization and the show time. You can place them around various sections of code and find those sections that are actually running slow. It will give you information about elapsed clock time, CPU time, page faults, etc., and that's very helpful. If something is not running as fast as you think it ought to, you can go and look and decide if it's your process or someone else who is hogging the system [if CPU time is small but elapsed time is large]. Bart:) on DTR-11 and PRO-DTR you don't have INIT\$TIMER, but you can do remote DTR and the log file will have some information on times and what DTR is doing with the retrievals. This will also work for VAX-DTR, and is another way to find out what is going on inside DTR. You can always do a remote DTR to your own node.

Why can't I put a READY inside my BEGIN-END loop?

(Andy:) [A suggested source for information on this point is the VAX-DTR internals session], but basically this has to do with the difference between commands and statements. In a nutshell, the reason you can't put a command like READY within a BEGIN-END block is because commands go through one path [when being processed by DTR] and statements go through another. When you BEGIN-END around something, what you have done is created one big statement out of everything within the BEGIN-END block. If you stick a command in there and DTR runs across it, it's not down the right path [internally] to execute it. (Larry:) the main thing is to understand that there are such things as commands and statements and that one can't go in the other. With the way the language is constructed there is little need to do that; there are ways around it. (Joe:) A simpler explanation is that statements manipulate the data within an environment, and commands change that environment. By putting a command within a BEGIN-END loop you've changed the environment while trying to work in it.

Can I read DTR files from another language?

(Dick:) There is no such thing as DTR files. There are RMS files, Rdb files, DBMS, etc. DTR does not create files of it's own. You can read RMS from BASIC, COBOL, or any language. The converse of that is DTR can read files created by other languages; even the editor if you are careful. (Bart:) the problem with the editor is that you may not align everything properly. You may also run into the problem where you create a record definition 80 bytes long, and you go into the editor and type data 80 bytes long but the editor creates a variable length file. When you ready the domain, DTR will give you a warning message that the file types don't match, but it will then go ahead and read it anyway. If DTR gets a record which is too short, it pads it out (and may give an error message); if it's too long, DTR truncates the record (and in the past, especially on the PDP-11, tends to abort). If you are in doubt, put a FILLER field on the end of the record definition to make the record definition too long; you may get warning messages but DTR will go ahead and read the data. You can then write it to another file with

fixed length records and DTR will be happy. (Larry:) a related question is, what if you have a nasty system manager who won't tell you what the file is like and you are trying to read it? The answer is, use the RMS utilities to find out how long the record is, then create a record definition of the same length with one big field with a PIC length the length of the record. EDIT_STRING T(80) [to make the data fit on the usual CRT screen], ready the domain, print some records out, and by looking at it you can usually figure out where the fields are, and revise your record definition. (Bart:) remember to ready the domain read only and shared, until you are certain you have the record definition correct. You don't want to modify anything until you know what it is.

Can you sort on a non-keyed field?

(Dick:) You can sort on any field you have in your record. (Not COMPUTED_BY fields on a PDP-11, but any real field.) On a VAX, it should be any field.

Why can't I prompt for a domain or a field?

(Andy:) DTR is really forgiving, but there are certain features intended to be used in some places and not others. Prompting, when you do a *.prompt, is for value expressions and value expressions only. A value expression is a value for a field, or a piece of text. They don't include things like key words or names of things, which is what a domain is. When you say READY —, what you are prompting for is a value expression, and DTR will say "oh no I won't!". Essentially, the contents of a quoted string is what it grabs, so when you prompt for anything it must be a value expression or piece. There are some workarounds, one being logical names: you prompt for a string, do an FN\$ to create a logical name translation, ready the logical name and DTR will translate one level of logical names.

How many records can I have in my domain? How large a record can I have?

(Bart:) Basically, the number of records you can have in your domain is limited by how large your disk is (or your disk quota if applicable). As for the size of the record: on the PDP-11 if it's very large you will run out of pool space. On the VAX there may be a limit, but I don't know anyone who has hit it. (Comment from audience indicating it had been reached.) There is a system wide RMS limit on the maximum size for any record on a VAX, and I believe it's set around 32,000 bytes. As far as the number of records, it's limited by the amount of disk space, and I've done domains with over 130,000 records. (Comment from audience indicating a user with 6000+ byte records, stating that the application seemed a bit slow, but when the application was up into smaller pieces with relevant sections connected by crosses, it ran faster. Doesn't it make more sense to keep the record size smaller?) (Bart:) It's partially record size, and partially the number of fields. If you don't need all 6000 bytes at once, breaking it up into smaller pieces that most logically go together will save you overhead. The other possibility is to have more than one record definition for the same file and use FILLER to skip over the pieces that aren't needed at that time, and that also cuts down the number of fields that DTR has to know about. Either of those approaches would give an improvement. (User: if you use filler, it cuts down the number of fields, but then you have the same number of FILLER fields.) You use one FILLER field to skip over all of them. (Andy:) One important point we are looking at for *way in the future* is that access to the CDD is very inefficient for metadata. For every attribute you have for every field DTR has to make a call to the CDD. If you have 400 fields, and each field has a

name, a query header, and edit string, a query name, missing value, default value, DTR makes one call for each [at least 2800 calls including the PIC clause]. If you can cut unnecessary attributes, or you have fields that aren't used often and you can skip over with FILLER, DTR jumps over FILLER and it's internal field tree is much, much simpler. Also, less memory is used, as it allocates a big block for each field, and this block is the same size for a field with no attributes or with many attributes. If you can eliminate fields, you save time and memory not allocating blocks. (Larry:) Besides, anyone with a record that has 6000 bytes in it needs to go back and re-evaluate how the data is being structured. (User:) the records were a complete record of our field engineers, including their education, experience, etc. In essence, we had 10 major areas of interest, and instead of 1 record we really had 10. It worked a lot faster [after we changed to 10 records.] (Bart:) Not just speed but other considerations apply: if you give someone write access to that domain, they now have access to everything in there, and do you really want to give them everything at once? From the management standpoint you also want to separate the data.

Can I do menu-driven applications in DTR?

(Larry:) Yes, and I know of about 4 different methods. The first is the way NOT to do it, and that is to use DCL and have it call DTR every time you need to do something (having the menu in DCL). This will work, but it's inefficient: you go through all of the overhead of starting up DTR whenever you want to do something.

I like to use the call interface, and a little program that feeds procedures back to DTR. Essentially, DTR tells the program what it wants to do next, and the program tells DTR to run it [a procedure] next. That works very well. (Dick Azzi:) I like to "pre-compile" DTR. Andy mentioned that anything within a BEGIN-END block is treated as one statement, and DTR always has to parse the next statement it is going to work on. We take maybe 75 to 100 "programs", put them all within a large BEGIN-END block with a menu so DTR treats it all as one statement: it takes 15 to 20 minutes to parse that statement (we bring it up on Monday morning and leave it up until everyone goes home Friday night). Included on the menu are functions "sleep" and "pause", which bring up an FMS screen with a no-echo password so that the user can leave a terminal and get back in only if they enter the same password. This process works well in our application, where we treat DTR as the center of the universe: if we have to do something in DCL we will spawn out of DTR, work in DCL (things like word processing, PHONE, running another program), and then return to DTR where all of the pre-compiled statements are still active, all READYs are still there, etc. This gives a very quick turn-around on menu response.

(Larry:) Another method is with logical names [calling a procedure which has one fixed name: a logical name assignment is made from the menu to translate that logical name to the name of one of a set of "real" procedures]. There are probably other methods as well.

(Mike Nickolas, Bank Ohio:) Another method of doing menus in DTR is to have a simple procedure called DISPLAY_MENU which has a print statement to clear the screen, displays an abbreviated form of the procedure such as ":M1", and that procedure only does one function such as ADD [a record], which would require only a very short compilation time.

(Chris:) I'd like to make an exception about how not to do a menu [using DCL described above]. There are times when a DCL menu is appropriate. If you have several choices on the menu and only one is going to be DTR, the startup delay occurs only after the choice of that option is made. The main menu can come up very quickly in a DCL procedure, and if there is only one which goes to DTR, or there are many options only one or two of which go to DTR, the total delay is less than if you have to go in and out of DTR a lot.

Using logical names is very similar to the "pre-compile" method. The difference is that in your choice statement you use the FN\$CREATE_LOGICAL, and at the bottom of the choice statement you invoke the logical name and DTR will execute the procedure name used in the create logical function: you can even go back and invoke the procedure you are in now. This appears to be recursive use of DTR but in fact is not really working recursively.

Susan Krantz, NKF Engineering: another way I use DCL and DTR together is in DBMS applications where I have a COBOL program using FMS doing an update function and then have a menu in the beginning asking if I want to use the update program or do I want to go into DTR and do my reporting. That's a good combination because they are either changing the database or they are doing report, and once you are in the report module everything is pre-compiled.

(Larry:) as Chris said, if you are doing one-shots then the DCL menu is good, but on the other hand if you are going to switch back and forth between updating and reporting I'd use the call interface and integrate [DTR] right in [to the COBOL program].

Will the DTR Call Interface support new languages?

Ron Swift, Xerox: we use a FORTRAN interface for some of [those applications, such as were discussed for the previous question] which allows us to leave the domains open. It allows us to go through a menu, and appears to speed up tremendously what we're doing. My question is, will there ever be a "C" interface to DTR

(Andy:) Do you mean, will there ever be something which DTR ships as part of its kit to allow you to automatically run with "C"? You can do it today, but you have to create your own DAB. The bottom line is: any language which conforms to the VAX calling standard can use callable DTR today, which means "C" can use it. We have chosen a subset to ship with our kit which ABs, examples, and so forth. We have been asked for "C" in the past and that may come, although it doesn't stop you from using it today, it just means you have to do some legwork up front.

(Bart:) A little advertisement for the DECUS library: if the first person to do it would please submit it to the library then everyone else will get it.

How do I used nested FOR loops?

(i.e., how do I optimize access to two domains?)

Bob Brown, INTEL: In regards to optimization, could someone explain to me how nested FOR loops work, where you put the keys (on the inside or outside): it's kind of difficult to understand from the manual.

(Bart:) [rather than transcribing the problem description, the example below shows the outline of a nested FOR statement being used to access two domains, with records in the second domain being selected according to the match of some field in the second domain equaling a field in the first domain.]

```
FOR domain_1 BEGIN
  FOR domain_2 WITH field_2 = field_1 BEGIN
    —
    work done here on one or both domains
    —
  END
END
```

The field that you are specifying in the second domain (the inner loop) [labeled field_2, belonging to domain_2 in the example above] is the one that should be keyed. This is exactly the same as the example shown earlier for the CROSS, where it was going nokey-key. If it's a VIEW, and you specify the first domain, and then the second domain occurs for a field equal to first domain, it's the second domain where the field should be keyed. For all cases, for a record in the first domain, DTR has to find the matching record in the second domain. [The following illustrates the case of a VIEW. As in the first example, field_2 belonging to domain_2 is the one which normally be keyed.]

```
DEFINE DOMAIN view_domain OF domain_1, domain_2 USING
01 FIRST OCCURS FOR domain_1.
  10 field_1 FROM domain_1.
  —
  other fields from domain_1.
10 SECOND OCCURS FOR domain_2 with field_2 = field_1.
  20 field_2 from domain_2.
  other fields from domain_2.
```

(Larry:) please understand that the field doesn't HAVE to be keyed, but if you want to take advantage of the keys, it must be the second domain that is keyed. (Bart:) If you would like it to run in a reasonable amount of time, the second should be keyed.

Bob Brown: even if the outside loop has an RSE?

(Bart:) Yes, because the outside loop is going to go in the order you specify in the RSE: it may or may not be keyed, depending on how you do it. But if you want the matches on the inner loop to be fast, then the second domain has to be keyed so the retrieval can be on the key.

"Field — is undefined or used out of context"?

(Joe:) [We now have] the infamous "undefined or used out of context" error message. This is probably the most frustrating and common occurrence for beginning users of DTR. There are some important obvious things, such as misspellings, that cause this problem. The real underlying cause is a cry for help from DTR because it does not understand what you have told it. There is something in the command that it does not understand, and when it doesn't understand it doesn't know where it doesn't understand (or maybe not know where it doesn't understand). (Larry:) here's the point: a lot of times when you get "undefined ..." the error is not on the quoted string in the error message, the error is going to be somewhere back upstream from that string. What I tell my users is to put your finger on that string, and where I'm pointing is where the error is [using your right hand, the error is somewhere left and up]. This can happen because DTR continues to parse for a while until finally things don't make sense, and then you get the error: the thing that caused it not to understand could be a comma or a space or word encountered previously. (Joe:) You look at the thing it's complaining about and back: sometimes what it's complaining about is a misspelling, but it's also possible that it's something back up the line. This happens because DTR is parsing and compiling, and it has a context within which it has to communicate with you, trying to understand what you're telling it: at this point it's saying "I don't understand anymore".

When do I use hierarchies? (OCCURS clauses in record definitions)

Ron Wilson, Wilson Concrete: I wonder if there are any rules when one should use "flat" records versus hierarchies?

(Larry:) [missing from the tape]. (Bart:) Basically, if you use an OCCURS, you limit how you can access that subordinate data. If you have a very well defined application where some data is definitely subordinate to a main data piece, and I'm absolutely positively always going to be getting the subordinate data with the main data then an OCCURS might make sense. The em is that when you do do an OCCURS, it makes it difficult to get to the subordinate data only. (Larry:) It's really hard to manipulate within an OCCURS clause. You are better off putting it in another domain and use a CROSS and treat it in a relational way.

(Joe:) I'd like to make a dissenting opinion. There are some applications I've used in a medical database where the data is naturally hierarchical; and because the underlying data is naturally hierarchical, DTR is, in my opinion, the best tool for accessing hierarchical data. There are certain prices you must pay in order to do that, but I would argue the other way around. If the data is naturally hierarchical in the way it's used, it should be stored hierarchically, either in an OCCURS within a domain, or in separate domains using a VIEW which in effect creates a hierarchy. (Bart:) [section missing from tape].

How do I pass information from DTR back to DCL?

[name of questioner missing from tape:] ... that logical names created through DTR are user mode logical names. Is there any way that [DTR can create logical names in other modes; question wasn't finished as Andy Schneider was shaking his head "no"]. We use DTR in certain situations to pass values out to DCL and use those values: now we can't do that.

(Andy:) FN\$CREATE_LOGICAL creates a user mode logical name for DTR's purposes only, because 9 people out of 10 will use the logical name while in DTR to optimize [an application] and don't want to have it "kicking around" afterwards so everyone picks it up. What I would suggest is if you want the logical name to be [in existence] afterwards is to use a DTR procedure to create an indirect command file that you execute when you leave DTR to create the logical names. (questioner:) just write it to a file basically. (Andy and Larry:) or create your own function [to create a logical name in some table or mode other than user]. And submit it to the DECUS library.

Preparing Datatrieve Plots for Printing

Donald E. Stern, Jr. - Warner Lambert Co., Milford, CT

One of the shortcomings of VAX Datatrieve is the inability to directly print Datatrieve plots on a system printer. The output of a Datatrieve is in ReGIS graphics code and very few printers support this protocol, *even Digital printers*. To print Datatrieve plots on a printer, such as an LN03 or LA100, the ReGIS code must be converted to SIXEL graphics code. This can be accomplished using DECslide or through the use of another utility (see, for example, my article in the May 1986 issue of the *Wombat Examiner and 4GL Dispatch*.) Once converted, the plot can be printed using the standard DCL PRINT command.

Some of the Datatrieve plots, such as pie charts and bar charts, are normally filled with color (or shading if monochromatic) and an LN03 cannot deal with color. Figure 1 illustrates the effect of a color filled plot when converted and printed.

A new plot, called MONO, was created by deleting a single line of code from the DEC supplied HARDCOPY plot. Figure 2 is the analog of Figure 1 but processed with the MONO plot before conversion.

```
!This Datatrieve plot will convert a color filled DTR plot into
!a monochromatic plot which can be converted to SIXEL and printed.
!
```

```
!Written by: Donald E. Stern, Jr. - September 4, 1986
!
```

```
DELETE MONO;
REDEFINE PLOT MONO
ENTRY 0
BEGIN
  PLOT HOUSEKEEP 4
  OUTPUT SEGMENT 5
  CLEAR SEGMENT 4
  SET SEGMENT 4
  OUTPUT SEGMENT 5.4.6
  PLOT HOUSEKEEP 2
END
END_PLOT
```

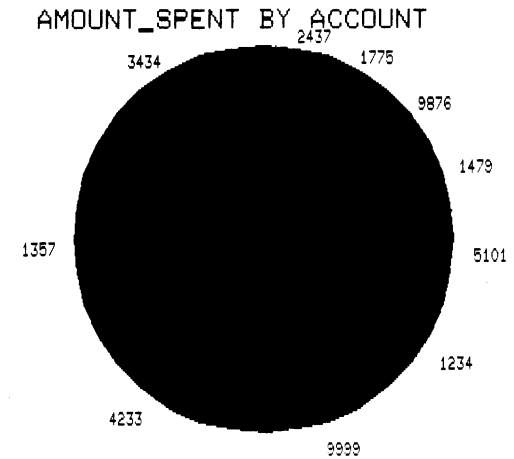


Figure 1 - Example of Color Filled DTR Plot

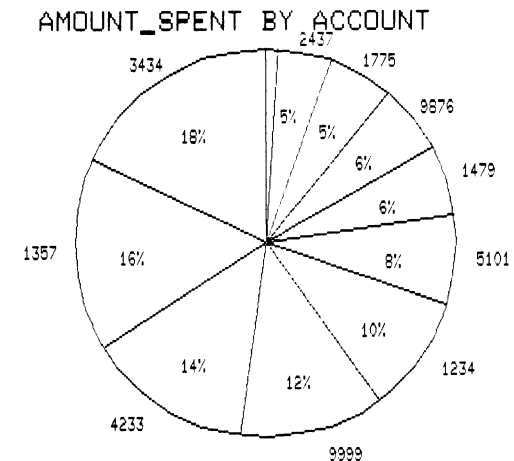


Figure 2 - DTR Plot Post-processed with the MONO Plot



edusig

EDUSIG

Chairman

Robert A. Shive, Jr.
Millsaps College
Jackson, MS

PSS Coordinator VAX Systems SIG Liaison

Donald C. Fuhr
Tuskegee Institute
Tuskegee Institute, AL

Symposium Coordinator

Sue Bates
Northwestern Michigan College
Traverse City, MI

Administrative Applications Coordinator

Dave Cothrun
Taft College
Taft, CA

Communications Committee Representative

Robert W. McCarley
Millsaps College
Jackson, MS

Courseware Coordinator

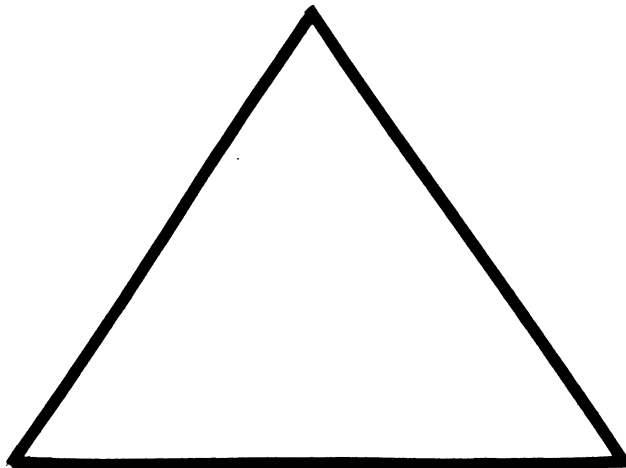
Mary Jac Reed
Off Comp Based Instruction
Newark, DE

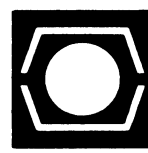
Newsletter Editor

Fred Bell
Taft College
Taft, CA

DEC Counterpart

Gary Finerty
Digital Equipment Corporation
Marlboro, MA





DECUS

GRAPHICS

GRAPHICS

Chairman

William Kramer
University of Delaware
Newark, DE

Library Coordinator

Mike McPherson
Michigan State University
East Lansing, MI

Symposium Coordinator

Bijoy Misra
Smithsonian Institution
Cambridge, MA

Standards Coordinator

Jim Flatten
Ames Lab
Ames, IA

Newsletter Editor

Michael P. Anton
Houston, TX

Volunteer Coordinator

Dick McCurdy
University of Florida
Gainesville, FL

Associate Newsletter Editor

Charles D. Carter
Huntington Alloys, Inc.
Technology Department
Huntington, West Virginia

Library Committee

James M. Turner
Saber Technology
San Jose, CA

Workstation Working Group Coordinator

Bob McCormick
Video Communications, Inc.
Feeding Hills, MA

DEC Counterpart

Rick Berzle
Digital Equipment Corporation
Spit Brook, NH

Engineering Graphics Working Group Coord.

Eric Rehm
Gonzaga University
SPOCAD
Spokane, WA 99258

Information Officer

Mike York
Boeing Computer Services
Seattle, WA

Session Note Editor

Carol Schwob
Florida Atlantic University
Academic Computing
Boca Raton, Florida 33431

Human Interface Working Group Coord

Dottie Elliott
Research Triangle PK, NC

Data Display Working Group Coord

Joy Williams
Eaton Corp.
Southfield, MI



H M S

HMS

HMS**Chairman VAX SIG Liaison**

Thomas J. Provost
MIT/LNS Bates Linac Facility
Middletown, MA

Product Planning Coordinator

George Hama
Synergistic Technology
Cupertino, CA

Symposium Seminar Coordinator

Mike Allen
Lawrence Livermore National Labs
Livermore, CA

Communications Coordinator

John G. Hayes
Information Systems - S. Central Bell
Birmingham, AL

Publications Coordinator (Editor)

Bill K. Walker
Monsanto Research Corp.
Miamisburg, OH

Session Notes**DAARC SIG Liaison**

Bill Tippie
Kinetic Systems Corp.
Lockport, IL

Standards Coordinator**CAMAC Working Group Coordinator**

Peter Clout
Los Alamos National Lab
Los Alamos, NM

LUG Coordinator

Gregg Giesler
Los Alamos Science Lab
Los Alamos, NM

Pre-Symposium Seminar Coordinator

Mike Allen
Lawrence Livermore National Labs
Livermore, CA

TOEM (Chips % Boards)

Jack J. Peterson
Horizon Data Systems
Richmond, VA

HHK (Hardware Hints & Kinks)

Wayne Kesling
Monsanto Research Corp.
Miamisburg, OH

UNIBUS Hardware

Ron Bogue
LIV Aerospace & Defense Co.
Dallas, TX

Performance Measurement Coordinator

William Wallace
600 W. Washington St.
Peoria, IL

CAMAC Coordinator

Peter Clout
Los Alamos National Lab
Los Alamos, NM

CSS Coordinator

Pratap Gohel
E.I. Dupont
Ingleside, TX

Networks SIG Liaison

Sandra Traylor
Target Systems
Yorba Linda, CA

VAX SIG Liaison

Dave Schmidt
5100 Centre Avenue
Pittsburgh, PA

DAARC SIG Liaison

Bill Tippie
Kinetic Systems Corp.
Lockport, IL

UNISIG SIG Liaison

Jim Livingston
1 Results Way
Cupertino, CA

SITE SIG Liaison

Emily Kitchen
A.H. Robbins Co.
Richmond, VA

RT-11 SIG Liaison

Gary Sallee
Sallee Software Consulting
Yorba Linda, CA

RSX SIG Liaison

Hans Jung
Associated Press
New York, NY

Members-At-Large

Mike Rembis
American Dade
Costa Mesa, CA

Hans Dahlke
Richland, WA

Jim Cutler
EDS Tower, 26533 Evergreen
Southfield, MI

DEC Counterparts

Terminals
Nina Abramson
Digital Equipment Corporation
Maynard, MA

TOEM (Chips & Boards)
Art Bigler
Digital Equipment Corporation
Marlboro, MA

Diagnostic
George D. Cooke
Digital Equipment Corporation
Maynard, MA

Storage
Marilyn Fedele
Digital Equipment Corporation
Maynard, MA

MSD (Micro Systems Development)
Roy Rodgers
Digital Equipment Corporation
Maynard, MA

Printer Products
Frank Orlando
Digital Equipment Corporation
Maynard, MA

DECUS Europe Liaison
Hans Zoller



**THE
DEVIAS
LETTER**

CONTRIBUTION GUIDELINES

Chairman

John Roman
Monsanto Agricultural Products Co
Mail Stop GG3I
700 Chesterfield Village Parkway
Chesterfield MO 63198
(314) 537-7044

Newsletter Editor

Frank R. Borger
Radiation Therapy
Michael Reese Medical Center
Lake Shore Drive @ 31st St
Chicago, IL 60616

WHIMS Coordinator

Kathleen Anderson
Eaton Information Management
System Division
Hampton, VA

RSX Liaison

Ray French
Boeing Computer Services
Seattle WA

Member-at-Large

Doug Reno
Abbot Laboratories
North Chicago, IL

DEC Counterpart

Mike Reilly
Digital Equipment Corp.
Maynard, MA

Chairman Emeritus

Bob Curley
Division of Medical Physics
University of Pennsylvania
Philadelphia PA

Symposium Coordinator

Mike Robitaille
Grumman - CTEC, Inc.
McLean, VA

Library Coordinator

Bob Schuldt
INCO INC.
McLean, VA

Member-at-Large

Kerry Wyckoff
Salt Lake City, UT

DEC Counterpart

Tim Leisman
Digital Equipment Corp.
Stow, MA

DEC Counterpart

Bob Mack
Digital Equipment Corp.
Landover, MD

Contributions for the newsletter can be sent to either of the following addresses:

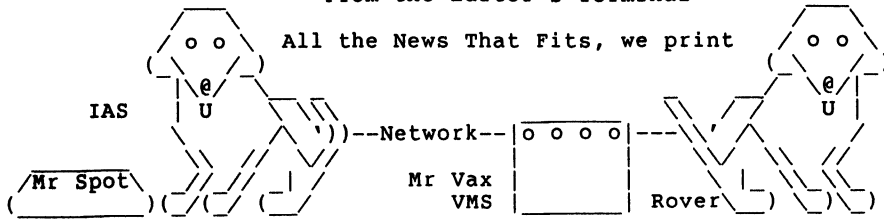
Editor, IAS SIG Newsletter
DECUS U.S. Chapter
219 Boston Post Road, BP02
Marlboro, MA 01752

Frank R. Borger
Michael Reese Medical Center
Department of Radiation Therapy
Lake Shore Drive at 31st St
Chicago, IL 60616

Contributions of letters, articles, important SPR's etc will be accepted in any form, (including notes jotted in pencil on gravy-stained tablecloths.) Contributions will be much more graciously accepted in one of the following formats:

1. Non machine readable sources, (SPR's etc,) should be reasonably dark to insure good photocopying. Text whatever should be the equivalent of 66 lines at 6 lpi, with 4-line top margin, 5-line bottom margin, left-margin 10, right margin 74 at 10cpi.
2. Machine readable sources may be submitted on 9-track Mag-tape, (800,1600, or 6250 BPI,) DEC-tape II, DecMate floppies, or whatever. We're not fussy, we'll even accept paper tape or cards. Preferred format is DOS or BRU for tapes, Files-11 for DEC-tape II.
3. 1200 baud dial-up modems are available on our IAS system and our VAX, with various servers available. Give the editor a call at (312)-791-2515 (preferably later in the day,) to obtain access information, etc.
4. If long distance dialout is not possible on your system, we'll be willing to call your system and do the work, (unless you want to transfer the entire manual set at 300 baud.)

From the Editor's Terminal



At the time I am writing this, Fall DECUS is about 3 weeks away. The deadline for submitting this issue is advanced, due to DECUS, and due to an upcoming family deadline of mine. (Our second child is supposed to arrive on the 21st of September, but if he's late, it could really throw a monkey wrench into my going to fall DECUS. Now that's a deadline.) By the time you read this, DECUS will be over and you will want the latest news, but it won't be in until the next issue. The result is that twice a year, this newsletter becomes a lame duck. Oh well, I promise I'll get news to you as quickly as I can.

This month's Program of the Month club program is one we use to solve a problem that often occurs with large disks. You write a nice program, then a month or so later you ask yourself, "Now where did I put that program?" VMS style named directories help in giving you an idea of what is in each directory, but really don't tell enough. A second problem is keeping track of just who is responsible for having 32 copies of the same FORTRAN source file in UIC [123,45]. We solved both problems by using a series of programs that operate on a Master disk directory. Directory information is stored in the following format.

```

U I C  USER  DESCRIPTION
[  1,  2] Dan   Help files etc
[  1,  3] Dan   Lost files
[102,  1] Frank Frank's master account

```

Options let you add, delete, or modify entries, compare the directory with a directory of [0,0], (to make sure all UIC's are accounted for,) and to do a list including total block usage.

In this month's tale of two systems, we talk about differences in calling system routines. Turns out that VMS looks an awful lot like IAS/RSX.

My August editorial got mentioned in the Digital Review, along with some choice comments from other editors and DEC people. Its nice to know that someone out there is reading this.

A final plea. If you haven't sent in a user survey form yet, please go back to the October issue, get the form, fill it out, send it in.

10 Years Ago This Month

The Multi-tasker continued to report several interesting SPR's.

1. Some DEC supplied command files for version 1.1 of IAS, (such as [1,1]PIPBLD.COM specified explicit version numbers for the task files. This could cause all sorts of problems, including systems with no functional copy of PIP.
2. Some software, in particular EDI, SLP and TKB did not always use the current default device and UIC. This created problems for SAVE/UNSAVE in EDI, indirect files in SLP and ODL files in TKB.
3. A FCS bug, (FCS would continue to read the same record if input buffer overflow occurred on a read,) would cause PDS to loop in command files that contained over 80 character records. PDS would continually print "COMMAND I/O FAILURE".

The European RSX sig report contained a report of the NEW features for version 3.0 of 11M. 11M would finally have loadable drivers, ANSI mag-tape support, a macro CREF, and would provide login protection, similiar to the use of HELLO and BYE in 11D. (If you thing HELLO and BYE even now provide any system protection at all, I've got a bridge for sale in NY, cheap! Editor) Several lively discussions centered on the topic "RSX11D -- Dead or Alive?" To counteract negative reactions, DEC presented a report on the next release of 11D, version 6.2, (ah memories.) New were a completely re-worked TT handler, good time-slicer, updated support for new disks and the new PDP11/34, and an overlapped seek RK05 handler, a file compare utility, (CMP) and capability of generating an RSX11S system. (Ah yes, 11D (dad) and 11M (mom) could both give birth to 11S (son). Then DEC blew it by coming up with IAS, which had no place to fit in the scheme of things. If they came up with something like 11P, (for patriarch,) things would have been much simpler.)

MICHAEL REESE MEDICAL CENTER DEPARTMENT OF MEDICAL PHYSICS COMPUTER
LISTING OF DR0:[001,001]MASTER.BAS;40 ON 5-SEP-86 AT 10:03:31

```
10 ! master.bas program to select master.inf file manipulation pgms
11 clear
20 print "          Available options are"
30 print "C          Change accounting or descriptor for UIC"
40 print "L          List directory"
45 print "S          Search directory for string match"
50 print "I          Insert new UIC into file"
60 print "D          Delete UIC from file"
65 print "N          Create new (blank) master file on another disk"
67 print "X          Compare master directory and master.inf"
68 print "B          List including block usage"
70 print
75 if end goto 200
80 set upper on
81 input "Change, List, Search, Insert, Delete New or Compare
(C,L,S,I,D,N,X) ";z1$
82 set upper off
85 print chr$(12)
90 if z1$="C" then run "LB:UPDMAS"
100 if z1$="L" then run "LB:LISMAS"
105 if z1$="S" then run "LB:SEARCHMAS"
110 if z1$="I" then run "LB:INSMAS"
120 if z1$="D" then run "LB:DELMAS"
125 if z1$="N" then run "LB:COPMAS"
126 if z1$="X" then run "LB:MASTERSRD"
127 if z1$="B" then run "LB:MASTERBLK"
130 goto 80
200 call "SPAWNB"("PIP [1,1]MASTER.INF/PU")
220 exit
230 end
```

MICHAEL REESE MEDICAL CENTER DEPARTMENT OF MEDICAL PHYSICS COMPUTER
LISTING OF DR3:[001,001]UPDMAS.BAS;3 ON 5-SEP-86 AT 10:04:32

```
10 ! PROGRAM UPDMAS.BAS
11 ! TO UPDATE ENTRIES IN ACCOUNT MASTER INDEX
20 DIM A$(80),B$(80),N$(40),T1$(3),R$(80)
30 N$="MASTER.INF/RN/LN:80/UP"
40 OPEN #3,N$
50 R$="UNASSIGNED/LARRY/FRANK/DAN/DEANA/HARVEY/BEVERLY/HANS"
60 R1=7 : ! DEFINE MAX NUMBER OF PERSONS RESPONSIBLE FOR ACCOUNTS
70 IF END THEN 600
100 INPUT "ACCOUNT TO UPDATE - (NO BRACKETS) ",G,U
110 FOR I=1 TO NRC(3)
120 INPUT LINE #3'I,A$
130 B$=SBS$(A$,2,8)
140 B$=LTR$(TRM$(B$))
150 T1$(1)=PIECE$(B$,"",1)
160 T1$(2)=PIECE$(B$,"",2)
170 G1=VAL(T1$(1)) : U1=VAL(T1$(2))
```

```
180 IF G1=G AND U1=U THEN 300
190 NEXT I
200 PRINT "ENTRY NOT FOUND"
210 GOTO 100
300 PRINT "ACCOUNT: ";SBS$(A$,1,9)
310 PRINT "CURRENT OWNER: ";PIECE$(R$,"/",VAL(SBS$(A$,10,2))+1)
320 PRINT "RESPONSIBILITY CHOICES:"
330 FOR J=1 TO R1
340 PRINT " "+STR$(J)+" " "+PIECE$(R$,"/",J+1)
350 NEXT J
360 INPUT LINE B$
365 IF LEN(B$)=0 THEN LET J1=VAL(SBS$(A$,10,2))
366 IF LEN(B$)<>0 THEN LET J1=VAL(B$)
370 IF J1<0 OR J1>R1 PRINT "INVALID ANSWER " : GOTO 320
420 B$=RJS$(STR$(J1),2)
430 CALL "INSTRG"(A$,B$,10)
440 GOTO 500
500 PRINT "DESC: ";SEG$(A$,12,80)
510 INPUT LINE "NEW DESC: ",B$
520 IF LEN(B$)=0 THEN 540
530 CALL "INSTRG"(A$,B$,12,69)
540 PRINT #3'I,A$
550 GOTO 100
600 CLOSE
605 IF END THEN 0
610 RUN "LB:MASTER"
```

MICHAEL REESE MEDICAL CENTER DEPARTMENT OF MEDICAL PHYSICS COMPUTER
LISTING OF DR3:[001,001]LISMAS.BAS;52 ON 5-SEP-86 AT 10:04:41

```
100 ! PROGRAM LISMAS.BAS
110 ! TO LIST ACCOUNT MASTER INDEX
120 DIM A$(10),B$(69),C$(1),D$(80),R$(80),E$(6)
130 R$="[1,1]MASTER.INF/LN:80/RO/EN:320"
140 OPEN #3,R$
141 SET UPPER ON
142 INPUT "OUTPUT DEVICE (TI:,LP:,LP1: [DON'T FORGET THE ':']) ",B$
143 SET UPPER OFF
144 if b$="" then let b$="TI:" : goto 148
145 IF B$<>"TI:" AND B$<>"LP:" AND B$<>"LP1:" THEN
PRINT "** INVALID CHOICE **";CHR$(7) : GOTO 142
148 OPEN #4,B$+"/LN:90/WR"
150 R$="          LARRY FRANK DAN  DEANA HARVEYBEV  HANS  "
160 PRINT #4,"          U I C  OWNER  DESCRIPTION"
170 PRINT #4,"          -----"
180 INPUT LINE #3,D$
190 A$=SBS$(D$,1,10)
200 C$=SBS$(D$,11,1)
210 B$=SBS$(D$,12,69)
220 C=ASC(C$)
230 IF C=32 THEN C=C+16
240 C=6*(C-48)+1
250 E$=SBS$(R$,C,6)
```

```

260 J=0
270 ! TRUNCATE DESCRIPTION OF UIC
280 D=POS(B$, " ")
290 B$=SBS$(B$,1,D)
300 PRINT #4, " ";A$;E$;" ";;B$
310 GOTO 180
320 CLOSE
330 RUN "LB:MASTER"
340 END

```

MICHAEL REESE MEDICAL CENTER DEPARTMENT OF MEDICAL PHYSICS COMPUTER
LISTING OF DR3:[001,001]SEARCHMAS.BAS;7 ON 5-SEP-86 AT 10:05:00

```

100 ! PROGRAM SEARCHMAS.BAS
110 ! TO SEARCH ACCOUNT MASTER INDEX FOR STRING
120 DIM A$(10),B$(69),C$(1),D$(80),R$(80),E$(6),M1$(10),D1$(80)
130 R$="[1,1]MASTER.INF/LN:80/RO/EN:320"
135 if end then 320
140 OPEN #3,R$
141 SET UPPER ON
142 INPUT "OUTPUT DEVICE (TI:,LP:,LP1: [DON'T FORGET THE ':']) ",B$
143 SET UPPER OFF
144 if b$="" then let b$="TI:" : goto 148
145 IF B$<>"TI:" AND B$<>"LP:" AND B$<>"LP1:" THEN
PRINT "* INVALID CHOICE *";CHR$(7) : GOTO 142
148 OPEN #4,B$+"/LN:90/WR"
150 R$=" LARRY FRANK DAN DEANA HARVEYBEV HANS "
155 input "Enter up to 10-character search string ";m1$
157 if len(m1$)<1 then 320
160 load "LB:[1,202]UPPCAS.ATK"
165 call "UPPCAS"(m1$,m1$)
180 PRINT #4, " U I C Owner Description"
185 PRINT #4, " -----"
190 INPUT LINE #3,D$
192 call "UPPCAS"(D$,D1$)
195 if pos(D1$,m1$)>0 then 205
200 goto 190
205 A$=SBS$(D$,1,10)
210 C$=SBS$(D$,11,1)
215 B$=SBS$(D$,12,69)
220 C=ASC(C$)
230 IF C=32 THEN C=C+16
240 C=6*(C-48)+1
250 E$=SBS$(R$,C,6)
260 J=0
270 ! TRUNCATE DESCRIPTION OF UIC
280 D=POS(B$, " ")
290 B$=SBS$(B$,1,D)
300 PRINT #4, " ";A$;E$;" ";;B$
310 GOTO 180
320 CLOSE
330 RUN "LB:MASTER"
340 END

```

IAS-5

MICHAEL REESE MEDICAL CENTER DEPARTMENT OF MEDICAL PHYSICS COMPUTER
LISTING OF DR3:[001,001]INSMAS.BAS;3 ON 5-SEP-86 AT 11:01:10

```

10 ! PROGRAM MULMAS.BAS
11 ! TO ENTER MULTIPLE IN ACCOUNT MASTER INDEX
20 DIM A$(80),B$(80),N$(40),T1$(3),V(2),R$(80)
30 N$="MASTER.INF/FX/LN:80/AP"
40 OPEN #3,N$
50 R$="UNASSIGNED/LARRY/FRANK/DAN/DEANA/HARVEY/BEVERLY/HANS"
60 R1=7 : ! DEFINE MAX NUMBER OF PERSONS RESPONSIBLE FOR ACCOUNTS
70 IF END THEN 600
100 INPUT "ACCOUNT TO INSERT - (NO BRACKETS) ",G,U
105 A$="["+FRMT$(G,3)+","+FRMT$(U,3)+"]"
320 PRINT "RESPONSIBILITY CHOICES:"
330 FOR J=1 TO R1
340 PRINT " "+STR$(J)+" "+PIECE$(R$,"/",J+1)
350 NEXT J
360 INPUT LINE B$
370 IF LEN(B$)=0 THEN LET J1=0
371 IF LEN(B$)<>0 THEN LET J1=VAL(B$)
410 IF J1 < 0 OR J1 > R1 THEN 450
420 B$=RJS$(STR$(J1),2)
430 CALL "INSTRG"(A$,B$,10)
440 GOTO 500
450 PRINT "ERROR IN YOUR ANSWER"
460 GOTO 360
500 INPUT LINE "DESCRIPTION: ",B$
520 IF LEN(B$)=0 THEN B$=""
530 CALL "INSTRG"(A$,B$,12,69)
540 PRINT #3,A$
550 GOTO 100
600 CLOSE
605 IF END THEN 0
608 B=0
609 CALL "SPAWN"("SRT MASTER.INF=MASTER.INF/SI:80/PR:R/KE:2.3:6.3",E
610 RUN "LB:MASTER"

```

MICHAEL REESE MEDICAL CENTER DEPARTMENT OF MEDICAL PHYSICS COMPUTER
LISTING OF DR3:[001,001]DELMAS.BAS;4 ON 5-SEP-86 AT 10:06:09

```

100 ! program delmas.bas
110 ! to remove a uic from the account master index
120 dim a$(80),b$(80),c$(1),d$(80),r$(80),e$(6),t1$(3),v(2),an$(1)
130 ! clear flag for line inserted
140 f=0
150 r$="[1,1]MASTER.INF/LN:80/RO/END:330"
160 open #3,r$
170 R$="[1,1]MASTER.INF/FX/LN:80/WR"
180 open #4,r$
190 input "Account to delete - (no bracket) ",g,u
200 input line #3,a$
210 b$=sbs$(a$,2,8)

```

IAS-6

```

220 b$=ltrim$(trim$(b$))
230 t1$(1)=piece$(b$,"",1)
240 t1$(2)=piece$(b$,"",2)
250 g1=val(t1$(1)) : u1=val(t1$(2))
260 if g1=g and u1=u then goto 280
270 goto 300
280 print "UIC deleted"
282 set upper on : input "Delete another ";an$ : set upper off
285 if an$<>"Y" then 200
287 input "Account to delete - (no bracket) ",g,u
290 goto 200
300 ! write current descriptor to new file
310 print #4,a$
320 goto 200
330 close
340 run "LB:MASTER"
350 end

```

MICHAEL REESE MEDICAL CENTER DEPARTMENT OF MEDICAL PHYSICS COMPUTER
LISTING OF DR3:[001,001]COPMAS.BAS;2 ON 5-SEP-86 AT 11:01:42

```

100 ! PROGRAM COPMAS.BAS
110 ! TO CREATE A DUMMY (BLANK) MASTER INDEX FILE ON SOME OTHER DISK
120 DIM A$(80),R$(80)V,DV$(4)V
130 INPUT "TARGET DEVICE FOR MASTER FILE", DV$
140 R$=DV$+"[1,1]MASTER.INF/FX/LN:80/WR"
150 OPEN #4,R$
160 A$="[ 1, 1] 1SYSTEM FILES, STB'S, LIBRARIES, ETC."
170 PRINT #4,A$
180 CLOSE
190 RUN "LB:MASTER"
200 END

```

MICHAEL REESE MEDICAL CENTER DEPARTMENT OF MEDICAL PHYSICS COMPUTER
LISTING OF DR3:[001,001]MASTERSRD.BAS;72 ON 5-SEP-86 AT 10:06:27

```

10 ! program to compare uic's under master.inf and srd.cmp
11 ! gc,pc are uic from master program
12 ! gb,pb are uic from srd
13 q=0
14 call "SPAWN"("SRD SRD.CMP=[0,0]/NA",q)
30 open #3, "MASTER.INF/RO"
40 open #4, "SRD.CMP/RO"
41 input line #4, l$
42 print l$
44 input line #4, l$
46 if l$ = " 000000 .DIR" then 50
48 goto 44
50 e3=0 : e4=0 : ! e3=end of master.inf e4=end of srd directory file

```

```

100 if end #3 then 109
102 input line #3,l$
105 gc=val(sbs$(l$,2,3)) : pc=val(sbs$(l$,6,3))
106 goto 110
109 e3=1
110 if end #4 then 119
110 a$=[ 1, 1] 1SYSTEM, LOG FILES, LIBRARIES, SYMBOL TABLES
113 gb=val(sbs$(a$,3,3))
114 pb=val(sbs$(a$,6,3))
115 goto 120
119 e4=1
120 if e3*e4=1 then goto 450
125 if gb>gc goto 400
130 if gb<gc goto 300
131 ! special case if they match and master.inf is [nnn,0]
132 if pc>0 goto 150
133 ! skip input from srd file until no match
134 if end #4 then 138
135 input line #4, a$ : gb=val(sbs$(a$,3,3)) : pb=val(sbs$(a$,6,3))
136 if gb=gc goto 135
137 goto 140
138 e4=1
140 ! now get next line from master.inf
142 if end #3 then 145
143 input line #3, l$ : gc=val(sbs$(l$,2,3)) : pc=val(sbs$(l$,6,3))
144 goto 120
145 e3=1
150 if pb>pc goto 400
152 if pb<pc goto 300
160 goto 100
300 ! srd has lower uic so real uic not in master file
301 if gb+pb=0 goto 310 : ! don't mention 0,0
305 if e4<>1 then print "UIC [";frmt$(gb,3);";";frmt$(pb,3);"] not
in MASTER.INF file"
310 if e3=1 then goto 345 : ! skip if at end of master file
315 if end #4 then 340
317 ! read new uic from srd file
320 input #4,a$
322 gb=val(sbs$(a$,3,3)) : pb=val(sbs$(a$,6,3))
330 goto 120 : ! and try again
340 e4=1 : ! show my end of file if at end
345 if e3<>1 then goto 400 : ! try other report if not both ended

```

MICHAEL REESE MEDICAL CENTER DEPARTMENT OF MEDICAL PHYSICS COMPUTER
LISTING OF DR3:[001,001]MASTERSRD.BAS;72 ON 5-SEP-86 AT 10:06:27

```

350 goto 450
400 ! uic does not exist but is in master file
405 if e3<>1 then print "UIC [";frmt$(gc,3);";";frmt$(pc,3);"] not
in master directory"
410 if e4=1 then goto 445 : ! skip if end of srd file
415 if end #3 then 440
420 input line #3, l$

```

```

425 gc=val(sbs$(1$,2,3)) : pc=val(sbs$(1$,6,3))
430 goto 120
440 e3=1
445 if e4<>1 then goto 300
450 call "SPAWN"("PIP SRD.CMP;*/DE",q)
460 run "LB:MASTER"

```

Macro-16 to Macro-32 A Tale of two Systems

Frank R. Borger
Michael Reese Medical Center

MICHAEL REESE MEDICAL CENTER DEPARTMENT OF MEDICAL PHYSICS COMPUTER
LISTING OF DR3:[001,001]MASTERBLK.BAS;45 ON 5-SEP-86 AT 10:06:38

```

10 ! combine.bas
100 dim i$(128)v,j$(128)v,a$(8)v,b$(8)v
101 set upper on : input "List (H)ere or at (L)P1: ";a$
102 set upper off
103 if a$="L" then run "LB:MASTERBLP"
105 print "Doing PIP total block listing, please wait"
110 er=0 : call "SPAWN"("PIP LB:[1,4]BLOCK.LST=[*,*]/TB",er)
120 open #3,"LB:[1,4]BLOCK.LST/LN:128/RO" : if end #3 then 360
125 print
130 er=0 : call "SPAWN"("PIP /FR",er)
140 input line #3,i$ : i=pos(i$,"[") : if i=0 then 360
150 il=pos(i$,"]") : i2=pos(i$,"") : a$=sbs$(i$,i+1,il-i-1)
160 gr=val(seg$(i$,i+1,i2-1)) : us=val(seg$(i$,i2+1,il-1))
170 input line #3,i$
180 open #5,"MASTER.INF/LN:80/RO/SH" : if end #5 then 280
200 input line #5,j$
210 i=pos(j$,"[") : i1=pos(j$,"]") : i2=pos(j$,"")
220 gx=val(seg$(j$,i+1,i2-1)) : ux=val(seg$(j$,i2+1,il-1))
240 if gr<>gx then 200
250 if ux=0 then 300 : ! match on [nnn,0]
260 if ux<>us then 200
270 goto 300
280 close 5 : print [";a$;"] : goto 340
300 close 5 : print [";a$;"];tab(9);ljs$(sbs$(j$,12),70)
340 x=pos(i$,"Grand") : if x>1 then 355
345 print " ;sbs$(i$,5)
350 goto 140
355 print " ;seg$(i$,5,x-1)
356 print sbs$(i$,x)
360 close
370 call "SPAWN"("PIP LB:[1,4]BLOCK.LST;*/DE",er)
390 RUN "LB:MASTER"

```

- Notes: 1. MASTERBLP is same as MASTERBLK but with output to LP1: via OPEN #4, "LP/WR" and PRINT #4 commands.
2. These programs use version 1 of SORT. Users having version 2 will have to change calling syntax.

In the previous installment of this series, I never even got to comparing IAS system directives with their VMS counterpart, called System services. Things have quieted down here, (no fires to speak of,) so we're back on track, although quite behind my original schedule. (As I write this, Its about 3 weeks to Fall DECUS, so no way will I have a working native VMS version ready.) Oh well, one proverb of management is that you can get conned into committing to an unreasonable deadline, but can't be forced into meeting it.

First, the good news ! The VMS designers started with the IAS/R SX systems as a benchmark. If you have a PDP11 QIOW\$ directive, the VMS SYS\$QIOW is remarkably alike. You won't have to re-learn everything. There are some changes, but by and large they are improvements. There is some bad news, thou, as we will see.

1. Under IAS/R SX you had to specify each directive you were going to use via a .M CALL. Under VMS, the assembler automatically looks thru the system library SYS\$LIBRARY:STARLET.MLB for the macros you specified.
2. Under IAS/R SX, there are 3 different forms of a directive, the DIR\$ form, which pushed the DPB address on the stack, the DIR\$\$ form, which pushed the actual block on the stack, and the DIR\$C form, which created the DPB in a separate P-Sect. Under VMS, the little used \$C form has been dropped, and the syntax for specifying arguments to the two remaining forms has been cleaned up and enhanced.
3. The standard calling format, with parameters separated by commas and positionally dependant still can be used. The preferred form is to use keywords. An example follows

RSX/IAS	VMS
direct\$s #1,#2,,addr	\$direct_s arg1=#1, - arg4=addr, - arg2=#2

Note that in the above case, the 3 arguments did NOT have to be in their correct order, and that one did not have to specify any optional arguments. I like this form. It's inherently more readable, you can't blow things up by having one to many or too few commas. With reasonable keywords, most directives are self documenting.

4. Some bad news. There is still the problem of whether to use #arg or just arg. (Wish I had a dollar for every time I did that the wrong way.) In general, the rules are the same as before.

1. If you are using the stack format, #'s should be used with parameters that are numeric values, and should not be used with parameters that are addresses.
2. If you are creating argument lists in memory, #'s should not be used.

5. Argument passing to system services is more complex than just address versus data. Note also that all argument lists end up containing a single longword with the number of arguments in the lowest byte of the longword, (rest of the longword must be cleared,) followed by n longword arguments. Each argument passes information in one of 3 ways.

1. **BY VALUE**
The longword argument contains the actual data.
2. **BY REFERENCE**
The longword argument contains the address of the actual data.
3. **BY DESCRIPTOR**
The longword argument contains the address of a data descriptor. A descriptor consists of two or more longwords which in turn describe the location, length and data type of the data to be passed. Although all data descriptors do not have the same layout, the following is an example of a descriptor for an input buffer for a text read.

```
READDS: .long 80 ;Length of buffer
        .address - READBF ;Address of read buffer
READBF: .blkb 80 ;Read buffer itself
```

6. Under IAS/RSX if you created a directive parameter block via a direct\$ form, you invoked the system directive via a DIR\$ #DPBADD call, specifying the address of the directive parameter block. The system knew which kind of call you were making, because if the directive parameter block was on the stack, the Directive Identification Code in the lower byte was odd, whereas if it were the address of the DPB, it was even. Under VMS, your DPB can start at an odd address, and the system can't be sure what form you are using. To solve this, you create the directive argument list using a \$name form, and then execute the request by using a \$name_G form of the macro.

7. Status returns are also different. There is no Directive status word. Instead, the status is returned in R0. Also a

positive status return does not indicate success, or a negative indicate failure.

1. The low order 3 bits contain the returned status severity code:

Value	Meaning	Symbolic name
0	Warning	STSSK_WARNING
1	Success	STSSK_SUCCESS
2	Error	STSSK_ERROR
3	Information	STSSK_INFO
4	Severe Error	STSSK_SEVERR

2. The higher order bits, (3 thru 31) contain specific status return values. (For system service routines, only bits 3 thru 15 are used.)

8. This also means that testing for error conditions is different. Comparing the two systems one sees the following:

<pre>IAS dir\$ #direct tst @#DSW bcc Okay . error code here .</pre>	<pre>VMS \$dir_s direct blbs R0,Okay . error code here .</pre>
---	--

Okay:

Okay:

Oh yes, that blbs instruction, its branch on lower bit set. (It's companion is blbc.) These perform the equivalent of BCS and BCC. In general, VMS does not use the "C" bit to communicate success or failure of a subroutine or system call.

If you are going to be doing much program debugging, its a good idea to create the following program:

```
.TITLE DEFINE
$$STSDEF GLOBAL ;Define severity codes
$$SDEF GLOBAL ;Define status codes
.end
```

Compile the program:

macro define

And produce a map

link/noexe/map/full define

This listing of all status returns will be quite valuable if you have to do any debugging. The /full specification for the map file is important so that you get a list of symbols sorted both by value and alphabetically. Its a real bear otherwise to go searching thru an alphabetic list, trying to find a certain value.

It's also important to note the differences between the status return, which are symbols of the form STS\$K_SUCCESS and STS\$K_WARNING and the condition values which are symbols of the form SS\$NOHOMEBLK. (OOH, that last condition value is one I don't want to see.) It also should be noted that SS\$xxxx values include the lower 3 bytes, so have their severity code buried in them. As an example, SS\$BUFFEROVF would be returned if you did a tape or terminal read into a buffer that was too small. Its Hex value of 601 has a STS\$K_SUCCESS value embedded into it. Looking at the last digit of a condition value yields the following:

Last Dig	Severity	Example
0 or 8	Warning	SS\$DIRFUL
1 or 9	Success	SS\$CONTROLY
2 or A	Error	SS\$PARTMAPPED
3 or B	Information	SS\$EOTIN
4 or C	Severe/Fatal	SS\$CONNECTFAIL

Now that we have described the major rule changes, lets look at a concrete example. A simple IO request to write a message to the terminal. In IAS one would do it this way.

```

        .title HELLO
        .mcall qiow$s,exit$s
;
start:  qiow$s  #IO.WVB,#5,#1,,#iosb,,<#MESS,#MESSLN,#40>
        tst    @#DSW          ;Did qio succeed ?
        bcc   1$              ;br if successful
        exit$s                ;else take coward's way out
1$:    .
        .                    (Rest of code here)
        .
        exit$s                ;Program exits here
;
;
mess:  .ascii  /Hello there, terminal person./
messln=.=mess
        .even
iosb:  .word   0,0            ; io status block
;
        .end    start

```

To do it under VMS, one does about the same thing. The only thing different is that you no longer assign luns at task build time, (and do not have the standard terminal lun of 5.) You must therefore explicitly ask for what is called an IO channel to use for your input and output.

```

        .title HELLO
;
        .psect code,rd,nowrt,exe
;
;
        .entry start,^m<R2,R3>          ;Entry mask
        $assign_s -
        devnam=toname, -
        chan=tochan
        blbs   r0,1$                  ;br if successful
        ret                             ;return if not
1$:    $qiow_s func=#IO$_writevblk, -
        chan=tochan, -
        efn=#1,
        iosb=iosb, -
        pl=mess, -
        p2=messln, -
        p4=#32
        blbs   r0,2$                  ;br if successful
        ret                             ;else take coward's way out
2$:    .
        .                    (Rest of code here)
        .
        ret                             ;normal exit here
;
;
        .psect rodata,noexe,nowrt,rd
;
mess:: .ascii  /Hello there, terminal person./
messln: .long  .-mess
toname: .ascii  /sys$output/
;
;
        .psect rwdata,noexe,wrt,rd
;
tochan::.blkw  1                      ;channel number
iosb:: .blkw  1                       ;io status block
iolen::.blkw  1                       ;length
iodev::.blkw  1                       ;device characteristics
;
        .end    start

```

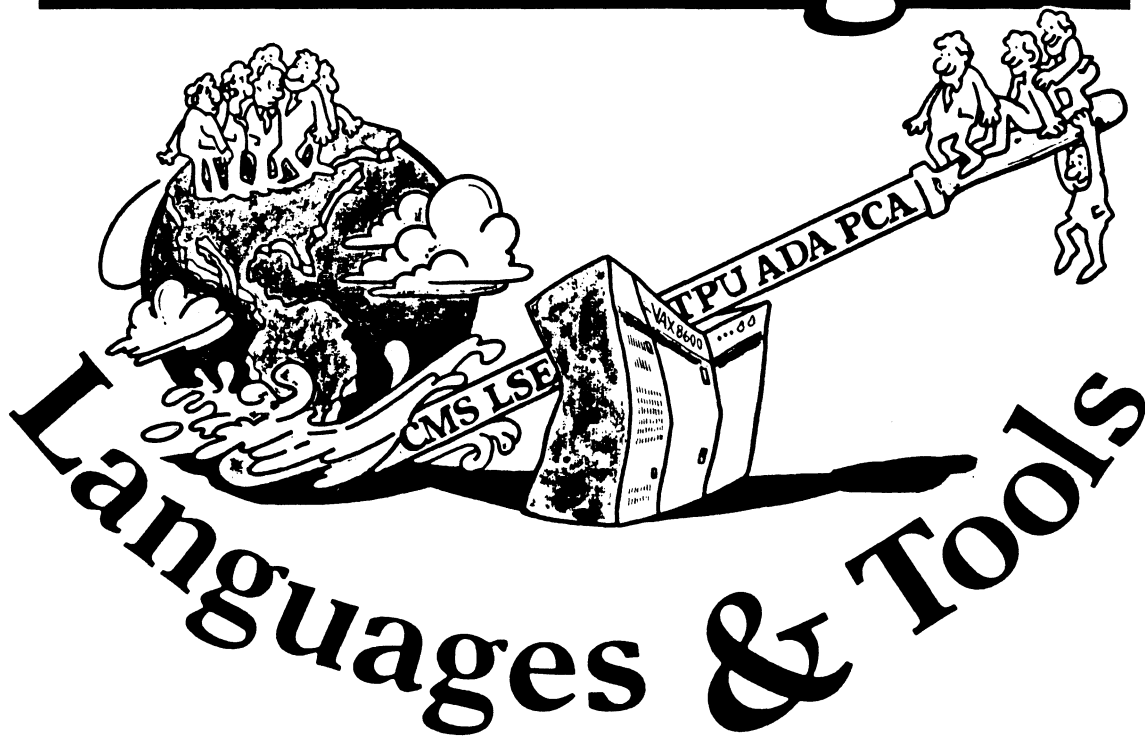
Some points should be made concerning the differences between the above examples:

1. Entry and exit code is different, since your program is a procedure that has been called by another procedure.
 1. The .entry command defines a procedure entry point. In this example it also specifies that registers r2 and r3 are to be saved on entry and restored upon return to the calling procedure.
 2. Since you were "called" rather than being "run" your program does a "ret" rather than an "exit\$s".

2. The \$assign is required to get an io channel, the VMS equivalent of a Lun. The io channel is not a fixed number as with IAS/RSX.
3. The IO status block is twice as big. The io status, (now called the condition value,) is now a word rather than byte value, and an additional longword receives a device specific return.
4. As mentioned before, the odd address boundary is no longer a problem. One does not have to use the even directive after ascii messages to word align addresses again.
5. It's very good practice to separate programs into code, read-only data and read-write data, mainly due to the intelligent debugger available on VMS. Its also good to use Global names liberally, so that the Debugger knows about them.

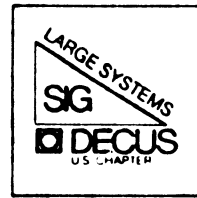
Now seems like a good time to break off this chapter in the series. There are a lot of differences in the two system interfaces, (mainly in the increased number of calls, and to the structural differences in the two systems.) A detailed comparison of equivalent system calls is beyond the scope of this article, (or beyond my willingness to spend my time at the keyboard.) All in all, it was a pleasant surprise for me to see how much of the style of IAS/RSX is present in VMS, (and how some limits and annoyances were noted and taken care of.)

Leverage

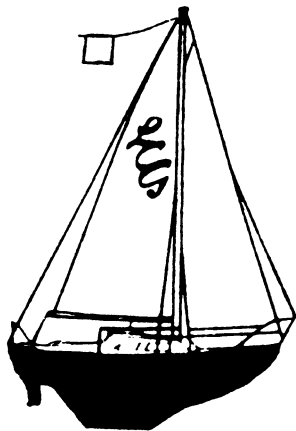


STEERING COMMITTEE MEMBERS LANGUAGES AND TOOLS SIG

Susan Abercrombie Ventrex Laboratories 217 Read Street Portland, ME 04103	Library Rep (207) 773-7231	Celeste LaRock 110 Spit Brook Rd ZK02-3/Q08 Nashua, NH 03062	DEC Counterpart
Bob Awde General Mills 9000 Plymouth Ave N Minneapolis, MN 55427	Steering Committee (612) 540-4432	Jim Livingston Measurex Corporation 1 Results Way Cupertino, CA 95014	Past Chair (408) 255-1500 X4468
Mark Dartelt HSC - Research Development Ctr 555 University Avenue Toronto, Ontario, Canada M5G 1X8	UNISIG Interface (416)598-5955	Dave Martin Hughes Aircraft Company PO Box 92426 Bldg R1, MS C320 Los Angeles, CA 90009	Tape Librarian STUG Interface (213) 648-9927
Doug Bohrer Bohrer & Co. 903 Ridge Rd. Suite 3 Wilmette, IL 60091	Publications Committee (312) 251-9449	Shava Nerad Systems Alternatives 43 State St Montpelier, VT 05602	Steering Committee (802) 229-0823
Gordon Drimble Bldg 180 Labs Area Defence Research Centre Box 2151 GPO Adelaide, S.A. Australia 5001	Australian L&T Interface (61)(8)259-6119 (office)	Al Rizzuto EMC Control, Inc PO Box 242 Cockeysville, MD 21030	Wishlist Coordinator (301) 628-8167 (work) (717) 450-5014 (recorder)
Barb Chase Hughes Aircraft P O Box 92426 Bldg R1 MSC327 Los Angeles, CA 90009	Human Interfaces Coordinator (213) 606-1601 (work)	Don Rosenthal Space Telescope Science Institute Homewood Campus Baltimore, MD 21218	LISP/AI Coordinator (301) 338-4844 (work)
Earl Cory Cory Computer Systems 306 North Nueve Court Camarillo, CA 93010	Symposium Coordinator (818) 706-5385 (work)	Tony Scandora Argonne National Laboratory CMT 205 Argonne, Illinois 60439	RSX Interface (312) 972-7541
Jack Davis Philips Home Interactive Systems 1111 North Shore Drive Knoxville, TN 37919	Modula II Coordinator (615) 558-5206 (work) (615) 588-5800 (switchbd)	Bill Segal Digital Equipment Corp. 110 Spit Brook Rd. ZK01 Nashua, NH 03062	Counterpart Emeritus (603) 881-1263
Jim Flatten Ames Lab 304 Metallurgy Ames, IA 50001	GAPSIG Interface (515) 294-4823 (work)	Kathy Tamer Rockwell International 1840 Nasa Rd./MS ZC01 Houston, TX 77058	Ada Packages Project (713) 333-0827 (work)
Alan Folsom, Jr. Fischer & Porter Co. E. County Line Rd. Warminster, PA 18974	Newsletter Editor (215) 674-7154 (work)	Pat VanMunn Measurex Inc. One Results Way Cupertino, CA 95014	Methods Coordinator PSS Committee Representative (408) 255-1500
Bob Gable Lear Siegler, Instrument Division 4141 Eastern SE MS 121 Grand Rapids, MI 49508	Ada Coordinator (616) 241-8273	Jay Wiley Bechtel Power Corp 12400 East Imperial Highway Norwalk, CA 90650	Standards Coordinator Fortran Coordinator (213) 807-4016 (work)
Dorothy Geiger Wollongong Logistics Group 49 Showers Drive # 451 Mountain View, CA 94040	Intersig Coordinator (415) 962-7160	JR Westmoreland Custom Software Products 6748 Acoma Rd Midvale, UT 84047	Assistant to the Chair C Coordinator Commercialism Task Force TeX/LaTeX Coordinator (801) 535-4784 (work)
Bernd Gliss Max-Planck-Institute HeisenbergstraÙe 1 7000 Stuttgart 80, W. Germany	European Methods, L&T Interface (711) 686-0251 (work)	Melodee Westmoreland Custom Software Products 6748 Acoma Rd Midvale, UT 84047	Recording Secretary (801) 533-2350 (work)
Keith Hare JCC PO Box 381 128 West Broadway Granville, Ohio 43023	DMS & DTR Liaison (614) 587-0157 (work)	Sam Whidden American Mathematical Society 201 Charles St PO Box 6248 Providence, RI 02940	36 bit Coordinator Store Liaison (401) 272-9500 (work)
Howard Holcombe RCA Front & Cooper St. Camden, NJ 08055	DEC Personnel Coordinator (609) 338-4946 (work)	Ed Whipple Lawrence Berkeley Labs University of California Berkeley, CA 94720	CMS/MMS Coordinator Session Chair Coordinator (415) 486-7167 (work)
Kathy Hornbach Lear Siegler/Instrument Division 4141 Eastern SE MS 121 Grand Rapids, MI 49508	Chair Productivity Tools Coordinator Pre-Symposium Seminar Coordinator (616) 241-8800	Louise Wholey Measurex Corp One Results Way Cupertino, CA 95014	VMS Interface (408) 255-1500 X4452 (work)
Mark Kats GTE Govt Systems 100 First Ave. Waltham, MA 02154	Session Notes Editor (617) 466-3437	Jim Wilson QZ Division PO Box 88 Terre Haute, IN 47808	Commercial Languages Interface (812) 299-2121 X271 (work)



DIGITAL EQUIPMENT COMPUTER USERS SOCIETY



AT LARGE

LS

The Newsletter of the Large Systems SIG

SIG STEERING COMMITTEE

SIG Chairperson

Leslie Maltz
Stevens Institute of Technology
Computer Center
Hoboken, NJ 07030
(201) 420-5478; BITNET:LMALTZ@SITVXB;
ARPANET:SIT.MALTZ@CU20B.COLUMBIA.EDU

Symposium Coordinator

Robert C. McQueen
Stevens Institute of Technology
Computer Center
Hoboken, NJ 07030
(201) 420-5454; BITNET:RMCQUEEN@SITVXB;
ARPANET:SIT.MCQUEEN@CU20B.COLUMBIA.EDU

Newsletter Editor

Clyde T. Poole
The University of Texas at Austin
Department of Computer Science
Taylor Hall 2.124
Austin, TX 78712-1188
(512) 471-9551
ARPANET:ctp@sally.utexas.edu

Menu Coordinator

Charles R. T. Bacon
National Institutes of Health
Building 12B Room 2N207
Bethesda, MD 20205
(303) 496-4823
BITNET:CRB@NIHCUDEC

Hardware Coordinator

Clive Dawson
Microelectronics & Computer Technology Corp.
9430 Research Blvd.
Echelon Bldg. #1, Suite 200
Austin, TX 78759
(512) 343-0860
ARPANET/CSNET:CLIVE.MCC

Languages Coordinator

David Edwards
SRI International
MS PN349
333 Ravenswood Ave.
Menlo Park, CA 94021
(415) 859-6136

Systems Software Coordinator

Betsy Ramsey
American Mathematical Society
P.O. Box 6248
Providence, RI 02940
(410) 272-9500 ext. 295
ARPANET:EWR@XX.LCS.MIT.EDU

Special Projects Coordinator

E. F. Berkley Shands
Washington University
Department of Computer Science
P.O. Box 1045
St. Louis, MO 63136
(314) 889-6636
BERKLEY@WUCS.UUCP

Networks Coordinator

Don Kassebaum
Computation Center
University of Texas at Austin
Austin, TX 78712
(512) 471-3241
ARPANET:CC.KASSEBAUM@A20.CC.UTEXAS.EDU

Systems Software Coordinator

Carla Rissmeyer
Computation Center
University of Texas at Austin
Austin, TX 78712
(512) 471-3241
ARPANET:CC.RISSMEYER@A20.CC.UTEXAS.EDU

DEC Counterparts

Dave Braithwaite
Digital Equipment Corporation
Marlboro, MA

Rich Whitman
Digital Equipment Corporation
Marlboro, MA

Reed Powell
Digital Equipment Corporation
Marlboro, MA

MUMPS

MUMPS SIG STEERING COMMITTEE

Chairman

Mark Berryman
Plessey Peripheral Systems
Irvine, CA

VAX Liaison

Coyett A.J. Dese
VA DM&S Verification & Dev. Ctr.
San Francisco, CA

Symposium Coordinator

Chris Richardson
Computer Sciences Corp.
Ridgecrest, CA

Digital Counterparts

Beatrice Walther
Digital Equipment Corp.
Marlboro, MA

Communications Rep.

Mark Hyde
Advanced Computing Services
DeWitt, NY

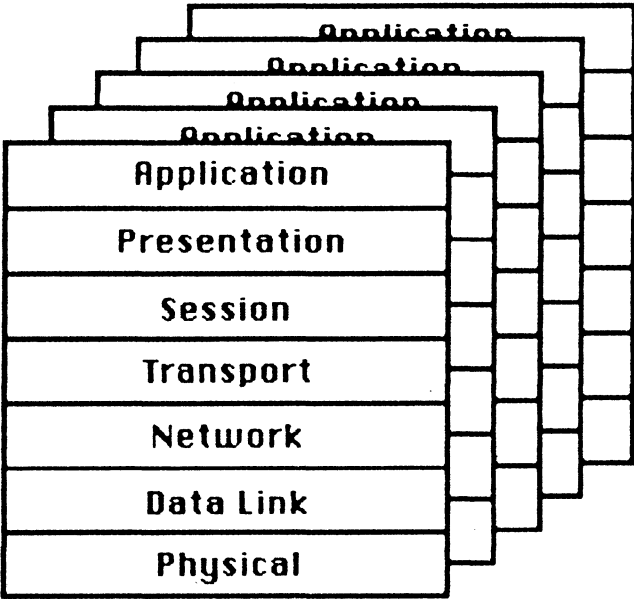
Diane Brown
Digital Equipment Corporation
Marlboro, MA

Newsletter Editor

Janet Berryman
Plessey Systems, Inc.
Irvine, Ca

NETwords

The NETWORKS SIG Newsletter



Networks SIG Steering Committee

Bill Brindley
Chairman
Naval Security Group Command
(202) 282-0527

Sandy Traylor
Symposia Coordinator
Target Systems, Inc.
(714) 921-0112

Jim Ebright
Communications Coordinator
Software Results Corporation
(614) 421-2094

Bill Hancock
Technology/
Standards Coordinator
(214) 495-7353

Vickie Hancock
Newsletter Editor
(214) 495-7353

Carole Greenfield
DEC Counterpart
Digital Equipment Corporation

The Networks Special Interest Group (SIG) is one of 25 SIG's within in Digital Equipment Computer User's Society (DECUS). The main purpose of the Networks SIG is to promulgate information concerning the use, development, and standardization of network products that function or involve Digital Equipment Corporation systems. Additional functions of the SIG include the coordination and scheduling of symposia sessions, providing methods for free-flow communications, publication of the Networks SIG newsletter NETWords, participation in domestic and international standards committees, input to Digital for new products and corrections to existing products, promotion of working groups for special network needs and topics, and many, many other functions.

The Networks SIG Steering Committee invites you to participate in the Networks SIG. There are many ways that you can help the Networks SIG. Some of those include chairing sessions at symposium, participation in the various Networks SIG working groups, participation in special research projects, and others. If you are interested in devoting your time and expertise, contact any of the steering committee members.

DECUS is run entirely by volunteer leadership. Help us make DECUS and the Networks SIG better - take an active part in **your** SIG!



**OFFICE
AUTOMATION**

OFFICE AUTOMATION SIG STEERING COMMITTEE

Chairman

Katherine 'Kit' Trimm
Pivotal, Inc
Tucson, AZ
(602)886-5563

Vice Chairman

Ralph Bradshaw
Johnson and Johnson
Raritan, NJ
(201)685-3434

Communications Committee Rep.

E. Catherine Ditamore
ARA Services
Philadelphia, PA
(215)238-3638

Symposium Coordinator

Mitch Brown
Gen Rad, Inc
Waltham, MA
(617)890-4900

Special Projects

Gene LeClair
HQ Dept. of Army
Washington, DC
(202)697-3234

BOF Coordinator

Ray Kaplan
PIVOTAL, Inc.
Tucson, AZ
(602)886-5563

Newsletter Editor

Therese LeBlanc
T.M. LeBlanc & Assoc.
Wheeling, IL
(312)459-1784

Library

Bob Hassinger
Liberty Mutual Research Center
Hopkington, MA
(617)435-9061

Tape Copy Coordinator

Randall Buck
Columbia Savings
Irvine, CA
(714)863-3030

ALL-IN-1 Working Group

Leo E. Ottley
Evans and Sutherland
Salt Lake City, UT
(801)582-5847

Symposia Assistant

Sal Gianni
Northeast Utilities
Hartford, CT
(213)665-5652

Store Coordinator

Mike Jackson
Air Force Operational
Test and Evaluation Center
Kirtland, AFB, NM
(505)256-0267

Personal Computer SIG Liaison

Cheryl Johnson
Grinnell College
Grinnell, IA
(515)236-2570

In This Issue

From The Editor.....	OA-1-2
-Therese LeBlanc-	
Fixing ALL-IN-1 Bugs.....	OA-2-1
-Tony Ioele0	
12 Bit News.....	OA-3-1
-Bob Hassinger-	
WPS Notes.....	OA-3-2
-Bob Hassinger-	

From The Editor

Our OA Newsletter this month is, as the saying goes, short and sweet. We have an excellent article on ALL-IN-1 bug fixes, some '12 Bit news', and some WPS-Plus and XAL information.

Watch for the latest SIR list and ballet in next months issue (or January at the latest). Please feel free to send any articles or information you would like to share with the rest of our readers. Your participation will keep our newsletter fun and interesting!

Regards,



Therese LeBlanc

FIXING ALL-IN-1 BUGS

Tony Ioele
ARA Services, Inc.
Philadelphia, Pa.

I have found two bugs with All-In-1 and, since the Support Center was not aware of the problems, I'd like to pass the fixes on. The first problem is in the File Cabinet Subsystem, specifically the Multiple Delete (MD) function, which uses the script file WPMAMP.SCP. When the selected folder is the same as that appearing in the Current Item Block (at the start of the function), the following error message will be generated at completion of the function: "The current file cabinet document couldn't be found". The second problem is in the Word Processing Subsystem: the Send (S) function, using the script file WPSAMP.SCP. The script file has a typographical error that causes cryptic error messages to be displayed. A detailed explanation of each bug and the code to correct them follows.

In both cases I decided to take the code on myself and was able to make the necessary changes to correct the problems. The complete script files will be placed on the OA SIG tape at the DECUS Symposium in San Francisco.

In my discussions with the Atlanta Support Center, I was told these two script files have not been modified in All-In-1 version 2.1. Don't forget to re-compile the TXL so these changes become part of your production All-In-1 system. As I find more workarounds, I'll pass them on; until then

The logic of the script file WPMAMP.SCP was not correct. The procedure assumed that the document appearing in the Current Item Block would not be deleted. Therefore, when it was deleted, the error was produced because the folder name of the document had been changed to WASTEBASKET. New code is bolded and modified code is underlined the excerpt below of WPMAMP.SCP.

```
.LABEL TOP
GET OA$FUNCTION="NEXT_LIST " #LIST
.IF $NEXT_LIST_ITEM EQS "" THEN .GOTO EXIT_FINISHED
GET #MDMPPDOC=$NEXT_LIST_ITEM
.IF #ORIG_DOC EQS #MDMPPDOC THEN GET #FIRST_DOC = #MDMPPDOC
.IF #PROMPT EQS "NOPROMPT" THEN .GOTO DO_FUNCTION

.LABEL EXIT_FINISHED
.IF #MULTOP NES "DELETE" THEN .GOTO ALL_DONE
DELETE MDLIST.SEL
.IF #FIRST_DOC EQS #ORIG_DOC THEN .GOTO SKIP_CURRENT

.LABEL CURRENT_OKAY
GET @#CURDOC=#ORIG_DOC
PROMPT #ORIG_DOC "for cab current"
```

```
CAB CURRENT @#CURDOC
.IF OA$STATUS == 1 THEN .GOTO ALL_DONE
PROMPT #ORIG_DOC " for next_doc"
```

```
.LABEL SKIP_CURRENT
CAB NEXT_DOCUMENT #ORIG_DOC:30,@#CURDOC
.IF OA$STATUS == 1 THEN .GOTO ALL_DONE
CAB NEXT_FOLDER #ORIG_DOC:30,@#CURDOC
.IF OA$STATUS == 1 THEN .GOTO ALL_DONE
CAB SELECT ,,,@#CURDOC
```

The script file WPSAMP.SCP had a typographical error in an .IF statement. The modified code is underlined in the excerpt below of WPSAMP.SCP.

```
.IF OA$FORM_DISPOSE EQ 0 THEN .GOTO QUIT_SEND
MAIL ATTACH $SEND
.IF OA$STATUS == 0 THEN .GOTO QUIT_SEND
```

12 Bit News

Yes, the 12 Bit world is still alive and it still has a voice here in the Office Automation Newsletter even though we have not been able to publish anything for the last several issues for one reason and another. Recently there have been a number of developments of interest to the 12 bit user community so it is fortunate that the problems have been resolved and we are back in business.

For those of you who wonder what 12 Bit means, it is a reference to what was called the 12 Bit Special Interest Group for many years. The 12 Bit SIG covered everything to do with the PDP-8 and PDP-12 computer families which of course are systems based on a 12 bit word size (as compared with the 16 bit PDP-11s for example).

Besides the hardware, one of the most popular areas for the 12 bit community has always been the OS/8 family of operating systems and the software that runs under them. OS/78 and now OS/278 are the newer members of the OS/8 family and many users still run them on their DECstation 78s and DECmate I, II, and III. In fact, OS/278 is available at very low cost from the DECUS program Library to run on the DECmate II and III and it can be very interesting for getting work done on those systems that WPS-8 can not handle.

OS/278 Sources Available

Recently DEC released the source kit for OS/278 to the DECUS library. This is quite a break through. For the first time a DEC operating system is fully in the public domain and accessible to users for modification and enhancement.

A number of users have already expressed interest in this development and have started looking at the source kit to

see what they can do with it. Among the projects is retrofitting OS/278 back to DECmate I systems and maybe even further back to DECstation 78s and the various PDP-8 models. This would help satisfy a need I hear from users regularly that they have trouble getting any operating system software for their 12 bit systems.

12 Bit Software Project

Time and space do not allow inclusion this month of the details but our long time contributor Wally Kalinowski has been keeping me up to date on his progress on his 12 Bit Software project. He has raised money and used it to get the right to submit some of the best proprietary software for the 12 bit family to the DECUS Program Library.

Wally has been working on this a long time and he has finally succeeded. The VISTA full screen editor is one of the better know programs among the ones he has been working on for example. More details about this in future issues.

Wally has also been forwarding material that has been coming his way from other 12 bit users in his part of the world. For example, he sent a patch to fix a long standing problem in OS/8 LIBRA.SV which we will try to publish soon.

As always, send you 12 bit questions and contributions to:

Bob Hassinger
Liberty Mutual Research Center
71 Frankland Road
Hopkinton, MA 01748
617-435-9061

WPS Notes

WPS-PLUS/VMS Articles

The last two months we have carried the Printer Table Utility (PTU) article provided to us by Jack Gilmore and John Connolly at DEC. This month we have the External Applications Applications Link (XAL) article from Jack Gilmore.

The PC version of WPS-PLUS has documentation on the version of PTU that comes with it but the VAX version of PTU is not documented elsewhere. The article on XAL contains an extended version of information that is supplied in the current release of WPS-PLUS/VMS. It documents a VMS specific capability to do things like pulling information out of an application such as a spread sheet program and insert it into a document you are editing.

Both of these capabilities are very useful extensions of WPS-PLUS. I found it was very helpful to spend some time looking them over so I knew what was available when I started getting requests to do things like interfacing WPS to unsupported printers and linking in information from sources outside of WPS-PLUS.

We would like to hear from you if you have tried using either one of these facilities. What have you tried? How did you make out? What problems did you have? Have you developed a printer table for a particular printer? Have you done something useful with the XAL? Both these facilities lend themselves to sharing your work with other users because the resulting tables and procedures are usually short enough to print here in the Newsletter.

If you have something to share, send it to me or give me call to talk about ways to get it to me in machine readable form.

Blue Sky Department

The basic capabilities of the XAL facility are very helpful in many cases but incurable system hackers might be tempted to look for additional ways to do more with WPS-PLUS/VMS.

For example, consider the GOLD \$ command which a lot of users have never noticed. It works both while you are in the WPS editor and when you are outside the editor in the menus. It lets you issue DCL commands while you are still in WPS. It is good for doing some limited things outside of a WPS session without having to EXIT and then restart WPS-PLUS afterward but unfortunately it seems to work by doing a subprocess SPAWN of the type that does not propagate the DCL symbols, process logical names and so on from your current process. If a WPS user has symbols and logical names as part of his working environment this can be a problem. Also, if you have to spawn a new subprocess each time you do GOLD \$ there is a fair amount of time and system overhead used in repeated process creations.

There is a way around this type of problem when you are using other software such as the new TPU editor. You can arrange to have TPU run in a subprocess and by doing an ATTACH from inside of TPU you can jump right out and back to your main process which of course still has your complete working environment including all your DCL symbols, logical names and so on.

At that point you can then do an ATTACH back into TPU in the subprocess whenever you want. The nice part is that the jumping back and forth only takes a second or two and you are still exactly where you were before. Since TPU takes a fairly long time to initialize, come up and start to

edit a file, this can be very nice if you want to get in and out of the editor frequently.

In the TPU documentation this idea was referred to as a "kept process". It turns out that for some software such as TPU all this can be very easily reduced to single key strokes to start up the the subprocess with TPU in it, reenter it after it has been created and jump out of it, back to DCL. I work this way all the time with kept subprocesses for TPU, Datatrieve and VMS MAIL. All it takes are a few very small DCL procedures. It seems as though this same kind of thing would be a very nice feature for some WPS users, much better than what you can do with the XAL or GOLD \$. The only problem is WPS-PLUS does not seem to provide the ATTACH capability that is required to make it work.

The key to this sort of thing, as well as many others such as adding your own functions to the WPS menus or modifying the way existing functions work or taking over something like the DECpage interface to use with another type setting program such as TeX is being able to get hold of some "hooks" into the software. The XAL and the printer tables are the only documented hooks so this is not very easy right now.

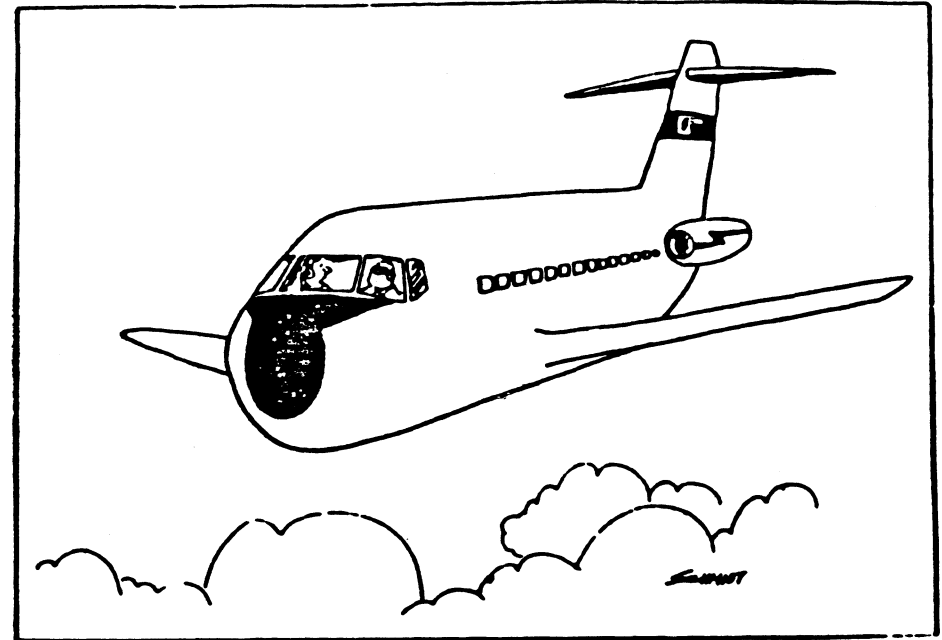
Because of this, it takes an adventure-some system programmer with a willingness to risk getting committed to software features DEC does not support. DEC has no responsibility to maintain undocumented hooks you might find and use like this so your modifications may well not work with future versions of WPS-PLUS and DEC may not be able to support WPS for you when you have made changes like this to it. All the same, it is fun to think about and investigate and it does present possibilities for meeting needs that DEC does not provide for and situations where DEC's solu-

tion is too expensive.

For example, people who are familiar with programming All-in-One based applications will find that the basis of WPS-PLUS/VMS looks very familiar. I seems to use the same basic system for menus, flow control and communications. Some of the clues you need to look into this area are in the WPS-PLUS installation procedure that is used by VMSINSTAL. Has anyone investigated any of the possibilities this opens up for people who have WPS-PLUS/VMS without All-in-One?

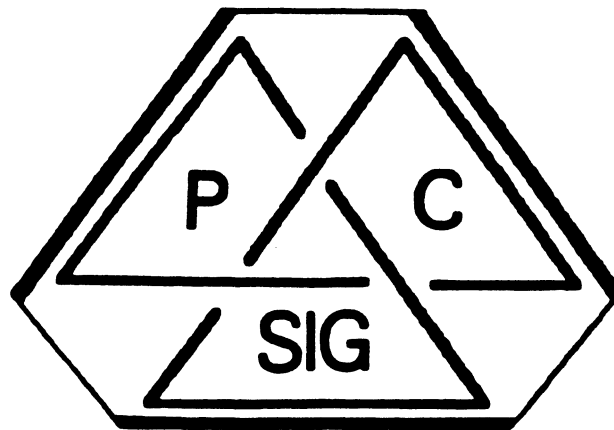
Right now there does not seem to be any other forum for users to explore this sort of thing so if there is interest we will try to cover it here. Write or call me if you have input, ideas, or questions.

Bob Hassinger
Liberty Mutual Research Center
71 Frankland Road
Hopkinton, MA 01748
617-435-9061



**"The tower Says the Computer Shows We've
Already Landed—So We Can't Do It Again..."**

Personal Computer Special Interest Group



Newsletter

Personal Computer SIG Steering Committee

SIG CHAIRMAN

Barbara Maaskant
UT Health Science Center
7703 Floyd Curl Drive
San Antonio, TX 78284
(512) 691-7351

PRO WORKING GROUP CHAIRMAN

Thomas R. Hintz
University of Florida
IFAS Computer Network
1022 McCarty Hall
Gainseville, FL 32611
(904) 392-5181

DECmate WORKING GROUP CHAIRMAN

Cheryl Johnson
Grinnell College
P.O. Box 805
Grinnell, IA 50112-0810
(515) 236-2570

RAINBOW WORKING GROUP CHAIRMAN

Lynn Jarrett
Union Tribune Publishing
P.O. Box 191
San Diego, CA 92108
(619) 299-3131 x1130

VAXmate WORKING GROUP CHAIRMAN

Frederick G. Howard
Eastman Kodak Company
901 Elmgrove Road, D345-LP
Rochester, NY 14650
(716) 724-5331

LIBRARY COMMITTEE REP./ LIBRARIAN

Ron S. Hafner
Hafner and Associates
P.O. Box 2924
Livermore, CA 94550
(415) 449-4178

COMMUNICATIONS COMMITTEE REP./NEWSLETTER EDITOR

Kenneth LeFebvre
Sytek, Inc.
19 Church St.
P.O. Box 128
Berea, OH 44017
(216) 243-1613

PROFESSIONAL CONTRIBUTING EDITOR

Gary Rice
McDonnell Douglas
K34-C636-4W
5701 Katella Avenue
Cypress, CA 90630

PRE-SYMPOSIA SESSION COORDINATOR

Vince Perriello
Crosfield Composition
Systems
570 Taxter Road
Elmsford, NY 10523
(914) 592-3600

SYMPOSIA COORDINATOR

Rick Eliopoulos
5258 Vickie Drive
San Diego, CA 92109
(619) 225-7867

SESSION NOTES EDITOR

Alan Bruns
Allied Electronics
401 E. 8th Street
Fort Worth, TX 76102
(817) 336-5401

CAMPGROUND COORDINATOR

Jim Wilson
Ntl Tech Inst for the Deaf
Rochester Inst of Tech
P.O. Box 9887
Rochester, NY 14623
(716) 475-6241

PROFESSIONAL LIBRARIAN

Peter Flack
Computer Sciences Corp.
P.O. Box 12233
Research Triangle Park,
NC 27709
(919) 541-4669

MEMBERS-AT-LARGE

Michael Bowers
University of California
Animal Science Department
Davis, CA 95616
(916) 752-6136

Theodore Needleman
Hardcopy Magazine
Seldin Publishing, Inc.
1061 S. Melrose, Suite D
Placentia, CA 92670

Russ Wertenberg
Sandia National Labs
Division 8352
Livermore, CA 94550
(415) 422-2663

DIGITAL COUNTERPARTS

PRO

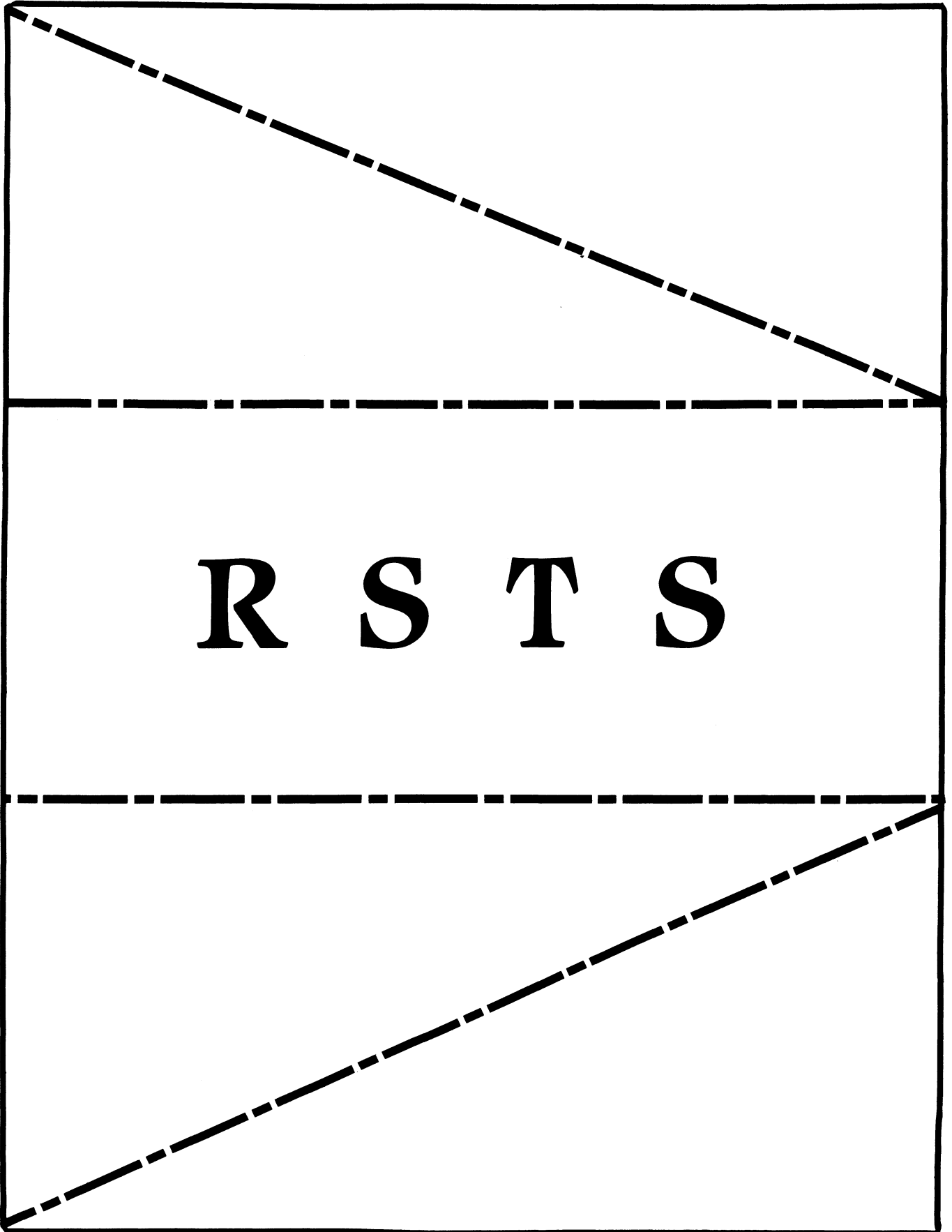
Lin Olsen
Digital Equipment
Corporation
30 Porter Road (LJ02/I3)
Littleton, MA 01460

DECmate

Ron Gemma
Digital Equipment
Corporation

Rainbow

Katrina Holman
Digital Equipment
Corporation
30 Porter Road (LJ02/I3)
Littleton, MA 01460



RST

RSTS

Chairman

Charles Mustain
Stark County School System
Louisville, OH

Symposium Coordinator

Scott W. Pandorf
Kittle's Home Furnishings
Indianapolis, IN

Assistant Symposium Coordinator

Wef Fleischman
Software Techniques
Cypress, CA

Newsletter Editor

Open

Library Representative

Susan Abercrombie
Ventrex Laboratories Inc.
Portland, ME

DEC Counterpart

Kathy Waldron
Digital Equipment Corporation
Merrimack, NH

Pre-Symposium Seminar Coordinator

Bruce Gaarder
Macalester College
St. Paul, MN

Wish Lists Coordinator

Neal E. Goldsmith
Software Techniques, Inc.
Cypress, CA

Vice SIG Chairman

Wish Lists & Tape Copy Coordinator

Philip Hunt
System Industries
Milpitas, CA

EDUSIG Liaison

George Wyncott
Purdue University Computing Center
W. Lafayette, IN

RSTS Product Planning Coordinator

Errol E. Ethier
Information Design and Management, Inc.
Shrewsbury, MA

Members At Large

Ed Beadel
Instructional Computer Center
Oswego, NY

Scott Daily
Great Lakes Chemical Corp.
W. Lafayette, IN

Mark Gilmore
Cal State University
Long Beach, CA

Mark Hartman
Jadtec Computer Group
Orange, CA

Jeff Killeen
Information Design & Management
Hopedale, MA

Newton J. Munson
Rochester Institute of Technology
Rochester, NY

RSX

MULTI-TASKER



RSX

Chairman

Dan Eisner
Perkin-Elmer Corp.
Garden Grove, CA

Symposium Coordinator

Rick Sharpe
Toledo Edison
Toledo, OH

Pre-Symposium Seminar Coordinator

Hans Jung
Associated Press
New York, NY

Communications Committee Representative

Allen Bennett
Lear Siegler Rapistan
Grand Rapids, MI

Newsletter Editor

Bruce Mitchell
Machine Intelligence &
Industry Magic
Byron MN

Store Coordinator

Jim Hopp
Carlton Financial Computation
South Bend IN

Session Note Editor

Burt Janz
Northern Telecom Inc.
Concord, NH

Librarian

Glenn Everhart
Mt. Holly, NJ

Campground Coordinator

Jerry Ethington
Prolifix Inc.
Frankfort, KY

DEC Counterparts

Lin Olsen
Digital Equipment Corporation
Nashua, NH

Dick Day
Digital Equipment Corporation
Nashua, NH

Working Group Coordinator

Sharon Johnson
Epidemiology
Minneapolis, MN

Working Group Chair

Evan Kudlajev
Philadelphia Electric Co.
Philadelphia, PA

RSX Group Chair Software Clinic Coord.

Roy S. Maull
U.S. Air Force
Offutt AFB, NE

Software Clinic Coordinator

Bruce Zielinski
RCA
Moorestown, NJ

Volunteer Coordinator

Gary Maxwell
U.S. Geological Survey
Menlo Park, CA

Multi-Processors Working Group Coordinator

Bruce Mitchell
Machine Intelligence & Indus. Magic
Hudson, WI

SRD Working Group Coordinator

Bob Turkelson
Goddard Space Flight Center
Greenbelt, MD

Accounting & Performance Working Group Coord.

Denny Walthers
American McGaw
Irvine, CA

Menu Coordinator

Ed Cetron
Center for Biomedical Design
Salt Lake City, UT

Members-At-Large

Jim McGlinchey
Warrenton, PA

Jim Neeland
Hughes Research Labs.
Malibu, CA

Anthony E. Scandora, Jr.
Argonne National Laboratory
Argonne, IL

Ralph Stamerjohn
Creve Coeur, MO

The Editor's Corner

Bruce R. Mitchell

By the time you read this, the Fall 1986 Symposium will be over. This issue is being submitted to print in late September, so the first reports you'll be reading on the Fall Symposium will be appearing in the December issue.

This issue marks the launching of a new, ongoing education project for the RSX SIG. Over the next year, as space permits, the Multi-Tasker will reprint some of the 'landmark' RSX technical papers and articles from the past. This series will include the original ACM paper on the RSX operating system by Dave Cutler, the exposition on SAV by Paul Bezeredi, the topological walk to an ODL by John Covert, the mathematics of RSX-11 by Ralph Stamerjohn, and other fundamental RSX articles which are now difficult to find. This month's article is the Files-11 disk specification.

There are also new articles for your enjoyment and edification. There are patches to RMD to enhance the functionality and correct the worst-case POOL statistics loss. There is an article dealing with the eternal problem of adding devices already present in a system without doing a complete SYSgen. There is a short article dealing with free software (although I rather tend to think it deals with system security). And there's a reminder in the Bag of Tricks about a PDP-11 architecture feature we sometimes tend to forget.

The editor wishes to thank each and every respondent to the call for more articles. Keep those articles coming; I've got a little something for all submitters. And don't forget to send in your entries for the puzzle contest from last month's issue. They're beginning to roll on in now, but not as many as I expected, so keep them coming.

And now, ladies and gentlemen, here's this month's dollop of food for thought.

----- Free PCs - Get 'Em While They Last -----

No doubt many of you wonder how the Multi-Tasker is published each month. Currently, it goes something like this:

- Editor accepts manuscripts up to 3rd week of the month
- Editor types manuscripts in to big RUNOFF file
- Shortly before deadline, editor makes frantic final edits
- Editor runs out final copy on letter-quality printer
- Editor (often Express) mails in final copy to DECUS HQ
- Editor breathes huge sigh of relief
- DECUS HQ adds article headers and page numbers
- DECUS HQ sends final copy to printers

And that's how the Multi-Tasker gets published every month. The Communications Committee, however, has a new and different idea. The idea goes something like this:

- DECUS buys each newsletter editor a PC,
a printer,
a 2400 baud modem,
and pays for a phone line.
- Contributors call the PC and KERMIT / EDT / DECnet in articles
- Editors prepare newsletters in a standard format using PCs
- Editors dial up a DECUS VAX and KERMIT / DECnet in newsletters

Sounds good, no? No doubt it sounds very good in some quarters. Editors would have a Rainbow or Professional to play with when not sending out the newsletters. And the cost is not actually very high; if the equipment costs \$2,500 per editor, over 23 newsletters the cost is \$57,500. Plus phone line installation and ongoing expenses, of course.

But from whence cometh this money? Newsletter subscription profits? Probably not. For argument, let's say it comes out of symposia fees. That's the biggest moneymaker. Attendance is in the 6,000 range, times 2 per year. That means attendees pay an extra \$5 for this project.

If true ... and this is only one project ... ever wonder why symposium registration fees keep going up? Five bucks here, five bucks there ... it does add up.

Now, with DECUS purportedly cutting expenses all over, some thought leads to a few interesting questions.

- Is standardization of newsletter formats necessary?
- Is standardization of newsletter formats even desirable?
- Will newsletter subscriptions increase as a result?
- Will this reduce paid manpower which gets the newsletters out?
- Has this project been reviewed by a disinterested member panel?*
- Would this money be better spent elsewhere?

Little perks for volunteers are fine. Volunteers make DECUS run. But is this project necessary? Or desirable? *This* editor doesn't think so.

Maybe you, the members-at-large, don't think so either. If

not, you can do something about it. Write to Bob Curley. He's on the Board of Directors. He's spearheading a project to improve communications between the membership and the leadership. He's an end user. He'll listen.

Robert F. Curley
Department of Radiation Therapy
University of Pennsylvania
Room 410
133 South 36th Street
Philadelphia, PA
19104-3246

Write now. Don't put it off or it will never get done. Tell him that you support this project or don't support it. Tell him that you like the way DECUS is going or that you don't like it. Tell him that you get your money's worth from symposia. Tell him that you don't attend symposia because they're too expensive. Tell him that you don't care. But do tell him something.

----- Submitting Articles to the Multi-Tasker -----

Please do submit machine readable media when it's possible. RX01, RX02 or RX50 diskette, or 800/1600 BPI 9 channel magtape are best. Any format is acceptable except ROLLIN, PRESRV or VMS backup. ANSI, BRU and DOS FLX formats are well-liked by the Editor's tape drive.

Submissions which aren't machine readable take longer to get into print. The editor is lazy and types mass quantities only once a month when progress reports are due.

If you preformat a submission in RUNOFF format, please set left margin 10, right margin 75, and when changing margins use incremental changes rather than absolute. The editor blesses you for the consideration.

Send all submissions to:

Bruce R. Mitchell
Machine Intelligence and Industrial Magic
PO Box 816
Byron, MN 55920

----- Answer to Last Month's Quiz -----

In the RSX Brain Teaser. But you'll have to ferret it out yourself.

----- And That's The Way Things Are -----

... this month in Pool Lowbegone, where all the Digital sales reps sell strong, all the IBM sales reps are good-looking, and the RSX SIG is above average.

The Bag of Tricks: MACRO-11

Bruce R. Mitchell
Machine Intelligence and Industrial Magic
PO Box 816
Byron, MN 55920

From time to time, assembly language programmers tend to find themselves in situations where they are climbing the walls in search of a program fault. This is sometimes due to an insufficient knowledge of PDP-11 architecture, in which the programmer *expects* the machine to do something it is *not* supposed to do.

There is one peculiarity in the PDP-11 architecture that has probably been encountered accidentally by more programmers than any other, and we're going to take a look at it this month.

Examine the following segment of code, which the programmer intended to read all the logical blocks on a disk, one at a time:

```
CLR      READPB+Q.IOPL+6    ; Clear high word of LBN 2word
CLR      READPB+Q.IOPL+8    ; Set up for disk LBN 0
10$:    DIR$    #READPB      ; Read the next logical block
        TSTB    IOSTAT      ; Did the read succeed?
        BMI     20$         ; If not, go process the error
... processing of disk block information ...
INC      READPB+Q.IOPL+8    ; Bump the LBN doubleword by 1
ADC      READPB+Q.IOPL+6    ; Add carry, for real big files
BR       10$              ; And go look at the next block
```

When this code is tested, it runs fine on all disks the programmer cares to try. Six months later, however, a site licensed for the program containing this code complains that the program never finishes.

"Those idiots are doing something wrong," thinks the programmer as he flies out to check out the problem. "If users would just read the documentation! Oh well, nothing like a few more frequent flyer miles."

On site, the programmer checks out the task with various disks. The task runs fine. "There's something wrong with your data!", he chortles gleefully, and leaves. As he goes out the door, the site's VP-Engineering is already having a nice chat with his Legal Department.

Let us gloss over a short but painful episode in the programmer's career. The upshot of it is that he returns on the next plane.

Back on site, he discovers that the task runs fine on small disks, but appears to loop forever when processing large disks. The symptom is that the program never ends up at 20\$ with error code IE.BLK, which is expected when attempting to read past end-of-disk. It loops forever on big disks. But little disks work just fine.

Why does this problem occur? If you already know the answer, you can skip on to the next article. If you don't, this may be an eye-opener and cause you to rewrite some of your own code.

The programmer expected the INC instruction to set the carry bit at block number 177777 (65,535) when BLKADD+2 rolled over to 0. The carry bit would then be added to the high word of BLKADD, making the doubleword (1,0) or 65,536. This is a common expectation; it is, however, incorrect. The PDP-11 INC and DEC instructions *do not* set the carry bit.

It is now evident why the program runs on "small" disks. It works correctly for any disk smaller than 65,536 blocks. On larger disks, it cycles through blocks 0 to 65,535 repeatedly.

The fix? Replace the INC instruction with ADD #1, <dd> in cases where it is important that the carry bit be set correctly.

RMDEMO Enhancements

John L. Newcomb
Hallmark Cards, Inc.
25 Bacon Road
Enfield, CT 06082

The time has come. I have procrastinated for as long as I can. The guilt is too much to bear. After many years of benefiting from DECUS supplied software, I am ready to offer a token of our appreciation: Enhancements for RMD.

RMD has two display pages (I and M) that contain information about mass storage devices. By default, RMD chooses these devices by following the DCB chain, although at build time the user may specify any or all devices via global patches in the RMD build file.

At runtime, whenever you desire RMD to display information on a device that is not one of the default devices, you must change the designation with the DEVICE= command. This must be done each time RMD is invoked. I'm sure this is as annoying to you as it is to me. Changing the default devices by rebuilding RMD each time is not a viable alternative. The larger the number of devices SYSgenned into each system just aggravates the problem.

The changes I have made to RMD offer a solution to this problem. I have tested the modifications on RSX-11M+, versions 2.1D and 3.0B. Because I have used the DEC supplied GIN\$ directive, these changes should work on any RSX that has the GIN\$ directive available.

After applying the SLP files listed below and rebuilding, RMD displays information on any mass storage device that has a local or global assignment to the mnemonics DV0: thru DV5: . The mnemonics DVx: correspond to the RMD DEVICE= command. A local assignment for a terminal running RMD overrides a global assignment. We make global assignments in our system startup file and then users may make their own local assignments.

For any RMD 'DEVICE' that does not have a corresponding DV: assignment, RMD displays the information for the default device. This is also the case for 'invalid' assignments such as 'ASN TT4:=DV2:'. I am not aware of any device conflict in using the DV: mnemonic. If there is a conflict, any mnemonic may be substituted in the SLP file for the RMDXCM.MAC module and then used in ASN statements.

Users can now invoke RMD without having to change device designations each time.

There have been several inquiries published requesting a fix to RMD to retain the 'worst case POOL statistics' when switching pages. I plead guilty to having the fix (as simple as it is) for a long time (say since M V3.2 days). DEC has also come up with a 'fix' for version 3.0, so if you have the problem and are running any system from M V3.2 to M+ V2.1, inclusive, do the following:

In the module MDCOM.MAC, delete the following lines:

```
$MDWPL::WORD 0 ; WORST CASE LARGEST POOL FRAGMENT
$MDWPT::WORD 0 ; SMALLEST POOL HAS GONE TO
$MDWPF::WORD 0 ; MOST NUMBER OF FRAGMENTS SO FAR
```

In the module RMDXCM.MAC, insert the above lines deleted from the module MDCOM.MAC. Then pick-up with step 2 below.

Do the following steps for the source modules you have changed. The source files are in [14,10].

1. Apply the following SLP patches to the appropriate source.
2. Assemble the module. The Macro command line may be obtained by examining the file [14,24]RMDASM.CMD. If you use the command lines from the file don't forget to make the assignments for IN:, OU:, and LI:
3. Replace the object files in [1,24]RMD.OLB using the commands:

```
SET /UIC=[1,24]
LBR RMD.OLB/-EP/RP=[14,10]module name,module name,etc
```
4. Rebuild RMD using 'TKB @RMDBLD'
5. VMR REMove the old task and VMR INSTall the new task if necessary.
6. REMove the old task and INSTall the new task in the running system if necessary.

Here are the SLP patches to be applied to RMD source file [14,10]RMDdemo.MAC:

```
RMDdemo.MAC/AU=RMDdemo.MAC
-/SCPAD/,/,/;JLN/
GL.BUF: .BLKW 6 ; GLUN$ BUFFER
GN.BUF: .BLKW 6 ; GIN$ BUFFER
-/.MCALL/,/,/;JLN/
```

```
.MCALL GLUN$$,GIN$$
-/40$:/,/,/;JLN/
GLUN$$ #DLUN, #GL.BUF ; GLUN$ on TI:
BCS EXJ ; Skip, GLUN$ failed
; This loop handles global assignments of type DVnn:
CLR R0 ; Counts DV units
CLR R1 ; RMD table offset
MOV #6, R2 ; 6 units max
LP: GIN$$ #GI.GAS, #GN.BUF, #6, #"DV, R0, #0, #0
BCS 100$ ; Skip, GIN$ failed
BIT #DV.DIR, GN.BUF+4 ; Directory device?
BEQ 100$ ; If not, don't do it
MOV GN.BUF, $MDDEV(R1) ; Load device name
MOV GN.BUF+2, $MDDEV+2(R1) ; Load unit number
100$: INC R0 ; Do next DV unit
ADD #6, R1 ; Point at next entry
SOB R2, LP ; And loop until done
; This loop handles local assignments of type DVnn:
CLR R0 ; Counts DV units
CLR R1 ; RMD table offset
MOV GN.BUF+G.LUNU, R3 ; Load TI: unit #
MOV #6, R2 ; 6 units max
LP2: GIN$$ #GI.GAS, #GN.BUF, #6, #"DV, R0, #"TT, R3
BCS 200$ ; Skip, GIN$ failed
BIT #DV.DIR, GN.BUF+4 ; Directory device?
BEQ 200$ ; If not, don't do it
MOV GN.BUF, $MDDEV(R1) ; Load device name
MOV GN.BUF+2, $MDDEV+2(R1) ; Load unit number
200$: INC R0 ; Do next DV unit
ADD #6, R1 ; Point at next entry
SOB R2, LP2 ; And loop until done
EXJ:
/
```

And here are the SLP patches to be applied to RMD source file [14,10]RMDXCM.MAC:

```
RMDXCM.MAC/AU=RMDXCM.MAC
```

```

-/$MDDEV:/.+12,/;JLN/
$MDDEV:..ASCII /DV/
      .WORD 0 ; Device name
      .WORD 0 ; Unit number
      .ASCII /DV/ ; Device UCB address
      .WORD 1, 0
      .ASCII /DV/
      .WORD 2, 0
      .ASCII /DV/
      .WORD 3, 0
      .ASCII /DV/
      .WORD 4, 0
      .ASCII /DV/
      .WORD 5, 0

```

The patches described by John won't work on RSX-11S/M; GIN\$ isn't available. However, for S/M, the same effect is available via ALUN\$, then GIUN\$ to DVnn: on a fake LUN; much the same information is returned. Coding is left as an exercise for the reader. --- The Editor

Rebuilding Device Drivers

Gerald Burgess
 McDonnell Douglas
 PO Box 2637
 Garden Grove, CA 92642-2637

Here is a nifty little command file I have found quite useful when installing foreign devices that emulate DEC devices.

Recently I was adding an SI 9761 disk drive to a PDP-11/70 which already supported DEC RM03s. Both devices require the use of the DR device driver. One of the RM03s is my boot device and thus DRDRV is loaded at VMR time. I was therefore looking at yet another SYSGEN.

Instead, I performed section H of SYSGEN, wrote and ran this command file. The new device driver is loaded and brought online during system startup. The system is now up and running with a DEC RM03 system disk and an SI 9761 data disk.

In the event you need to add a DEC emulated device like I did, here is a sample execution of "RENAMEDRV.CMD" :

```

>@RENAMEDRV
RENAMEDRV -- Cmd file to rename a device driver
>* Enter new driver name [S: R:2-2]: QA
>* Create new driver ? [Y/N]: Y
>* Assemble new driver ? [Y/N]: Y
>* Build new driver ? [Y/N]: Y
>* Enter old driver name [S: R:2-2]: DR
>* Enter old driver code filename [S: R:5-5 D:"DRDRV"]: DR
>* Enter old driver database filename [S: R:5-5 D:"DRTAB"]: DR
>* Enter new driver code filename [S: R:5-5 D:"QADRV"]: QA
>* Enter new driver database filename [S: R:5-5 D:"QADRV"]: QA

RENAMEDRV -- Creating QATAB.MAC
DR1:[11,10]QATAB.MAC;1 336 lines
[EOB]
RENAMEDRV -- Creating QADRV.MAC
DR1:[11,10]QADRV.MAC;1 1313 lines
[EOB]
RENAMEDRV -- Making device assignments
RENAMEDRV -- Assembling driver
RENAMEDRV -- Building driver
RENAMEDRV -- STOP
@<EOF>

```

And here is the source text for the indirect command file itself:

```

.; RENAMEDRV -- Cmd file to rename a device driver
.; Gerald Burgess May 1986
.;

.ENABLE GLOBAL
.ENABLE SUBSTITUTION
.ENABLE QUIET

.OPEN T1:
.DATA RENAMEDRV -- Cmd file to rename a device driver
.CLOSE

.ASKS [2:2] NEW Enter new driver name
.ASK EDT Create new driver
.ASK ASM Assemble new driver
.ASK BLD Build new driver
.SETL ASMBLD ASM!BLD
.SETL ANY ASMBLD!EDT
.IFF ANY .GOTO DONE
.IFF EDT .IFT ASMBLD .GOTO ASMBLD

.ASKS [2:2] OLD Enter old driver name
.SETS OLDDV OLD+"DRV"
.SETS OLDTB OLD+"TAB"

```

```

.SETS NEWDV NEW+"DRV"
.SETS NEWTB NEW+"TAB"

.ASKS [5:5:OLDDV] OLDDV Enter old driver code filename
.ASKS [5:5:OLDTB] OLDTB Enter old driver database filename
.ASKS [5:5:NEWDV] NEWDV Enter new driver code filename
.ASKS [5:5:NEWTB] NEWTB Enter new driver database filename

.SETS OLDDV OLDDV+".MAC"
.SETS OLDTB OLDTB+".MAC"
.SETS NEWDV NEWDV+".MAC"
.SETS NEWTB NEWTB+".MAC"

.OPEN RENAMEDRV.EDT
.DATA S/'OLD'/<>/W/NOTYPE
.DATA S/<><>/<>DR/W/NOTYPE
.DATA S/<>IVE/DRIVE/W/NOTYPE
.DATA S/AD<>ESS/ADDRESS/W/NOTYPE
.DATA S/<>Y/DRY/W/NOTYPE
.DATA S/A<>/ADR/W/NOTYPE
.DATA S/.EN<>/.ENDR/W/NOTYPE
.DATA S/S3.<>L/S3.DRL/W/NOTYPE
.DATA S/<>/'NEW'/W/NOTYPE
.DATA EX
.CLOSE

.OPEN TI:
.DATA RENAMEDRV -- Creating 'NEWTB'
.CLOSE
EDT 'NEWTB'='OLDTB',RENAMEDRV.EDT
.OPEN TI:
.DATA RENAMEDRV -- Creating 'NEWDV'
.CLOSE
EDT 'NEWDV'='OLDDV',RENAMEDRV.EDT
.IFF ASMBLD .GOTO DONE

.ASMBLD:
.OPEN TI:
.DATA RENAMEDRV -- Making device assignments
.CLOSE

ASN SY:=LB:
.WAIT ASN
ASN SY:=IN:
.WAIT ASN
ASN SY:=OU:
.WAIT ASN
ASN SY:=LS:
.WAIT ASN

.ASM:
.IFF ASM .GOTO BLD

```

```

.OPEN TI:
.DATA RENAMEDRV -- Assembling driver
.CLOSE

SET /UIC=[11,10]
.WAIT SET
.OPEN 'NEW'MC.MAC
.DATA LD$'NEW'=0 ; ** 'NEW' DRV IS LOADABLE **
.CLOSE
PIP 'NEW'MC.MAC/AP=RSXMC.MAC
.WAIT PIP
.OPEN [200,200]'NEW'DRVASM.CMD

.DATA ;
.DATA ; 'NEW'DRVASM.CMD -- Loadable 'NEW': driver command file
.DATA ;
.DATA ; Created on '<DATE>' at '<TIME>'
.DATA ;
.DATA OU:[11,24]'NEW'DRV,LS:[11,34]'NEW'DRV/-SP=-
.DATA IN:[1,1]EXEMC/ML,[11,10]'NEW'MC/PA:1,'NEW'DRV
.DATA OU:[11,24]'NEW'TAB,LS:[11,34]'NEW'TAB/-SP=-
.DATA IN:[1,1]EXEMC/ML,[11,10]'NEW'MC/PA:1,'NEW'TAB
.CLOSE
SET /UIC=[11,24]
.WAIT SET
.IFNINS ...MAC INS $MAC
.IFNINS ...MAC INS $MACFSL
MAC @[200,200]'NEW'DRVASM

.BLD:
.IFF BLD .GOTO DONE

.OPEN TI:
.DATA RENAMEDRV -- Building driver
.CLOSE

SET /UIC=[1,54]
.WAIT SET
.OPEN [200,200]'NEW'DRVBLD.CMD

.DATA ;
.DATA ; 'NEW'DRVBLD.CMD -- Loadable 'NEW': driver command file
.DATA ;
.DATA ; Created on '<DATE>' at '<TIME>'
.DATA ;
.DATA OU:[1,54]'NEW'DRV/-MM/-HD,SY:[1,34]'NEW'DRV/SH/-SP,-
.DATA OU:[1,54]'NEW'DRV=SY:[11,24]'NEW'DRV,SY:[11,24]'NEW'TAB;-
.DATA SY:[1,54]RSX11M.STB/SS, LB:[1,1]EXELIB/LB
.DATA /
.DATA STACK=0
.DATA PAR=DRVPAR:120000:20000
.DATA /
.CLOSE
.IFNINS ...TKB INS $TKB
.IFNINS ...TKB INS $TKBFSL

```

TKB @[200,200]'NEW'DRVBLD

.DONE:
.OPEN TI:
.DATA RENAMEDRV -- STOP
.CLOSE
.EXIT

Free Software

Justin L. Hewser
Nocturnal Aviation Software
Aerial Piracy Division
Somewhere in California

This article was inexplicably left out of the last three Multi-Taskers. After a midnight visit to the Multi-Tasker offices and an investigation of the wastebasket and personal files, I was able to convince the editor that it should be printed as a public service. --- Justin

There must have been a mistake somewhere at the printers. I'm really quite sure I told them to print this, not burn and lose it. --- The Editor

The high cost of software, whether from DEC or other source, is a factor which inhibits productivity at many sites. I don't know about your site, but ours can barely keep up with the cost of keeping M-Plus and Fortran 77 on contract. We can't even afford to think about things like full A-to-Z, DECnet or Datatrieve. Far too expensive to buy, too expensive to maintain.

Nocturnal Aviation works on a shoestring budget, I should have mentioned. We're now seriously looking at our first hardware upgrade in several years ... except that we really like the RK05s, and are still not so sure about these new technology RK06 drives. Lots of mass storage isn't everything, right?

Back to the subject at hand. The other day, one of my Mongolian DECUS friends shipped me some tapes to swap for copies of old RSX SIG tapes. Never having seen "Yak Watch" tapes before, I figured I'd check them out before use. I checked them out with PIP, FLX, BRU, and a tape cracker I wrote myself.

You can imagine my surprise (*and ill-concealed glee, no*

doubt -- Ed.) when these tapes turned out to contain backups of his entire system. "Well, what a surprise!" sez I. "Oh look, here's a big mailing list, here's the DECnet source, here's Minitab, here's SORT-11, and" Anyway, you get the idea.

It wouldn't have been fair to ship the tapes back, because then he wouldn't have the SIG tapes he wanted. So I just made backups of the important things, in case he might ever lose them and need backup copies. And then I copied the SIG tapes right on top of those old system backups.

Like many other sites, we make frequent backups to tape. And while we do believe in off-site storage of backups, we don't think that our friends should have to do it for us. So when we send out old tapes for copying of SIG tapes and the like, we use a bulk eraser on all tapes before they go out the door. That way we save ourselves and our friends a lot of trouble.

Well, that's all very interesting, but not really related to Free Software. And I see I'm running out of space for this article, not to mention that there's a lot of loud pounding and yelling to "open up" at the door of the computer room, so perhaps we'll continue this discussion of Free Software some other time.

(A word to the wise should be sufficient -- Ed.)

Files-11 On Disk Structure Specification

Digital Equipment Corporation
Maynard, MA

This is the first in a series of reprints of landmark RSX articles and technical papers. This article originally appeared in the April, 1982 Multi-Tasker. It is still the only available specification of the ODS-1 file system. --- The Editor

Files-11 On-Disk Structure Specification

19-June-1975

Revised 15-June-1977

Revised 15-April-1981

Edited 20-September-1986

Copyright (c) 1975, 1977, 1981

Digital Equipment Corporation, Maynard, Mass.

The material included in this functional specification, including but not limited to instruction times and operating speeds, is for informational purposes only. All such material is subject to change without notice. Consequently, Digital Equipment Corporation makes no claim and shall not be liable for its accuracy.

This software is furnished under a license for use only on a single computer system and may be copied only with the inclusion of the above copyright notice. This software, or any other copies thereof, may not be provided or otherwise made available to any other person except for use on such system and to one who agrees to these license terms. Title to and ownership of the software shall at all times remain in Digital Equipment Corporation.

The information in this document is subject to change without notice and should not be construed as a commitment by Digital Equipment Corporation.

Digital Equipment Corporation assumes no responsibility for the use or reliability of its software on equipment which is not supplied by Digital Equipment Corporation.

1.0 Scope

This document is a specification of the on-media structure that is used by Files-11. Files-11 is a general purpose file structure which is intended to be the standard file structure for all medium to large PDP-11 systems. Small systems such as RT-11 are specifically excluded because the complexity of Files-11 imposes too great a burden on their simplicity and small size.

This document describes structure level 1 of Files-11, also referred to as ODS-1 (on-disk structure version 1). This is implemented on the RSX-11 family, (RSX-11M, RSX-11M-Plus, IAS and RSX-11D) and on VMS. This document describes the final level of functionality for ODS-1. Structure level 2 (ODS-2) is implemented on VMS and is the basis for all new disk structure enhancements.

1.1 Summary of Revisions Made to This Specification

1. Expanded file characteristics to include most ODS-2 options.
2. Corrected H.FPRO to H.DFPR.
3. Added new fields to home block for date and count of home block modifications.
4. Added single directory support description and home block field.
5. Added field in home block for pack serial number (H.PKSR).
6. Added description of modified storage control block format to support large disks.
7. Restricted maximum number of blocks supported on a volume to 1,044,480.
8. Restricted ODS-1 to one block "clusters".
9. Restricted ODS-1 to single volume structures.
10. Clarified and expanded references to operating system support and relationship to ODS-2.
11. Removed RMS-11 definitions, to be provided in separate specification common to ODS-1 and ODS-2.

2.0 Medium

Files-11 is a structure which is imposed on a medium. That medium must have certain properties, which are described in the following section. Generally speaking, block addressable storage devices such as disks and DECTape are suitable for Files-11; hence, Files-11 structured media are generically referred to as disks.

2.1 Volume

The basic medium that carries a Files-11 structure is referred to as a volume. A volume (also often referred to as a unit) is defined as an ordered set of logical blocks. A logical block is an array of 512 8-bit bytes. The logical blocks in a volume are consecutively numbered from 0 to (n-1), where the volume contains n logical blocks.

The number assigned to a logical block is called its logical block number, or LBN. Files-11 is theoretically capable of describing volumes up to 2**32 blocks in size. In practice, a volume should be at least 100 blocks in size to be useful. Current implementations of Files-11 handle volumes up to 2**24 blocks.

The logical blocks of a volume must be randomly addressable. The volume must also allow transfers of any length up to 65K bytes; consecutively numbered logical blocks are transferred until the byte count is satisfied. In other words, the volume can be viewed as a partitioned array of bytes. It must allow reads and writes of arrays of any length less than 65K bytes, provided that they start on a logical block boundary and that the length is a multiple of four bytes. When only part of a block is written, the contents of the remainder of that logical block are undefined.

2.2 Volume Sets

This section is of historical interest only. ODS-1 does not and will not support volume sets. A volume set is a collection of related units that are normally treated as one logical device in the usual operating system concept. Each unit contains its own Files-11 structure; however, files on the various units in a volume set may be referenced with a relative volume number, which uniquely determines which unit in the set the file is located on. Other sections in this specification make occasional reference to

volume sets and relative volume number where "hooks" for their implementation exist. Since volume sets have not been implemented as yet, however, no complete specification is provided here.

3.0 Files

Any data in a volume or volume set that are of any interest (i.e., all blocks not available for allocation) are contained in a file. A file is an ordered set of virtual blocks, where a virtual block is an array of 512 8-bit bytes. The virtual blocks of a file are consecutively numbered from 1 to n, where n blocks have been allocated to the file.

The number assigned to a virtual block is called (obviously) its virtual block number, or VBN. Each virtual block is mapped to a unique logical block in the volume set by Files-11. Virtual blocks may be processed in the the same manner as logical blocks. Any array of bytes less than 65K in length may be read or written, provided that the transfer starts on a virtual block boundary and that its length is a multiple of four bytes.

3.1 File ID

Each file in a volume set is uniquely identified by a File ID. A File ID is a binary value consisting of 48 bits (3 PDP-11 words). It is supplied by the file system when the file is created, and must be supplied by the user whenever he wishes to reference a particular file.

The three words of the File ID are used as follows:

Word 1 File Number

Locates the file within a particular unit of the volume set. File numbers must lie in the range 1 through 65,535. The set of file numbers on a unit is moderately (but not totally) dense; at any instant in time, a file number uniquely identifies one file within that unit.

Word 2 File Sequence Number

Identifies the current use of an individual file number on a unit. File numbers are re-used; when a file is deleted its file number becomes available for future use for some other file. Each time a file number is re-used, a different file sequence number is assigned to

distinguish the uses of that file number. The file sequence number is essential since it is perfectly legal for users to remember and attempt to use a File ID long after that file has been deleted.

Word 3 Relative Volume Number

Identifies which unit of a volume set the file is located on. Volume sets are, at present, not implemented; the only legal value for the relative volume number in any context is zero.

3.2 File Header

Each file on a Files-11 volume is described by a file header. The file header is a block that contains all the information necessary to access the file. It is not part of the file; rather, it is contained in the volume's index file. (The index file is described in section 5.1). The header block is organized into four areas, of which the first three are variable in size.

3.2.1 Header Area

The information in the header area permits the file system to verify that this block is in fact a file header and, in particular, is the header being sought by the user. It contains the file number and file sequence number of the file, as well as its ownership and protection codes. This area also contains offsets to the other areas of the file header, thus defining their size. Finally, the header area contains a user attribute area, which may be used by the user to store a limited amount of data describing the file.

3.2.2 Ident Area

The ident area of a file header contains identification and accounting data about the file. Stored here are the primary name of the file; its creation date and time; revision count, date and time; and expiration date.

3.2.3 Map Area

The map area describes the mapping of virtual blocks of the file to the logical blocks of the volume. The mapping data consists of a list of retrieval pointers. Each retrieval pointer describes a logically contiguous segment of the file. The map area also contains the linkage to the next extension segment of the file, if such exists.

3.2.4 End Checksum

The last two bytes of the file header contain a 16-bit additive checksum of the preceding 255 words of the file header. The checksum is used to help verify that the block is in fact a file header.

3.3 Extension Headers

Since the file header is of fixed size, it is inevitable that for some files the mapping information does not fit in the allocated space. A file with a large amount of mapping data is therefore represented with a chain of file headers. Each header maps a consecutive set of virtual blocks; the extension linkage in the map area links the headers together in order of ascending virtual block numbers.

Multiple headers are also needed for files that span units in a volume set. A header may only map logical blocks located on its unit; therefore, a multi-volume file is represented by headers on all units that contain portions of that file.

3.4 File Header - Detailed Description

This section describes in detail the items contained in the file header. Each item is identified by a symbol which represents the offset address of that item within its area in the file header. Any item may be located in the file header by locating the area to which it belongs, and then adding the value of its offset address. Users who concern themselves with the contents of file headers are strongly urged to use the offset symbols. The symbols may be defined in assembly language programs by calling and invoking the macro FHDOF\$, which may be found in the macro library of any system that supports Files-11. Alternatively, one may find the macro in the file F11MAC.MAC.

3.4.1 Header Area Description

The header area of the file header always starts of byte 0. It contains the basic information needed for checking the validity of accesses to the file.

3.4.1.1 H.IDOF 1 byte Ident Area Offset

This byte contains the number of 16-bit words between the start of the file header and the start of the ident area. It defines the location of the ident area and the size of the header area.

3.4.1.2 H.MPOF 1 byte Map Area Offset

This byte contains the number of 16-bit words between the start of the file header and the start of the map area. It defines the location of the map area and, together with H.IDOF, the size of the ident area.

3.4.1.3 H.FNUM 2 bytes File Number

This word contains the file number of the file.

3.4.1.4 H.FSEQ 2 bytes File Sequence Number

This word contains the file sequence number of the file.

3.4.1.5 H.FLEV 2 bytes File Structure Level

The file structure level is used to identify different versions of Files-11 as they affect the structure of the file header. This permits upward compatibility of file structures as Files-11 evolves, in that the structure level word identifies the version of Files-11 that created this particular file. This document describes version 1 of Files-11; the only legal contents for H.FLEV is 401 octal.

3.4.1.6 H.FOWN 2 bytes File Owner UIC H.PROG = H.FOWN + 0 Programmer (Member) Number H.PROJ = H.FOWN + 1 Project (Group) Number

This word contains the binary User Identification Code (UIC) of the owner of the file. The file owner is usually (but not necessarily) the creator of the file.

3.4.1.7 H.FPRO 2 bytes File Protection Code

This word controls what access all users in the system may have to the file. Accessors of a file are categorized according to the relationship between the UIC of the accessor and the UIC of the owner of the file. Each category is controlled by a four bit field in the protection word. The category of the accessor is selected as follows:

System Bits 0 - 3

The accessor is subject to system protection if the project number of the UIC under which he is running is 10 octal or less.

Owner Bits 4 -7

The accessor is subject to owner protection if the UIC under which he is running exactly matches the file owner UIC.

Group Bits 8 - 11

The accessor is subject to group protection if the project number of his UIC matches the project number of the file owner UIC.

World Bits 12 - 15

The accessor is subject to world protection if he does not fit into any of the above categories.

Four types of access intents are defined in Files-11: Read, Write, Extend and Delete. Each four bit field in the protection word is bit encoded to permit or deny any combination of the four types of access to that category of accessors. Setting a bit denies that type of access to that category. The bits are defined as follows (these values apply to a right-justified protection field):

FP.RDV Deny read access
 FP.WRV Deny write access
 FP.EXT Deny extend access
 FP.DEL Deny delete access

When a user attempts to access a file, protection checks are performed in all the categories to which he is eligible, in the order: System, Owner, Group, World. The user is granted access to the file if any of the categories to which he is eligible grants his access.

3.4.1.8 H.FCHA 2 bytes File Characteristics
 H.UCHA = H.FCHA + 0 User Controlled Characteristics
 H.SCHA = H.FCHA + 1 System Controller Characteristics

The user controlled characteristics byte contains the following flag bits:

Reserved, 1 bit

UC.NID Set if incremental dump (backup) is to be disabled for this file.

UC.WBC Set if the file is to be write-back cached; i.e., if a cache is used for the file data, data written by a user is only written back to the disk when it is removed from the cache. Clear for write-through cache operation.

UC.RCK Set if the file is to be read-checked. All read operations on the file, including reads of the file header(s), are performed with a read, read-compare to assure data integrity.

UC.WCK Set if the file is to be write-checked. All write operations on the file, including modifications of the file header(s), are performed with a write, read-compare to assure data integrity.

UC.CNB Set if the file is allocated contiguous best effort; i.e., as contiguous as possible.

UC.DLK Set if the file is deaccess-locked. This bit is used as a flag warning that the file was not properly closed and may contain inconsistent data. Access to the file is denied if this bit is set.

UC.CON Set if the file is logically contiguous; i.e., if for all virtual blocks in the file, virtual

block i maps to logical block $k+i$ for some constant k . This bit may be implicitly set or cleared by file system operations that allocate space to the file; the user may only clear it explicitly.

The system controlled characteristics byte contains the following flag bits:

Reserved, 3 bits

Reserved, Access Control List

SC.SPL Set if the file is an intermediate file for spooling.

SC.DIR Set if the file is a directory.

SC.BAD Set if there is a bad data block in the file. This bit is as yet unimplemented. It is intended for dynamic bad block handling.

SC.MDL Set if the file is marked for delete. If this bit is set, further accesses to the file are denied, and the file is physically deleted when no users are accessing it.

3.4.1.9 H.UFAT 32 bytes User Attribute Area

This area is intended for the storage of a limited quantity of "user file attributes", i.e., any data the user deems useful for processing the file that is not part of the file itself. An example of the use of the user attribute area is presented in section 6.1 (FCS File Attributes).

3.4.1.10 S.HDHD 46 bytes Size of Header Area

This symbol represents the total size of the header area containing all of the above entries.

3.4.2 Ident Area Description

The ident area of the file header begins at the word indicated by H.IDOF. It contains identification and accounting data about the file.

3.4.2.1 I.FNAM 6 bytes File Name

These three words contain the name of the file, packed three Radix-50 characters to the word. This name usually, but not necessarily, corresponds to the name of the file's primary directory entry.

3.4.2.2 I.FTYP 2 bytes File Type

This word contains the type of the file in the form of three Radix-50 characters.

3.4.2.3 I.FVER 2 bytes Version Number

This word contains the version number of the file in binary form.

3.4.2.4 I.RVNO 2 bytes Revision Number

This word contains the revision count of the file. The revision count is the number of times the file has been accessed for write.

3.4.2.5 I.RVDT 7 bytes Revision Date

The revision date is the date on which the file was last deaccessed after being accessed for write. It is stored in ASCII in the form "DDMMYY", where DD are two digits representing the day of the month, MMM are three characters representing the month, and YY are the last two digits of the year.

3.4.2.6 I.RVTI 6 bytes Revision Time

The revision time is the time of day on which the file was last deaccessed for write. It is stored in ASCII in the format "HHMMSS", where HH are the hour, MM are the minute, and SS are the second.

3.4.2.7 I.CRDT 7 bytes Creation Date

These seven bytes contain the date on which the file was created. The format is the same as that of the revision date above.

3.4.2.8 I.CRTI 6 bytes Creation Time

These six bytes contain the time of day at which the file was created. The format is the same as that of the revision time above.

3.4.2.9 I.EXDT 7 bytes Expiration Date

These seven bytes contain the date on which the file becomes eligible to be deleted. The format is the same as that of the revision and creation dates above.

3.4.2.10 1 byte Unused

This unused byte is present to round up the size of the ident area to a word boundary.

3.4.2.11 S.IDHD 46 bytes Size of Ident Area

This symbol represents the size of the ident area containing all of the above entries.

3.4.3 Map Area Description

The map area of the file header begins at the word indicated by H.MPOF. It contains the information necessary to map the virtual blocks of the file to the logical blocks of the volume.

3.4.3.1 M.ESQN 1 byte Extension Segment Number

This byte contains the value n, where this header is the (n+1)th header of the file; i.e., headers of a file are numbered sequentially starting with 0.

3.4.3.2 M.ERVN 1 byte Extension Relative Volume Number

This byte contains the relative volume number of the unit in the volume set that contains the next sequential extension header for this file. If there is no extension header, or if the extension header is located on the same unit as this header, this byte contains 0.

3.4.3.3 M.EFNU 2 bytes Extension File Number

This word contains the file number of the next sequential extension header for this file. If there is no extension header, this word contains 0.

3.4.3.4 M.EFSQ 2 bytes Extension File Sequence Number

This word contains the file sequence number of the next sequential extension header for this file. If there is no extension header, this word contains 0.

3.4.3.5 M.CTSZ 1 byte Block Count Field Size

This byte contains a count of the number of bytes used to represent the count field in the retrieval pointers in the map area. The retrieval pointer format is described in section 3.4.3.9 below.

3.4.3.6 M.LBSZ 1 byte LBN Field Size

This byte contains a count of the number of bytes used to represent the logical block number field in the retrieval pointers in the map area. The contents of M.CTSZ and M.LBSZ must add up to an even number.

3.4.3.7 M.USE 1 byte Map Words in Use

This byte contains a count of the number of words in the map area that are presently occupied by retrieval pointers.

3.4.3.8 M.MAX 1 byte Map Words Available

This byte contains the total number of words available for retrieval pointers in the map area.

3.4.3.9 M.RTRV variable Retrieval Pointers

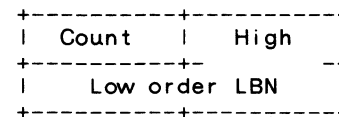
This area contains the retrieval pointers that actually map the virtual blocks of the file to the logical blocks of the volume. Each retrieval pointer describes a consecutively numbered group of logical blocks which is part of the file. The count field contains the binary value n to represent a group of (n+1) logical blocks. The logical block number field contains the logical block number of the first logical block in the group.

Thus, each retrieval pointer maps virtual blocks j through (j+n) into logical blocks k through (k+n), respectively, where j is the total number plus one of virtual blocks represented by all preceding retrieval points in this and all preceding headers of the file, n is the value contained in the count field, and k is the value contained in the logical block number field.

Although the data in the map area provided for arbitrarily extensible retrieval pointer forms, Files-11 has defined only three. Of these, only the first is currently implemented. The other two are presented out of historical interest; they will never be supported.

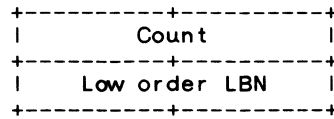
Format 1: M.CTSZ = 1
M.LBSZ = 3

The total retrieval pointer length is four bytes. Byte 1 contains the high order bits of the 24-bit LBN. Byte 2 contains the count field, and bytes 3 and 4 contain the low 16 bits of the LBN.



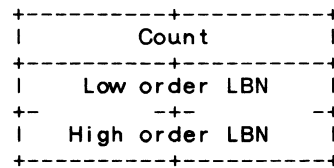
Format 2: M.CTSZ = 2
M.LBSZ = 2

The total retrieval pointer length is four bytes. The first word contains a 16-bit count field and the second word contains a 16-bit LBN field.



Format 3: M.CTSZ = 2
M.LBSZ = 4

The total retrieval pointer length is six bytes. The first word contains a 16-bit count field and the second and third words contain a 32-bit LBN field.



3.4.3.10 S.MPHD 10 bytes Size of Map Area

This symbol represents the size of the map area, not including the space used for the retrieval pointers.

3.4.4 End Checksum Description

The header check sum occupies the last two bytes of the file header. It is verified every time a header is read, and is recomputed every time a header is written.

3.4.4.1 H.CKSM 2 bytes Block Checksum

This word is a simple additive checksum of all other words in the block. It is computed by the following PDP-11 routine or its equivalent:

```

MOV  Header-address, R0
CLR  R1

```

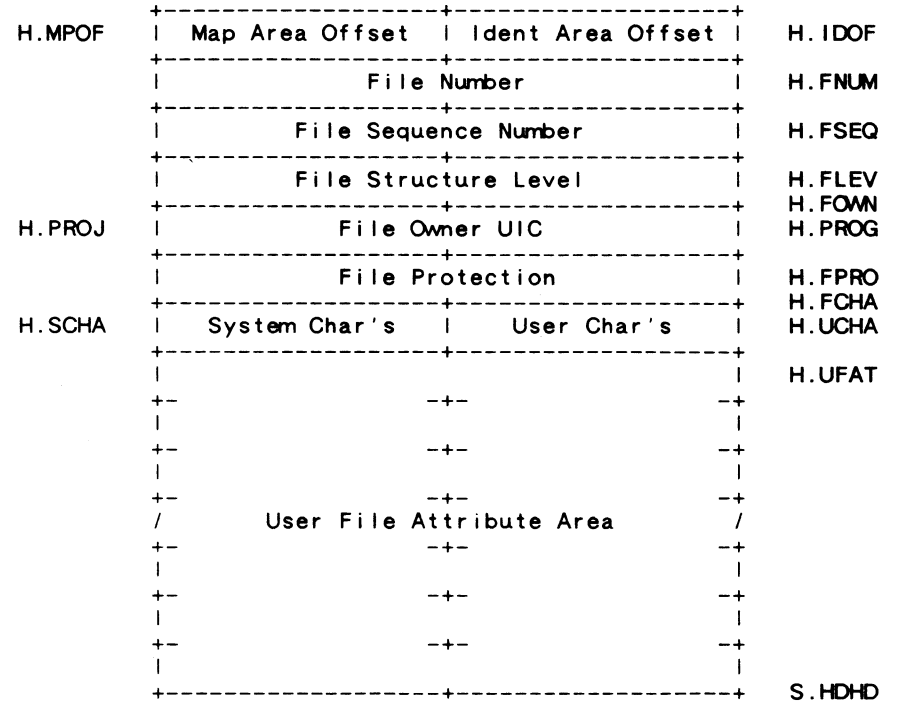
```

10$:  MOV  #255., R2
      ADD  (R0)+, R1
      SOB  R2, 10$
      MOV  R1, (R0)

```

3.4.5 File Header Layout

Header Area



Ident Area

		I.FNAM
	File Name	
	File Type	I.FTYP
	Version Number	I.FVER
	Revision Number	I.RVNO
	Revision Date	I.RVDT
I.RVTI		
	Revision Time	
I.CRDT		
	Creation Date	
	Creation Time	I.CRTI
	Revision Date	I.EXDT
	Unused	S.IDHD

Map Area

M.ERVN	Extension RVN	Extension Seq Num	M.ESQN
	Extension File Number		M.EFNU
	Extension File Sequence Number		M.EFSQ
M.LBSZ	LBN Field Size	Count Field Size	M.CTSZ
M.MAX	Map Words Avail.	Map Words in Use	M.USE
			S.MPHD
	Retrieval Pointers		M.RTRV
	File Header Checksum		H.CKSM

4.0 Directories

Files-11 provides directories to allow the organization of files in a meaningful way. While the File ID is sufficient to locate a file uniquely on a volume set, it is hardly mnemonic. Directories are files whose sole function is to associate file name strings with File IDs.

4.1 Directory Hierarchies

Since directories are files with no special attributes, directories may list files that are in turn directories. Thus, the user may construct directory hierarchies of arbitrary depth and complexity to structure his files as he pleases.

4.1.1 User File Directories

Current implementations of Files-11 all support a two level directory hierarchy which is tied in with the user identification mechanism of the operating system. Each UIC is associated with a user file directory (UFD). References to

files that do not specify a directory are generally defaulted to the user's UIC. All UFDs are listed in the volume's MFD under a file name constructed from the UIC. A UIC of [n,m] associates with a directory name of "nnnmmm.DIR;1" where nnn and mmm are n and m padded out to three digits each with leading zeroes. Note that all number conversions are done in octal.

Two points should be noted here. The UFD structure described here is not intrinsically part of the Files-11 on-disk structure; rather, it is a convenient cataloging system applied by various operating systems. Also, there is no hard and fast relationship between the owner UIC of a file and the UFD in which it is listed. Generally, they do correspond, but not necessarily.

4.2 Directory Structure

A directory is a file consisting of 16-byte records. It is structured as an FCS fixed length record file, with no carriage control attributes (see section 6 for a description of FCS files). Each record is a directory entry. The entries are not required to be ordered, or densely packed, nor do they have any other relationship to each other, except that no two entries in one directory may contain the same name, type, and version. Each entry contains the following:

- File ID** The three word binary File ID of the file that this directory entry represents. If the file number portion of the File ID field is zero, then this record is empty and may be used for a new directory entry.
- Name** The name of the file may be up to 9 characters. It is stored as three words, each containing three Radix-50 packed characters.
- Type** The type of the file (also historically referred to as the extension) may be up to three characters. It is stored as one word of Radix-50 packed characters.
- Version** The version number of the file is stored in binary in one word.

4.2.1 Directory File Entry Layout

```

+-----+-----+
|                                     |
+--          --+                     +--
|                                     |
|                                     |
|                                     |
|                                     |
+-----+-----+
|                                     |
+--          --+                     +--
|                                     |
|                                     |
|                                     |
|                                     |
+-----+-----+
|                                     |
|                                     |
|                                     |
|                                     |
+-----+-----+
|                                     |
|                                     |
|                                     |
|                                     |
+-----+-----+

```

4.3 Directory Protection

Since directories are files with no special characteristics, they may be accessed like all other files, and are subject to the same protection mechanism. However, implementations of Files-11 support three special functions for the management of directories, namely FIND, REMOVE and ENTER. A user performing such a directory operation must first have the following privileges to be allowed the various functions:

```

FIND:    Read
REMOVE:  Read, Write
ENTER:   Read, Write

```

Note that the same privilege is required for both ENTER and REMOVE. The recovery for an operation that involves a REMOVE at the beginning of the sequence is an ENTER.

5.0 Known Files

Clearly, any file system must maintain some data structure on the medium which is used to control the file organization. In Files-11 this data is kept in five files. These files are created when a new volume is initialized. They are unique in that their File IDs are known constants. These five files have the following uses:

File ID 1,1,0 is the index file. The index file is the root of the entire Files-11 structure. It contains the

volume's bootstrap block and the home block, which is used to identify the volume and locate the rest of the file structure. The index file also contains all of the file headers for the volume, and a bitmap to control the allocation of file headers.

File ID 2,2,0 is the storage bitmap file. It is used to control the allocation of logical blocks on the volume.

File ID 3,3,0 is the bad block file. It is a file containing all of the known bad blocks on the volume.

File ID 4,4,0 is the volume master file directory, or MFD. It forms the root of the volume's directory structure. The MFD lists the five known files, all first level user directories, and whatever other files the user chooses to enter.

File ID 5,5,0 is the system core image file. Its use is operating system dependent; its basic purpose is to provide a file of known File ID for the use of the operating system.

5.1 Index File

The index file is File ID 1,1,0. It is listed in the MFD as INDEXF.SYS:1. The index file is the root of the Files-11 structure in that it provides the means for identification and initial access to a Files-11 volume, and contains the access data for all files on the volume, including itself.

5.1.1 Bootstrap Block

Virtual block 1 of the index file is the volume's boot block. It is always mapped to logical block 0 of the volume. If the volume is the system device of an operating system, the boot block contains an operating system dependent program which reads the operating system into memory when the boot block is read and executed by a machine's hardware bootstrap. If the volume is not a system device, the boot block contains a small program that outputs a message on the system console to inform the operator to that effect.

5.1.2 Home Block

Virtual block 2 of the index file is the volume's home block. The logical block containing the home block is the first good block on the volume from the sequence 1, 256, 512, 768, 1024, 1280 ... (256*n). The purpose of the home block is

to identify the volume as Files-11, establish the specific identity of the volume, and serve as the ground zero entry point into the volume's file structure. The home block is recognized as a home block by the presence of checksums in known places and by the presence of predictable values in certain locations.

Items contained in the home block are identified by symbolic offsets in the same manner as items in the file header. The symbols may be defined in assembly language programs by calling and invoking the macro HMBOF\$, which may be found in the macro library of any system that supports Files-11. Alternatively, one may find the macro in the file F11MAC.MAC.

5.1.2.1 H.IBSZ 2 bytes Index File Bitmap Size

This 16-bit word contains the number of blocks that make up the index file bitmap. (The index file bitmap is discussed in section 5.1.4.) This value must be non-zero for a valid home block.

5.1.2.2 H.IBLB 4 bytes Index File Bitmap LBN

This doubleword contains the starting logical block address of the index file bitmap. Once the home block of a volume has been found, it is this value that provides access to the rest of the index file and to the volume. The LBN is stored with the high order in the first 16 bits, followed by the low order portion. This value must be non-zero for a valid home block.

5.1.2.3 H.FMAX 2 bytes Maximum Number of Files

This word contains the maximum number of files that may be present on the volume at any time. This value must be non-zero for a valid home block.

5.1.2.4 H.SBCL 2 bytes Storage Bitmap Cluster Factor

This word contains the cluster factor used in the storage bitmap file. The cluster factor is the number of blocks represented by each bit in the storage bitmap. Volume clustering is not implemented in ODS-1; the only legal value for this item is 1.

5.1.2.5 H.DVTY 2 bytes Disk Device Type

This word is an index identifying the type of disk that contains this volume. It is currently not used and always contains 0.

5.1.2.6 H.VLEV 2 bytes Volume Structure Level

This word identifies the volume's structure level. Like the file structure level, this word identifies the version of Files-11 which created this volume and permits upward compatibility of media as Files-11 evolves. The volume structure level is affected by all portions of the Files-11 structure except the contents of the file header. This document describes Files-11 version 1; the only legal values for the structure level are 401 and 402 octal. The former (401) is the standard value for most volumes. The latter (402) is an advisory that the volume contains a multiheader index file. (A multiheader index file is required to support more than about 26,000 files. The index file may in fact be multiheader without the volume having a structure level of 402.)

5.1.2.7 H.VNAM 12 bytes Volume Name

This area contains the volume label as an ASCII string. It is padded out to 12 bytes with nulls. The volume label is used to identify individual Files-11 volumes.

5.1.2.8 4 bytes Unused

5.1.2.9 H.VOWN 2 bytes Volume Owner UIC

This word contains the binary UIC of the owner of the volume. The format is the same as that of the file owner UIC stored in the file header.

5.1.2.10 H.VPRO 2 bytes Volume Protection Code

This word contains the protection code for the entire volume. Its contents are coded in the same manner as the file protection code stored in the file header, and it is interpreted in the same way in conjunction with the volume owner UIC. All operations on all files on the volume must pass both the volume and the file protection check to be permitted. (Refer to the discussion on file protection in section 3.4.1.7).

5.1.2.11 H.VCHA 2 bytes Volume Characteristics

This word contains bits which provide additional control over access to the volume. The following bits are defined:

CH.NDC Obsolete, used by RSX-11D / IAS. Set if device control functions are not permitted on this volume. Device control functions are those which can threaten the integrity of the volume, such as direct reading and writing of logical blocks, etc.

CH.NAT Obsolete, used by RSX-11D / IAS. Set if the volume may not be attached, i.e., reserved for exclusive use by one task.

CH.SDI Set if the volume contains only a single directory. If this bit is set, no directories should be created on the volume other than the MFD. The access methods should also be informed of this situation, e.g. by setting the DV.SDI bit in the UCB device characteristics word.

5.1.2.12 H.DFPR 2 bytes Default File Protection

This word contains the file protection that is assigned to all files created on this volume if no file protection is specified by the user.

5.1.2.13 6 bytes Unused

5.1.2.14 H.WISZ 1 byte Default Window Size

This word contains the number of retrieval pointers used for the "window" (in-memory file access data) when files are accessed on the volume, if not otherwise specified by the accessor.

5.1.2.15 H.FIEX 1 byte Default File Extend

This byte contains the number of blocks that are allocated to a file when a user extends the file and asks for the system default value for allocation.

5.1.2.16 H.LRUC 1 byte Directory Pre-Access Limit

This byte contains a count of the number of directories to be stored in the file system's directory access cache. More generally, it is an estimate of the number of concurrent users of the volume, and its use may be generalized in the future.

5.1.2.17 H.REVD 7 bytes Date of Last Home Block Revision

This field (ill defined field) is in the standard ASCII date format and reflects the date of the last modifications to fields in the home block.

5.1.2.18 H.REVC 2 bytes Count of Home Block Revisions

This field reflects the number of above mentioned modifications.

5.1.2.19 2 bytes Unused

5.1.2.20 H.CHK1 2 bytes First Checksum

This word is an additive checksum of all entries preceding in the home block (i.e., all those listed above). It is computed by the same sort of algorithm as the file header checksum (see section 3.4.4.1).

5.1.2.21 H.VDAT 14 bytes Volume Creation Date

This area contains the date and time that the volume was initialized. It is in the format "DDMMYYHHMMSS", followed by a single null. (The same format is used in the ident area of the file header, section 3.4.2).

5.1.2.22 382 bytes Unused

This area is reserved for the relative volume table for volume sets. This field is not used, although some versions of DSC referred to this area.

5.1.2.23 H.PKSR 4 bytes Pack Serial Number

This area contains the manufacturer supplied serial number for the physical volume. For last trace devices, the pack serial number is contained on the volume in the manufacturer data. For other devices the user must supply this information manually. The serial number is contained in the home block for convenience and consistency.

5.1.2.24 12 bytes Unused

5.1.2.25 H.INDN 12 bytes Volume Name

This area contains another copy of the ASCII volume label. It is padded out to 12 bytes with spaces. It is placed here in accordance with the volume identification standard (STD 167).

5.1.2.26 H.INDO 12 bytes Volume Owner

This area contains an ASCII expansion of the volume owner UIC in the form "[proj,prog]". Both numbers are expressed in decimal and are padded to three digits with leading zeroes. The area is padded out to 12 bytes with trailing spaces. It is placed here in accordance with the volume identification standard (STD 167).

5.1.2.27 H.INDF 12 bytes Format Type

This field contains the ASCII string "DECFILE11A" padded out to 12 bytes with spaces. It identifies the volume as being of Files-11 format. It is placed here in accordance with the volume identification standard (STD 167).

5.1.2.28 2 bytes Unused

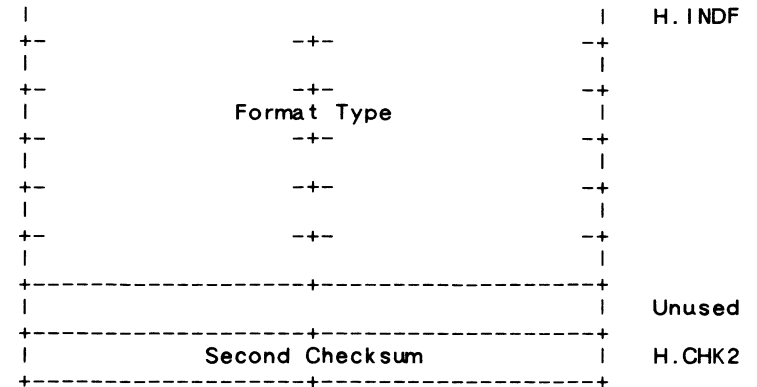
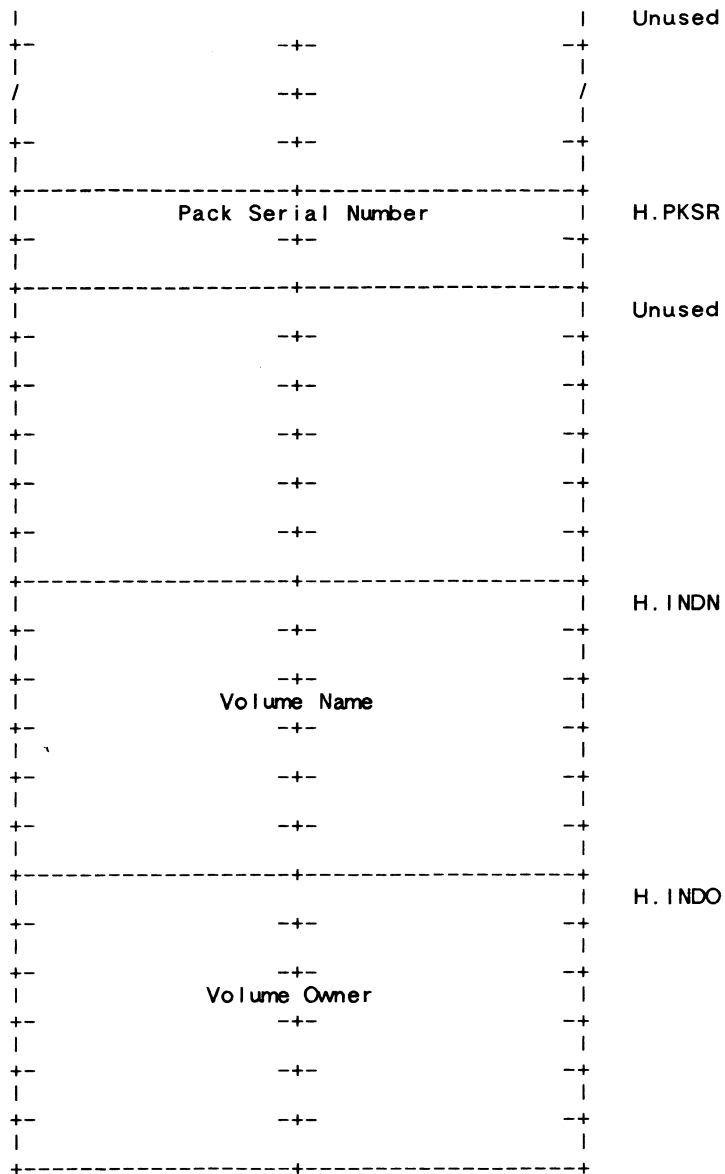
5.1.2.29 H.CHK2 2 bytes Second Checksum

This word is the last word of the home block. It contains an additive checksum of the preceding 255 words of the home block, computed according to the algorithm in section 3.4.4.1.

5.1.3 Home Block Layout

Index File Bitmap Size	H.IBSZ
Index File Bitmap LBN	H.IBLB
Maximum Number of Files	H.FMAX
Storage Bitmap Cluster Factor	H.SBCL
Disk Device Type	H.DVTY
Volume Structure Level	H.VLEV
	H.VNAM
Volume Name	

		Unused
	Volume Owner UIC	H.VOWN
	Volume Protection	H.VPRO
	Volume Characteristics	H.VCHA
	Default File Protection	H.DFPR
		Unused
H.FIEX	Def. File Extend	H.WISZ
H.REVD	Directory Limit	H.LRUC
	Volume Modification Date	
	Volume Modification Count	H.REVC
		Unused
	First Checksum	H.CHK1
		H.VDAT
	Volume Creation Date	



5.1.4 Index File Bitmap

The index file bitmap is used to control the allocation of file numbers (and hence file headers). It is simply a bit string of length n , where n is the maximum number of files permitted on the volume (contained in offset H.FMAX in the home block). The bitmap spans as many blocks as are necessary to hold it, i.e., maximum number of files divided by 4096 and rounded up. The number of blocks in the bitmap is contained in offset H.IBSZ of the home block.

The bits in the index file bitmap are numbered sequentially from 0 to $(n-1)$ in the obvious manner, i.e., from right to left in each byte, and in order of increasing byte address. Bit j is used to represent file number $(j+1)$: If the bit is 1, then that file is in use; if the bit is 0, then that file number is not in use and may be assigned to a newly created file.

The index file bitmap starts at virtual block 3 of the index file and continues through VBN $(2+m)$, where m is the number of blocks in the bitmap. It is located at the logical block indicated by offset H.IBLB in the home block.

5.1.5 File Headers

The rest of the index file contains all the file headers for the volume. The first 16 file headers (for file numbers 1 through 16) are logically contiguous with the index file bitmap to facilitate their location; the rest may be allocated wherever the file system sees fit. Thus, the first 16 file headers may be located from the data in the home block (H.IBSZ

and H.IBLB) while the rest must be located through the mapping data in the index file header. The file header for file number n is located at virtual block $(2+m+n)$, where m is the number of blocks in the index file bitmap.

5.2 Storage Bitmap File

The storage bitmap file is File ID 2,2,0. It is listed in the MFD as BITMAP.SYS;1. The storage bitmap is used to control the available space on a unit. It consists of a storage control block which contains summary information about the unit, and the bitmap itself which lists the availability of individual blocks.

5.2.1 Storage Control Block

Virtual block 1 of the storage bitmap is the storage control block. It contains summary information intended to optimize allocation of space on the unit. The storage control block has the following format for disks with less than 516,096 (4096*126) blocks:

3 bytes: Unused (zero)
1 byte: Number of storage bitmap blocks (less than 127)
2 bytes: Number of free blocks in bitmap block 1
2 bytes: Free block pointer in bitmap block 1
2 bytes: Number of free blocks in block 2
2 bytes: Free block pointer in bitmap block 2
.
.
.
2 bytes: Number of free blocks in block n
2 bytes: Free block pointer in bitmap block n
4 bytes: Size of the unit in logical blocks

The storage control block has the following format for disks with more than 516,096 (4096*126) blocks:

3 bytes: Unused (zero)
1 byte: Number of storage bitmap blocks (greater than 126)
4 bytes: Size of the unit in logical blocks
246 bytes: Unused (zero)

Note: Current implementations of Files-11 do not correctly initialize the word pairs containing the number of free blocks and the free block pointer for each bitmap block, nor are those values maintained as space is allocated and freed on the unit. They are therefore best looked upon as $(2*n)$ garbage words, and

should not be used by future implementations of Files-11 until the disk structure is formally updated.

5.2.2 Storage Bitmap

Virtual blocks 2 through $(n+1)$ are the storage bitmap itself. It is best viewed as a bit string of length m , numbered from 0 to $(m-1)$, where m is the total number of logical blocks on the unit rounded up to the next integer multiple of 4096. The bits are addressed in the usual manner (packed right to left in sequentially numbered bytes). Since each virtual block holds 4096 bits, n blocks (where $n = m/4096$) are used to hold the bitmap.

Bit j of the bitmap represents logical block j of the volume; if the bit is set, the block is free; if clear, the block is allocated. Clearly, the last k bits of the map are always clear, where k is the difference between the true size of the volume and m , the length of the bitmap.

The size of the bitmap is limited to 256 blocks. In fact, due to existing implementations on all RSX system, the retrieval pointers must be in one of the following two forms:

1. A single retrieval pointer mapping the entire BITMAP.SYS file.
2. Two retrieval pointers, the first mapping only the storage bitmap control block, and the second mapping the entire storage bitmap.

This restriction limits ODS-1 to a volume of 4096*255 blocks (1,044,480 blocks) or about 500 megabytes.

5.3 Bad Block File

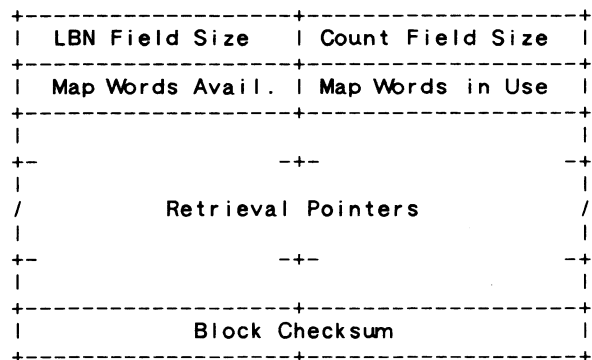
The bad block file is File ID 3,3,0. It is listed in the MFD as BADBLK.SYS;1. The bad block file is simply a file containing all of the known bad blocks on the volume.

5.3.1 Bad Block Descriptor

Virtual block 1 of the bad block file is the bad block descriptor for the volume. It is always located on the last good block of the volume. This block may contain a listing of

the bad blocks on the volume, produced by a bad block scan program or diagnostic. The format of the bad block data is identical to the map area of a file header, except that the first four entries (M.ESQN, M.ERVN, M.EFNU and M.EFSQ) are not present. The last word of the block contains the usual additive checksum. (See section 3.4.3 for a description of the map area.) This block is included in the bad block file to save the data it contains for future re-initializations of the volume.

5.3.2 Bad Block Descriptor Layout



5.4 Master File Directory

The master file directory (MFD) is File ID 4,4,0. It is listed in the MFD (itself) as 000000.DIR;1. The MFD is the root of the volume's directory structure. It lists the five known files, plus whatever the user chooses to enter. In the two level UFD structure described in section 4.1.1, the MFD contains entries for all user file directories.

5.5 Core Image File

The core image file is File ID 5,5,0. It is listed in the MFD as CORIMG.SYS;1. Its use is operating system dependent. In general, it provides a file of known File ID for the use of the operating system for use as a swap area, as a monitor overlay area, etc.

6.0 FCS File Structure

File Control Services (FCS) is a user level interface to Files-11. Its principal features are a record control facility that allows sequential processing of variable length records, and sequential and random access to fixed length record files. FCS interfaces to the virtual block facility provided by the basic Files-11 structure.

6.1 FCS File Attributes

FCS stores attribute information about the file in the file header's user attribute area (H.UFAT - see section 3.4.1.9). It uses only the first 7 words; the rest are ignored by FCS, but are reserved by DEC. RMS uses an additional 3 words - 10 words in all - for relative and indexed file attributes.

The following items are contained in the attribute area; they are identified by the usual symbolic offsets relative to the start of the attribute area. The offsets may be defined in assembly language programs by calling and invoking the macro FDOFF\$. Flag values and bits may be defined by calling and invoking the macro FCSBT\$. These macros are in the system macro library of any operating system that supports Files-11. Alternatively, all these values are defined in the system object library of any system that supports Files-11, and may be obtained at link time.

6.1.1 F.RTYP 1 byte Record Type

This byte identifies the type of records contained in this file. The following three values are legal:

R.FIX Fixed length records
R.VAR Variable length records
R.SEQ Sequenced variable length records

6.1.2 F.RATT 1 byte Record Attributes

This byte contains record attribute bits that control the handling of records under various contexts. The following flag bits are defined:

FD.FTN Use Fortran carriage control. The first byte of

each record is to be interpreted as a standard Fortran carriage control character when the record is copied to a carriage control device.

FD.CR Use implied carriage control. When the file is copied to a carriage control device, each record is to be preceded by a line feed and followed by a carriage return. FD.FTN and FD.CR are mutually exclusive.

FD.PRN The two byte sequence number field for R.SEQ record format is to be interpreted as print control information. See section 6.3.3.1 for format of print information.

FD.BLK Records do not cross block boundaries if set. Generally, there is dead space at the end of each block; how this is handled is explained in the description of record formats in section 6.3.

6.1.6 F.FBY 2 bytes First Free Byte

This word is a count of the number of bytes in use in the virtual block containing the end of file, i.e., it is the offset to the first byte of the file available for appending. Note that an end of file that falls on a block boundary may be represented in either of two ways:

1. F.EFBK contains precisely n blocks, F.EFBK contains 512.
2. F.EFBK contains (n+1), F.EFBK contains 0.

6.1.7 S.FATT 14 bytes Size of Attribute Block

This symbol represents the total number of bytes in the FCS file attribute block.

6.1.3 F.RSIZ 2 bytes Record Size

In a fixed length record file, this word contains the size of the records in bytes. In a variable length record file, this word contains the size in bytes of the longest record in the file.

6.1.4 F.HIBK 4 bytes Highest VBN Allocated

This 32-bit number is a count of the number of virtual blocks allocated to the file. Since this value is maintained by FCS, it is usually correct; however, it is not guaranteed, since FCS is a user level package.

6.1.5 F.EFBK 4 bytes End of File Block

This 32-bit number is the VBN in which the end of file is located. Both F.HIBK and F.EFBK are stored with the high order half in the first two bytes, followed by the low order half.

6.2 FCS File Attribute Block Layout

F.RATT	Record Attributes	Record Type	F.RTYP
	Record Size		F.RSIZ
	Highest VBN Allocated		F.HIBK
	End of File VBN		F.EFBK
	First Free Byte		F.FBY
			S.FATT

6.3 Record Structure

This section describes how records are packed in the virtual blocks of a disk file. In general, FCS treats a disk file as a sequentially numbered array of bytes. Records are numbered consecutively starting with 1.

6.3.1 Fixed Length Records

In a file containing fixed length records, the records are simply packed end to end with no additional control information. If the record length is odd, each record is padded with a single byte. The content of the pad byte is undefined. For direct access, the address of a record is computed as follows:

Let n = record number
 k = record size in bytes
 m = byte address of record in file
 q = number of records per block
 j = VBN containing the start of the record
 i = byte offset within VBN j

Then $h = ((k + 1) / 2) * 2$ (rounded-up record length)
 $m = (n - 1) * h$
 $j = (m / 512) + 1$ (truncated)
 $i = m \bmod 512$

The previous discussion assumes that records cross block boundaries, that is, FD.BLK is not set. If records do not cross block boundaries, they are limited to 512 bytes, and the following equations apply (variables defined as above):

$h = ((k + 1) / 2) * 2$ (rounded-up record length)
 $q = 512 / k$ (truncated)
 $j = ((n - 1) / q) + 1$ (truncated)
 $i = ((n - 1) \bmod q) * h$

6.3.2 Variable Length Records

In a file containing variable length records, records may be up to 32,767 bytes in length. Each record is preceded by a two byte binary count of the bytes in the record. The count does not include itself. A null record is represented by a single 0 word. The byte count is always word-aligned; i.e., if a record ends on an odd byte boundary, it is padded with a single byte. The contents of the pad are undefined.

If records do not cross block boundaries (FD.BLK is set), they are limited to a size of 510 bytes. A byte count of -1 is used as a flag to signal that there are no more records in a particular block. The remainder of that block is then dead space, and the next record in the file starts at the beginning of the next block.

6.3.3 Sequenced Variable Length Records

The format of the sequenced file is identical to a variable length record file, except that a two-byte sequence number field is located immediately after the byte count field of each record. This field contains a binary value which is usually interpreted as the line number of that record (see section 6.1.2, FD.PRN, and section 6.3.3.1).

The sequence number is not returned as part of the data when a record is read, but is available separately. Note that the record byte count field counts the sequence number field as well as the data of the record.

6.3.3.1 Format of 2-Byte Print Control Field in R.SEQ Records

If the FD.PRN bit is set in the record attribute, then the two-byte "sequence number" field is used to contain carriage control data for the record. Byte 0 is print control information to act upon before the record data is output to a unit record device. Byte 1 is print control information to act upon after the record data is output to a unit record device.

The format of each byte is as follows:

Bit 7	Bits 6 - 0	Meaning
0	0	No carriage control
0	1 - 127	Count of new lines (CR/LF)

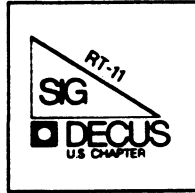
Bit 7	Bit 6	Bit 5	Bits 4-0	Meaning
1	0	0	ASCII C0 set	Char to output
1	0	1	ASCII C1 set	Char to output
1	1	0	Code 0 - 63	Device specific code
1	1	1	-	Reserved

Note: The print control field is not currently supported by FCS or RMS-11.

THE
mini-
tasker

DECUS

RT-11 SIG NEWSLETTER



RT-11 SIG Steering Committee

RT-11 SIG Chairman

John T. Rasted
JTR Associates
58 Rasted Lane
Meriden CN 06450
203 / 634-1632

Newsletter Editor

COBOL Contact
Bill Leroy
The Software House, Inc.
P. O. Box 52661
Atlanta, GA 30355-0661
404 / 231-1484

Standards Coordinator

Kenneth L. Aydlott
Teledyne Hastings-Raydist
P. O. Box 1275
Hampton, VA 23661
804 / 723-6531

*Tape Copy Generation
Contact*

Ralston Barnard
Division 7523
Sandia Laboratories
Albuquerque, NM 87185
505 / 844-5115

APL Contact

Doug Bohrer
Bohrer and Company
903 Ridge Road, Suite 3
Wilmette, IL 60091
312 / 251-9449

MACRO Contact

Nick Bourgeois
NAB Software Services Inc
P. O. Box 20009
Albuquerque, NM 87154
505 / 298-2346

TECO Contact

Product Planning Contact
John Crowell
CROWELL Ltd.
145 Andanada
Los Alamos, NM 87544
505 / 662-3893

DECNET Contact

Ken Demers
Adaptive Automation
5 Science Park
New Haven, CT 06511
203 / 786-5050

*RT-11 Hardware Contact
C Contact*

Carl Lowenstein
Marine Physical Lab
Scripps Inst Oc'graphy
San Diego, CA 92152
619 / 294-3678

Wish List Contact

UNIX Contact
Bradford Lubbell
L. A. Heart Lab
UCLA A3-381 CHS
Los Angeles, CA 90024
213 / 825-9290

TSX Contact

Jack Peterson
Horizon Data Systems
1899-E Billingsgate Cir
Richmond, VA 23233
804 / 740-9244

FMS Contact

CommComm Representative
Susan Rasted
Software Dynamics Inc.
85 Barnes Road
Wallingford, CT 06492
203 / 265-2226

Symposia Coordinator

Ned Rhodes
Software Systems Group
1684 Gude Road
Rockville, MD 20850
301 / 340-2773

Tape Copy Distribution

RT DECUS Library Contact
Tom Shinall
General Scientific Corp.
1684 East Gude Drive
Rockville, MD 20850
301 / 340-2773

Pre-Symposia Seminar

RT-11 Suite Manager
Bruce Sidlinger
Sidlinger Computer Corp
4335 N.W. Loop 410, #209
San Antonio, TX 78229
512 / DIG-ITAL

BASIC Contact

Ed Stevens
E M D A Inc.
77 N Oak Knoll #104
Pasadena, CA 91101
818 / 795-5991

CAMAC Contact

J. W. Tippie
Kinetic Systems, Inc.
11 Mary Knoll Drive
Lockport, IL 60441
815 / 838-0005

LUG Contact

Personal Computers
Bill Walker
Monsanto Research Corp.
P. O. Box 32
Miamisburg, OH 45342
513 / 865-3557

RUNOFF Contact

FORTTRAN Contact
Robert Walraven
Multiware, Inc.
139 G Street, Suite 161
Davis, CA 95616
916 / 756-3291

Notes from the RT-11 World

Copyright's

All copyrights in the RT-11 Mini-Tasker belong to the owner/submitter of the material in the Mini-Tasker, and not to the RT-11 SIG, DECUS or Digital Equipment Corporation. If you have a question about any article in the Mini-Tasker, please contact the author directly - not the Editor - and not DECUS.

We DO solicit signed articles for insertion in the Mini-Tasker, on or about bugs, features, nifty things, etc., all about the RT-11 operating system and its environment. Write it up, send it to me (with a note to rewrite if you wish) and I will try and get it in an upcoming issue of the Mini-Tasker.

Trivia

With thanks to Diana Miller, RT-11 Product Manager, Digital put together an interesting DECUS program last year on PDP-11 trivia. Following are some interesting RT-11 (and PDP-11) facts of life.

1. What does "SIG" stand for?
2. What was the first industry standard magnetic tape on the Q-bus?
3. What was a DX11?
4. Who was the first PDP-11 architect?
5. How many orders for the PDP-11 were received during the first week of its introduction?
6. Who did Digital ship the first PDP-11 to?
7. Where was the 1974 Fall DECUS held?

Cheers and Jeers

Contrary to rumor, RT-11 is alive and kicking and has an active Digital development team. [Come to a national DECUS Symposia, and you can actually meet and talk to them.] In fact, Digital sold the 100,000th license this past year to a national laboratory in Los Alamos. However, we need to convince Digital and others that a great many of these RT-11 licenses are still active -- whether stand-alone, or running as the base for third party multi-user systems -- and to continue to promote RT-11, both PDP-11 (!) and VAX (?) flavors, in all of the Digital publications, and those of the rest of the world.

In that light, I would like to offer the following "Cheers" and "Jeers" to people who have either "helped" or "hurt" the RT-11 world recently.

Cheers to Digital for a new marketing plan, "SOFTbase system (tm)", an online system that provides to all internal employees information

Notes from the RT-11 World

about software products that operate on Digital hardware. SOFTbase will become the main source of solution information for Digital's sales force, and will also be used as the source of software and supplier information for Digital's reference catalogs.

Jeers to "Hardcopy" magazine for dropping Milton Campbells "RT-11 Perspective" column. If you read Milton's column and would like to try and get it reinstated, please write to the editors to inform them of that fact -- and ask them to support RT-11 in general.

Cheers to "Digital Review" newspaper for Bill Walkers column, and the nice complete page on why we need RT-32 on the VAX. Bill is the LUG and Personal Computer contact on the RT-11 SIG. [Bill is also smart -- in each column he puts what he will write about in the next column as a teaser.]

[p.s. Send in your own Cheers and Jeers, and we will try and print them in future issues.]

Notes from the 1986 Australia DECUS Symposium

Rally Barnard recently returned from the Australian Symposium, and has submitted a dynamite article. Please read it carefully. Rally is the Tape Copy Generation Contact on the RT-11 SIG.

If you attend the Symposia of another chapter, please let us know of your experiences. Sally Antine of the European RT-11 community visited the Dallas Symposium -- Nick Bourgeois will be attending the European Symposia, Bob Walraven has been and will again. -- Possibly we should work to a joint "Woods" meeting between the RT-11 SIG's of the four DECUS chapters.

RT-11 and LUG's

Please keep up your efforts to have an active RT-11 group within each LUG. If you will let me know, we will be happy to publicize your good work in the Mini-Tasker. We continue to praise the activities of the SCURT LUG, the Southern California Users of RT-11.

Electronic Distribution of RT-11 SIG tape

To: The RT-11 world

From: Tom Shinal, RT-11 SIG Tape Copy Coordinator

Re: Electronic Distribution of RT-11 SIG tape

The August 1986 figures are in for the usage of the Electronic Distribution of selected portions of the Fall RT-11 SIG tape.

There were 73 sessions for a total of 163 connect hours (not necessarily valid due to hung modems, noisy lines, etc.), using 35:22.1 minutes of CPU time logged on a PDP-11/73. This statistic is significant.

The test has shown itself to be a success.

Trivia - Answers

1. Special Interest Group
2. TSV05
3. A unibus to IBM 360 channel interface (CSS product)
4. Bruce Delagi
5. 150
6. Two customers were shipped on the same day:
Radiation, Inc. and Computer Machine Corp.
7. San Diego, California

And finally, I am always looking for something to print.

Please send your submissions to The Mini-Tasker (RX-01, RX-50, or pieces of paper) to me at:

Bill Leroy
The Software House, Inc.
2964 Peachtree Road, NW #320
P. O. Box 52661
Atlanta, GA 30355-0661
404/231-1484 or DCS (LERoy)

Notes from the
1986 DECUS Australia Symposium

R. W. Barnard
Albuquerque, NM 87185

Approximately 600 people attended the Symposium, which was held in Melbourne, Victoria. The Symposium included a DEC exhibit; there was nothing of the nature of a competing DEXPO exhibit. The following notes on the Symposium are centered on the PDP-11 hardware and software.

The DEC product exhibit was directed exclusively toward VAXes and "workstation" PC's. There were no PDP-11's in the exhibit, and the non-hardware displays (documentation, software, etc) ignored them equally. There was a well-equipped campground with several PDP's, but this was not associated with the DEC exhibit. All the PDP operating systems were available in the campground.

DEC sent several representatives to the Symposium, including one each from the various PDP operating system groups (Jim Metsch was the representative for RT).

Notes on Sessions

Of special interest to PDP users was a three-part session on "The Future of the PDP-11". The first part was delivered by LeRoy Rodgers of DEC, entitled "The Commitment Continues". It turned out to be largely the same presentation as was made at DECworld in February. The essence of this talk was that as long as PDP's keep selling, DEC will keep cranking them out. Specifically, he said that PDP sales volume is still quite large, and although it is not growing, it is not rapidly decreasing either. He also discussed the new products (11/83-84 and 11/53) as illustrations of DEC's commitment to new PDP-11 products.

The second part of the series was "The User Community's View", which turned into a gripe session directed at DEC management. Some of the user comments included:

1. Australian users now find it hard to get DEC to sell them a PDP. The salepeople don't like to deal with this product line.
2. DEC is stressing system products over board-level products. In many cases there is not reasonable upgrade path for users who wish to only replace the CPU in a system.
3. There were many complaints about the inadequacy of the J-11 based PDP's and VAXes as replacements for 11/70's or other PDP-11's in some specialized applications. Both real-time and commercial inadequacies were described.

4. In contrast, there were some representatives of commercial sites who had kind words for DEC's current strategy, claiming that the J-11 based systems sold in boxes met their needs well.
5. Another shot was aimed at what users considered wasted efforts by DEC - the PDT and PRO, which are (or soon may be) dead-end models. The 11/53, which uses a detuned J-11 chip, was also criticized because the detuning efforts (removing the cache, providing one level of interrupts) appeared to involve no cost savings in the hardware.
6. Because of the proliferation of PC's, the competition for new customers for which the PDP-11 is appropriate is intense. On the other hand, the installed base of PDP-11 customers is very large. A major complaint of users is that DEC's marketing and product planning strategy for PDP's still seems to be aimed more at trying to get new customers than it is at retaining and satisfying the current ones. It is feared that their failure to get new customers will be used as an excuse to kill the PDP.

I did not attend the third session on the future of the PDP because I needed to return to vacationing.

RT-11 Specific Items

The major complaints directed toward the RT development effort were that too much time and effort was being expended on supporting new hardware devices (i.e., fishing for new customers), and not enough on enhancing the existing RT components (i.e., supporting the installed base). The DU handler support for RA-class disk drives was considered an example of a wasted effort, and the lack of an upgrade of KED was considered an omission which would have benefitted current users.

The wish list session produced a number of requests, including the perennial request for an initialization file for KED. Also for KED, VT200 support, replace mode, multiple LEARN-mode macros, and journaling were requested. Several constructs similar to those available in MS-DOS were requested. For example, a PATH construct would allow the .LOOKUP programmed request to be done on a specified sequence of devices. An ASSIGN/PROTECT command would prevent the DEASSIGN command from automatically deassigning that logical name. (This would be very useful for assigning LS to be LP, for example). A request was made for inclusion of time in the directory the way TSX+ does. A request was made for a BACKUP/NOVERIFY command. This would allow kamikaze-style backups where the user doesn't need to be assured that the backup was error-free.

The RT users' group is very active and vocal (no surprise, I guess). The group is led by Dr. Chester Wilson and Ray Di Marco; both are extremely competent RT gurus. The SIG is preparing a Symposium Tape, which should be available shortly.

Summary and Conclusions

From the sessions and discussions I attended, it appears that PDP-11 users are getting a bit nervous about DEC's intentions toward their favorite computer.

Other conclusions and observations are:

DEC's new market strategy of trying to be a supplier of systems instead of board-level products is not always in the best interest of the PDP user community.

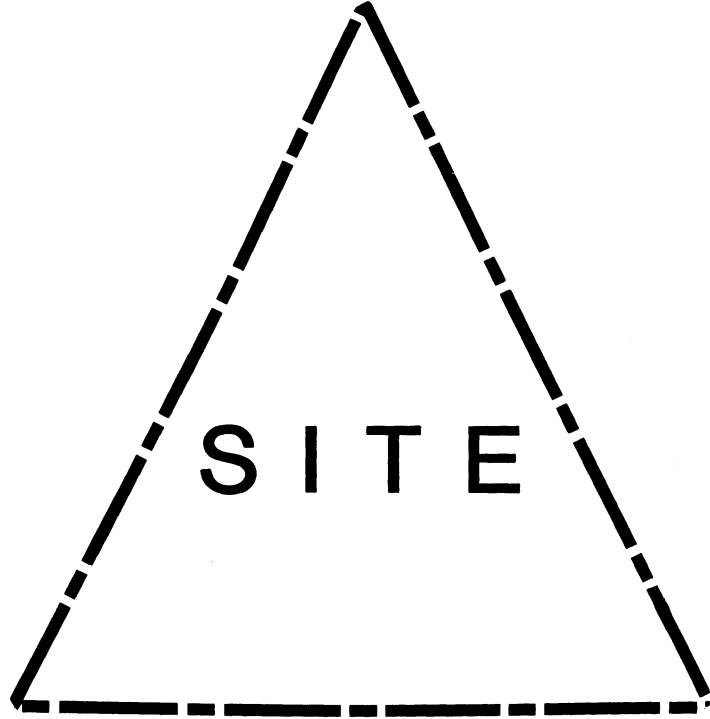
DEC engineering has always done a superb job of creating and manufacturing their products, but their product planning is often a disaster.

The DEC strategy of "One architecture, one operating system, one ..." seems to exclude PDP-11's. The stress on networking similarly is not always appropriate for PDP-11's, either because of software overhead or because networking is inappropriate for the application. There are many DEC representatives who seem to feel that the highest and best use of a computer is to be a node on a network.

The RT-11 users in Australia are very active and interested in improving RT.



DECUS



SITE

SITE SIG STEERING COMMITTEE

CHAIRMAN

DMS SIG Liason
Larry W. Hicks
Relational Database services
P.O. Box 644
121 S. Main St.
Kernersville, NC. 27285-0644
(919) 996-4882

SYMPOSIA COORDINATOR

Sue Abercrombie
48 Malilly Rd.
Portland, ME 04103
(207) 772-2837

SESSION NOTE EDITOR

Large SIG liason
Gary Bremer
Emerson Electric Co.
8100 W. Florissant
St. Louis, MO. 63136
(314) 553-4448

NEWSLETTER EDITOR

Networks SIG Liason
OA SIG Liason
Gregory N. Brooks
Washington University
Behavior Research Labs
1420 Grattan St.
St. Louis, MO. 63104
(314) 241-7600 ext. 257

LIBRARY COORDINATOR

RSTS SIG Liason
Timothy Frazer
Specialized Bicycle Components
15130 Concord Circle #77
Morgan Hill, CA. 95037
(408) 779-6229

HARDWARE COORDINATOR

HMS SIG Liason
Emily Kitchen
A.H. Robbins Co.
1211 Sherwood Ave. RT-2
Richmond, VA. 23220
(804) 257-2925

COMMUNICATIONS COMMITTEE REP

AI SIG Liason
Terry C. Shannon
Digital Review
160 State St.
6th Floor
Boston, MA. 02109
(617) 367-7190

PRE-SYMPOSIA SEMINAR COORD

Phillip Ventura

STAFF MANAGEMENT

Adam Zavitski
Simmonds Precision ICD
3100 Highland Blvd.
Raleigh, NC. 27625
(919) 872-9500

MEMBERS-AT LARGE

Ann Goergen
Texas Instruments
13510 N. Central
M/S 437
Dallas, TX. 75266
(214) 995-4629

HMS SIG Liason

RT SIG Liason
David Hunt
Lawrence Livermore National Lab
MS L-230
P.O. Box 808
Livermore CA. 94550
(802) 656-3190

Gary Siftar
Digital Equipment Corporation
Tulsa, OK.

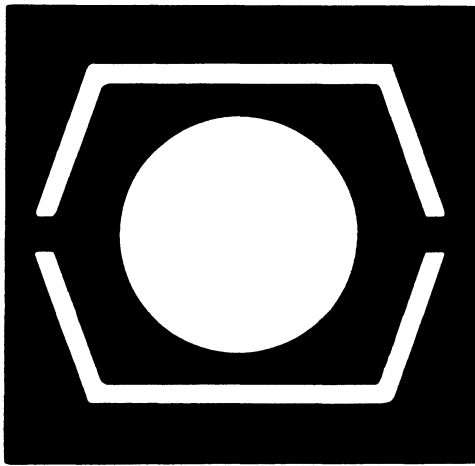
DEC COUNTERPARTS

Joe Allen
Digital Equipment Corp.

Lil Holloway
Digital Equipment Corp.

Susan Porada
Digital Equipment Corp.

UNISIG



DECUS

UNISIG Chair

James W. Livingston, Jr.
Measurex Corporation
1 Results Way
Cupertino, CA 95014
408-255-1500 x5556
ihnp4!decwrl!jwl

UNISIG Symposia Coordinator

Stephen M. Lazarus
Ford Aerospace, MS X-20
3939 Fabian Way
Palo Alto, CA 94303
415-852-4203
ihnp4!fortune!wdl!sml

UNISIG Session Notes Editor

Kurt L. Reisler
Hadron Incorporated
9990 Lee Highway
Fairfax, VA 22030
703-359-6100
decvax!seismo!hadron!klr

UNISIG Newsletter Co-editor

William Toth
Harvard-Smithsonian
Center for Astrophysics
60 Garden Street, P-353
Cambridge, MA 02138
617-495-7181
harvard!hrvsmth!toth

UNISIG Newsletter Co-editor

James W. Livingston, Jr.

UNISIG Administrative Daemon

Dorothy Geiger
The Wollongong Group
49 Showers Drive, 451
Mountain View, CA 94040
415-948-1003
ihnp4!decwrl!dgeiger

UNISIG Tape Librarian

Carl Lowenstein
Marine Physical Laboratory
Scripps Institute of Oceanography, P-004
La Jolla, CA 92093
619-294-3678
(ihnp4|decvax|akgua|dcdwest|ucbvax)
!sdcsvax!mplvax!cdl

UNISIG Usenet Liason

Joe Kelsey
FlexComm Corporation
711 Powell Ave. SW
Renton, WA 98055
allegra!fluke!joe

UNISIG Standards Coordinator

Jeff Gilliam
National Semiconductor
2900 Semiconductor Drive, MS C2303
Santa Clara, CA 95051
408-721-3801
ihnp4!nsc!voder!jeff

UNISIG Minister Without Portfolio

Norman Wilson
Bell Laboratories, 2C-529
600 Mountain Avenue
Murray Hill, NJ 07974
201-582-2842
(decvax|ihnp4)!research!norman

UNISIG DEC Counterpart

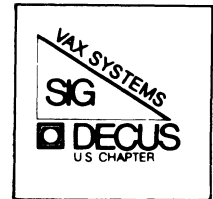
Roseann Maclean
DEC, MKO2-1/H10
Continental Boulevard
Merrimack, NH 03054
603-884-5702
decvax!maclean

NEWSLETTER OF THE VAX SYSTEMS SIG



Pageswapper

Our Mascot



Editor's Workfile	VAX-3
INPUT/OUTPUT	VAX-4
Registration for Non-DEC VMS Products	VAX-24
Tape Management System Needs	VAX-28
VAX System SIG Committee List	VAX-46

Forms at the End

INPUT/OUTPUT Submission Form	QU-9
System Improvement Request Submission Form	QU-11
Non-DEC VMS Product Registration Form	QU-13
VAX Tape Drive Questionnaire	QU-15

General material for publication in the Pageswapper should be sent (US mail only -- no "express" services please) to:

Larry Kilgallen, PAGESWAPPER Editor
Box 81, MIT Station
Cambridge, MA 02139-0901
USA

Preference is given to material submitted as machine-readable text (best is Runoff source). Line length should not exceed 64 characters and the number of text lines per page should not exceed 48 (these limits are particularly important for sample commands, etc. where simple text justification will not produce a meaningful result).

Please do not submit program source, as that is better distributed on the VAX SIG tape.

Please do not submit "slides" from DECUS Symposia presentations (or other meetings) as they are generally a very incomplete treatment for those readers of the Pageswapper who are not so fortunate as to be able to travel to Symposia. Please DO write articles based on such slides to get the content across to a wider audience than is able to attend.

For information about on-line submission to the Pageswapper dial:

(617) 262-6830

(in the United States) using a 1200 baud modem and log in with the username PAGESWAPPER.

Change of address, reports of non-receipt, and other circulation correspondence should be sent to:

DECUS U.S. Chapter
Attention: Publications Department
249 Northboro Road (BPO2)
Marlborough, MA 01752
USA

Only if discrepancies of the mailing system are reported can they be analyzed and corrected.

Editor's Workfile

by Larry Kilgallen, Pageswapper Editor

Software Facility Registration -

is certainly a idea whose time has come. Apparently the reception from the user community has been underwhelming, perhaps because the "Developer's Guide to VMSINSTAL" is somewhat hard to come by (I have had one on order for several months). So this month you will find a description in the Pageswapper and the relevant form at the back of the combined newsletters to apply for registration.

At any rate, the SQM people at DEC have been really helpful in setting up registrations, and the consistency across software from various sources, DEC, 3rd parties and even DECUS tapes should be of great assistance. If you haven't considered what a lack of coordination could do, check out Notes 532.* in the Input/Output section this month.

Perhaps the most appealing thing about this registration program is that it is entirely divorced from marketing. The goal is a technical one, letting software from various sources co-exist under VMS without conflict. The solution is straightforward, and DEC's SQM group had the good sense not to take on more than they could handle. I asked about some other items (beyond those covered by the article and signup form in this issue) and the response was that since various technical problems prevented effective action on those items, they were not attempting to include those items in the process. Most important, I think, is they were not waiting until every last nit was ironed out before moving forward in the areas where coordination COULD be accomplished.

Speaking of not waiting for total solutions -

Dave Schmidt has done an admirable job of cataloging a number of tape library issues for all of us. While I would like to see DEC (or others) offer general solutions in this area, I think the most important thing is to provide the VMS hooks in tape handling (including backup) to allow sites to implement their own in the meantime. Even if one wants to spend a lot of money implementing site-specific software to implement a labeled tape shop, it can't be done properly today without modifying VMS!!!

INPUT/OUTPUT

A SIG Information Interchange

A form for INPUT/OUTPUT submissions is available at the back of the issue.

For information about on-line submission to the Pageswapper dial:

(617) 262-6830

(in the United States) using a 1200 baud modem and log in with the username PAGESWAPPER.

```
=====
Note 331.1 Listing of n Largest Files on a Files=ll Device 1 of
"Jamie Hanrahan"          9 lines  8-SEP-1986 12:47
      -< Use DIRECTORY and SORT >-
-----
```

Why not do a DIRECTORY/NOHEAD/NOTRAIL/SIZE into a disk file, then SORT

the resulting file into descending order using the columns where the file size appears as your key field, then print the first 'n'

records of the sorted file? A special-purpose program is unlikely

to be significantly faster than the directory utility. Besides, you've ALREADY GOT the directory utility. Use what you have!

The only likely glitch is that DIRECTORY may emit two lines per file if the file names are long; this could be cleaned up with a

filter program before you sort.

Jamie Hanrahan
Simpact Associates
c/o PO Box 261687
San Diego, CA 92126
619-565-1865

```
=====
Note 331.2 Listing of n Largest Files on a Files-ll Device 2 of
"Bruce Bowler"          3 lines  8-SEP-1986 14:06
      -< SIG tape, V4.x DIR >-
-----
```

I seem to recall a program on a SIG tape not too long ago called bigfiles (or something similar). It worked well. Also in 4.x you can tell DIR to select based on file size, see DCL manual for syntax.

Bruce Bowler
General Electric
1 River Road
Bldg 2 Room 609
Schenectady, NY 12345

```
=====
Note 331.3 Listing of n Largest Files on a Files-ll Device 3 of
"James B. Fischer"      5 lines  24-SEP-1986 08:49
      -< HELP with DIR output >-
-----
```

Sorting the DIRECTORY output is the way to go. To avoid the possible multiple line output from DIRECTORY (for file names that are long) use the /WIDTH=FILENAME:number to specify larger fields for files. (i.e. type \$ HELP DIRECTORY /WIDTH for more ...)

James B. Fischer
MIVAXLUG Chair
EDS / Plant Support
P.O. Box 7019
Troy, Michigan 48007-7019
(313) 524-8887

```
=====
Note 387.1      Disk Usage by Directory Tree      1 of 1
"Jamie Hanrahan"      5 lines      8-SEP-1986 13:02
      -< not elegant, but it works >-
-----
```

I did this with a command procedure. It uses f\$search to find the name of each .DIR file in [000000], then plugs this name into DIR [name'...]/SIZE/GRAND_TOTAL. Hokey, but fast enough for our purposes, even on a full RA81.

Jamie Hanrahan
Simpact Associates
c/o PO Box 261687
San Diego, CA 92126
619-565-1865

```
=====
Note 477.2      Using incoming modems as outgoing modems      2 of 3
"Offline Submission"      13 lines      9-SEP-1986 23:45
      -< Another Incoming/Outgoing Modem Security Hazard >-
-----
```

The DZ11 terminal driver, also used for Emulex DH/DM emulators, does not insist upon seeing DCD and DSR before transmitting. It reacts only to CHANGES in DCD, i.e., if DCD goes from ON to OFF, it will hang up the phone.

Michael Laboley
VAX Systems Manager
General Research Corporation
5383 Hollister Avenue
Santa Barbara 93111
Telephone (805) 964-7724 x271

September 2, 1986

```
=====
Note 477.3      Using incoming modems as outgoing modems      3 of 3
"Offline Submission"      54 lines      23-SEP-1986 10:29
      -< Incoming/Outgoing Modems >-
-----
```

To review the problem described in I/O 477.1, a port used for an incoming modem must be defined as /MODEM. This causes VMS to stop the active process when it detects the drop in CD/DSR when the line is disconnected. If the drop in CD/DSR was not detected, the next user dialing into the same line would get into the previous user's active process. However, VMS insists on seeing CD/DSR before it will send any output to the modem. Therefore, because CD/DSR is usually not raised until a call has been connected, you cannot send a dialing string to an autodial modem unless you set CD/DSR high on the modem. But, since VMS will no longer see CD/DSR drop, this defeats the ability of VMS to detect a line disconnect.

Although our modems are connected to a data switch (so they can be shared), the way it handles the CD/DSR control signals is similar. We solved the problem by purchasing modems with the capability to drop CD/DSR/CTS for only a few seconds when the line disconnects. After trial testing various 2400 bps modems, we decided to purchase Microcom AX2400C modems because of the options for handling CD/DTR/CTS, error correction, data compression, speed conversion, and the fact that all of the native mode commands were supported in the Hayes-type AT command set.

However, there is one catch: once the Microcom modems see someone dial out at 1200 bps, they assume there is a terminal connected at 1200 bps and anyone dialing in gets a 1200 bps connection instead of a 2400 bps one, even after a software reset. You must either dial out at 2400 bps or higher or press the reset button on the back of the modem before incoming calls will get connected at 2400 bps again. We solved this by switching requests for dialing out at 1200 bps to Hayes 1200 bps modems. With the data switch, this can be done automatically. With VMS, you should set the port speed to 2400 bps with "SET TERM /PERM /SPEED=2400 /NOAUTOBAUD".

When users access the modem with "SET HOST /DTE termid", VMS will convert the speed to 2400 bps. On incoming calls, the modem can convert a 1200 bps call to the 2400 bps needed for the port.

Robert G. Simpson
Dravo Corporation, 300-12
One Oliver Plaza, 16th floor
Pittsburgh, PA 15222

Telephone: (412) 566-5325

September 16, 1986

=====
Note 483.1 Dialing out on the VAX using a DF112 Modem 1 of 1
"James R. Ostrosky" 9 lines 20-SEP-1986 12:34
-< DF112 DIAL OUT >-

Are you using a DMF-32? A note in the release notes for V4.4 indicates that the response from the DF112 modem is ignored by the DMF-32 until it sees carrier detect. Workaround was to ignore response from modem and poll (using IO\$SENSEMODE!IO\$M_RD_MODEM) for CD. I wrote autodialer in Fortran-77 using this with great success.

James R. Ostrosky
3910 OLD WM PENN HWY
PITTSBURGH, PA 15235

=====
Note 485.1 Executing a CLI command from within a program 1 of 1
"Jamie Hanrahan" 11 lines 8-SEP-1986 13:36
-< not that easy >-

The problem is that the ^C - DCL command - CONTINUE sequence only works if the DCL command does not invoke an image. If you invoke an image after typing ^C, your previous image will be run down, and CONTINUE won't work.

The system service SYS\$SETDDIR (system services book p. 446.3 lets you change the default directory string of your process. To change the default device, just redefine the logical name SYS\$DISK.

Jamie Hanrahan
Simpact Associates
c/o PO Box 261687
San Diego, CA 92126
619-565-1865

=====
Note 501.2 RK05 driver needs for VAX750 (VMS V4.3) 2 of 2
"Offline Submission" 12 lines 23-SEP-1986 10:30
-< RK05 Device Driver >-

I am running VMS V4.2 on a VAX 11/750 with dual RK05s. I have a driver, formatter, and some diagnostic tools.

Michael N. Levine
Naval Weapons Center
Code 3514
China Lake, CA 93555

Telephone: (609) 939-2465

September 15, 1986

=====
Note 502.2 SET HOST/DTE on RACAL/VADIC 3451-PA 2 of 2
"RICHARD WISEMAN" 13 lines 5-SEP-1986 17:02
-< AUTODIAL PROG FOR RACAL/VADIC 3451 >-

I have modified the program vaxdial the is supplied on the decus tapes. to work with a racal/vidic 3451 autodial. it has been working with vms 4.0 thru 4.4 . if you need a copy please call.

RICHARD WISEMAN
STORAGE TECHNOLOGY CORP
2270 SOUTH 88TH STREET
MAIL STOP G4
LOUISVILLE CO 80233-0001

```
=====
Note 503.1          File Transfer to IBM          1 of 1
"RICHARD WISEMAN"      12 lines   5-SEP-1986 17:23
      -< VAX TO IBM FILE TRANSFER >-
-----
```

HAVE YOU LOOKED AT DEC'S 2780/3780 PROTOCOL EMULATOR... IT WILL TRANSFER FILES TO AND FROM IBM MAINFRAMES USING THE BISYNC PROTOCOL.. WE CURRENTLY ARE USING IT WITH GOOD SUCCESS.

RICHARD WISEMAN
 STORAGE TECHNOLOGY CORP
 2270 SOUTH 88TH STREET
 MAIL STOP G4
 LOUISVILLE CO 80233-0001

```
=====
Note 506.2          Printer on DMF Printer Port      2 of 2
"Offline Submission"  44 lines   9-SEP-1986 23:50
      -< DMF-32 Line Printer Port >-
-----
```

I have successfully built my own cables and have attached Dataproducts printers to the line printer ports. The problem is that the cable diagram in the DMF Manual is wrong. Use the following diagram and everything will work. Try to stay within a 50 foot limit.

Printer end	DMF-32 end	Description
19	26	Data 1
3	30	Ret
20	20	Data 2
4	34	Ret
1	22	Data 1
2	28	Ret
41	1	Data 4
40	33	Ret
34	24	Data 5
18	35	Ret
43	23	Data 6
42	34	Ret
36	5	Data 7
35	30	Ret
30	6	Data 8

```
14          28          Ret
38          8          Strobe
37          35          Ret
46          14          Conn Vfy L
45          32          Ret
21          12          On Line High
5           32          Ret
23          18          Demand
7           33          Ret
26          17          DAVFU
10          31          Ret
Shell      Shell      ground
```

Bill Swisher
 Clermont College
 725 College Drive
 Batavia, OH 45103
 Telephone: 513-732-2990

September 4, 1986

```
=====
Note 525.0          DHV Problems          2 replies
"James Littlefield"  6 lines   29-AUG-1986 14:50
-----
```

I have been having trouble getting a Hayes Smartmodem 1200 to answer correctly when attached to a DHV on a MicroVAX. It works fine on a 750 attached to a DMF port, but answers and immediately hangs up on the MicroVAX. Both ports are configured the same. The interesting thing is that it WILL work if I set the port for SECURE_SERVER. All this is happening under VMS V4.3.

James Littlefield
 170 Aquidneck Ave
 Middletown, RI 02840
 (401) 849-8440


```
=====
Note 525.1          DHV Problems          1 of 2
"Larry Kilgallen"  12 lines  30-AUG-1986 08:17
  -< I have a site where Hayes works fine on DHV >-
=====
```

I will try to remember to transfer a listing of the terminal settings from that machine to here. This site, in fact, prefers the Hayes to a Vadec because it works BETTER for incoming.

One question would be which ports of the DMF on the 750 work. Ports 0 and 1 are more like the DHV ports in that they offer modem control. If it is ports 2 - 7 on which it works but not 0 and 1, making it work on 0 and 1 (and the DHV) would require a change in your cable connections.

Larry Kilgallen
Box 81, MIT Station
Cambridge, MA 02139-0901

```
=====
Note 525.2          DHV Problems          2 of 2
"James Littlefield" 2 lines  16-SEP-1986 11:37
  -< DHV response >-
=====
```

No, I was using it on Ports 0 and 1. It's really strange that it only answers with SECURE_SERVER turned on.

James Littlefield
170 Aquidneck Ave
Middletown, RI 02840
(401) 849-8440

```
=====
Note 526.0          "Learn Mode" for ALL-IN-1 V2.x UDPs      No replies
"Alan D. Hull - Digital Equipment" 82 lines  1-SEP-1986 10:09
-----
  -< Code for Learn Mode UDP >-
*****
```

Here's a learn mode for UDP's that I hacked out at the AI prog class last December.

Extract form DEFAULT from MEMRES into a copy called MYDEFAULT in your USER flb. Add the following two key definitions to the named data for MYDEFAULT:

```
-----
;;.GOLD !;;

oa$tra_set input, log\DISPLAY Beginning to log all keystrokes...

;;.GOLD @;;

oa$tra_set off, log\display Logging of keystrokes has
stopped...\force \FORM UDPENT\IFEXIT\DISPLAY Creating new UDP .
. \FORCE \GET #UDPPFILE=".UDP" $CURUDP ".UDP"\RENAME
"ALTRACE.LOG" #UDPPFILE \script clean_udp\EDIT #UDPPFILE
-----
```

Here's the clean_udp.scp - all it does is go back into the captured keystrokes file and strip off the bottom line, which is always the GOLD @ used to turn off the learn mode.

```
!*****
!          clean_udp.scp
!          written by Alan D. Hull 12-6-85

. IF OA$DEFAULT_EDITOR EQS 'EDT' THEN .GOTO edt_style
. IF OA$DEFAULT_EDITOR:3 EQS 'WPS' THEN .GOTO WPS_STYLE
. GOTO BAD_EDITOR

..label WPS_STYLE

{GOLD B}
{GOLD DEL}
{GOLD F}
.goto common_exit
```

```
..label EDT_STYLE
{GOLD B}
{UP}
{PF4}
{GOLD F}
.goto common_exit

..label BAD_EDITOR
.function Display Unknown editor type. Edit UDP & remove
last line manually.\force
..label common_exit
.EXIT
!*****
Remember to do a <SET MENU MYDEFAULT to get these definitions in
effect in your process. I use a UDP called SETDEF to save
keystrokes for this:
```

(contents of SETDEF UDP):

```
*****
..function set_menu MYDEFAULT
..function Display Default named data search now set
to form MYDEFAULT\force
*****
```

If you wanted this available system-wide, just stick it in the real DEFAULT in MEMRES, relink and have fun!

This works great for saving keystrokes for an IVP style demo of anything.

ALAN D. HULL
Digital Equipment Corp
Field Application Center
34119 W. 12 mile Rd.
Farmington Hills, MI. 48018

```
=====
Note 527.0          drvllwa drive for microvax          1 reply
"RICHARD WISEMAN"          21 lines      8-SEP-1986 14:37
-----
```

The DEC supplied xadriver for the microvax does not work with the new rev. level drvll-wa etch c level boards.

Has any one gotten it to work..

The io writes seem to work fine but when you try to do a read qio with modifiers the driver errors out..

TSC in Colorado springs knows about the error but says that VMS engineering also knows but currenty has not done any thing about as of yet...

If any one can help please call or write to me and let me know..

RICHARD WISEMAN
STORAGE TECHNOLOGY CORP
2270 SOUTH 88TH STREET
MAIL STOP G4
LOUISVILLE CO 80233-0001

```
=====
Note 527.1          drvllwa drive for microvax          1 of 1
"WENDY KOENIG"          15 lines      18-SEP-1986 15:20
-----
-< drvllwa board for uvax >-
```

We have used the DEC supplied xadriver for the microvax with a drvll-wa revision c etch e level board and it has worked successfully. I wasn't sure if this was the same level board that you were referring to. It is not officially supported by the DEC driver, and I've already talked to some VMS people about that. We are doing both io reads and writes, but maybe not exactly what you're doing. If you want to talk about it some more, you can give me a call.

WENDY KOENIG
80 Blanchard St.
Burlington, Mass 01803

=====
Note 529.0 WPS+ Printer tables 2 replies
"Jack Patteeuw" 2 lines 9-SEP-1986 08:47

Now that ALLIN1 V2.1 is out, has anybody figured out printer tables for HP LaserJet's? How about XEROX 2700's?

Jack Patteeuw
Ford Motor Co.
Electrical and Electronics Division
31630 Wyoming
Livonia, MI 48150
313-323-8643

=====
Note 529.1 WPS+ Printer tables 1 of 2
"Bob Hassinger" 17 lines 9-SEP-1986 09:09
 -< Other printers, input request for OA SIG >-

Also how about things like the Qume. We have a Diablo D80IF and need a printer table for it to use with WPS-PLUS. The book for it does not really say but I have the impression it looks like a more or less industry standard - a "630".

How about the Apple Laserwriter?

I am working with the OA SIG to try to get information like this into their Newsletter, onto Symposium tapes and into the DECUS Program Library so if you have any information in this area please let me know.

Bob Hassinger
Liberty Mutual Research Center
71 Frankland Road
Hopkinton, MA 01748
617-435-9061

=====
Note 530.0 Remote Print Symbiont wanted 2 replies
"Jamie Hanrahan" 9 lines 12-SEP-1986 13:49

In the NOTES discussions that were reprinted from the Spring DECUS symposium there was some discussion of the right way to implement remote printing over DECnet. A few people were mentioned who had implemented a user-modified print symbiont that connected to a queue on another DECnet node; the software was in the public domain but names and addresses were not given. I was looking at doing something like this (in my copious free time) until I saw these notes. Can anyone provide pointers to the people involved?

Jamie Hanrahan
Simpact Associates
c/o PO Box 261687
San Diego, CA 92126
619-565-1865

=====
Note 530.1 Remote Print Symbiont wanted 1 of 2
"Jack Patteeuw" 18 lines 18-SEP-1986 08:40
 -< DEC Remote Print Symbiont >-

DEC now supports some kind of print symbiont/queuing mechanism for printing on remote printer. I can say this because of a new product they just introduced. The PrintServer 40 (LNV11 I think) is a medium speed (40 pages/min.) laser graphics (uses Adobe's Postscript page formatting language) printer that sit on the Ethernet. Therefore it looks like any other node to all the rest of the network. Each VAX which wishes to queue print requests to the PrintServer must run PrintServer "Client" software which handles requests for the "remote device". The "Client" software license is packaged with the hardware and includes "rights-to-copy".

Now if we all in DECUS could pressure DEC into making this remote print symbiont standard to VMS ...

P.S. If your sales rep doesn't know anything about the PrintServer 40 (do they ever know anything ...) tell him to read his "Sales Update"/Vol. 18 No.5 pages 1-10.

Jack Patteeuw
Ford Motor Co.
Electrical and Electronics Division
31630 Wyoming
Livonia, MI 48150
313-323-8643

```
=====
Note 530.2      Remote Print Symbiont wanted      2 of 2
"Jamie Hanrahan"      3 lines 18-SEP-1986 12:25
-< that's interesting, but... >-
-----
```

That really doesn't solve my problem. It sounds to me as if this will only work on this funny device, not on an ordinary printer on an ordinary VAX.

Jamie Hanrahan
Simpact Associates
c/o PO Box 261687
San Diego, CA 92126
619-565-1865

```
=====
Note 531.0      DMV11 on uVAX install help?      1 reply
"Bernard Klatt"      22 lines 15-SEP-1986 18:19
-----
```

Has anyone successfully installed a DMV11 (DDCMP sync line interface) on a MicroVAX II and has it working with DECnet? The DMV11-M is a M8053 and uses the CK-DMV11-AB for RS-232 interface. I'm trying to install a 4800 baud point-to-point link between the MicroVAX and a PDP 11/44 using a DUP11. After running NETCONFIG.COM it shows up as device XDA0: online. NCP shows line DMP=0 state is ON, shows circuit DMP=0 state ON - starting. EIA break-out box shows no activity, modem control lines look OK. The \$SHO ERR command shows that there are errors associated with XDA0: but no error messages are displayed and there are no errors logged in ERRLOG.SYS. Running the stand-alone 'field service' functional test and performance exerciser diagnostic with the H3254/H3255 test connectors

installed shows DMV11A - Error Number 1502 .. Incorrect initial modem status. Would any of the E101 switch settings affect this? The info on page 3-17 of the MicroVAX Tech Manual doesn't really specify what the switch settings SHOULD be. The DMV11 User's Guide only discusses installation on a PDP11, no mention of installing it on a MicroVAX.

Bernard Klatt
Microtronix
1556 Halford Ave #184
Santa Clara ca 95051
408 991-5149

```
=====
Note 531.1      DMV11 on uVAX install help?      1 of 1
"Jamie Hanrahan"      18 lines 16-SEP-1986 17:17
-< Check switches on dist. panel >-
-----
```

We have used a DMV11 successfully in point-to-point mode on a uV2, both under DECnet and with direct \$QIO access to DDCMP. A phone conversation with Mr. Klatt revealed that he did not have a copy of the DEC manual describing the new RS232 distribution panel being shipped with DMV11s, and that some of the switches on his panel were set wrong. This is not his fault, as DEC does not to my knowledge supply the correct information with the DMV! In particular, Table 2-8 in each of the following manuals is NOT applicable:

EK-DMV11-UG-001, DMV11 Sync Ctlr User's Guide (1981)
EK-DMVQM-UG-001, QMA DMV11 Sync Ctlr User's Guide (1984)
EK-DMVQM-UG-001, QMA DMV11 Sync Ctlr Tech Manual (1984)

The right place to look is in Volume 2 of the "Communications Options Minireference Manual" (EK-CMIV2-RM-002), page DMV11-24. It indicates that S1, S2, S3, S5, S6, S9, S15, S16, and S20 should be off and all others on for normal operation.

Jamie Hanrahan
Simpact Associates
c/o PO Box 261687
San Diego, CA 92126
619-565-1865

```
=====
Note 532.0      Warning - BSO/DECnet problem      2 replies
"Dave Close"    10 lines 19-SEP-1986 12:49
-----
```

Software vendor Boston Systems Office supplies us with cross-assemblers for several microprocessors which run on the VAX. Their latest update specifies that, after installation of the new versions, two system logical names should be defined in the SYSTARTUP file. One of these names is "RT". Defining this name prevents REMACP from starting and instead results in a system crash, fatal bugcheck, code NOTIRPAQB. The problem may be easily reproduced by defining RT, then running @STARTNET. The work-around, of course, is not to define the logical name RT.

Dave Close
Anadex/Printronic
1080 Avenida Acaso
Camarillo, CA 93010
805/987-9660

```
=====
Note 532.1      Warning - BSO/DECnet problem      1 of 2
"Jamie Hanrahan" 18 lines 19-SEP-1986 17:27
-< have you tried... >-
-----
```

What happens if you define the name RT *after* the STARTNET procedure completes? This might really screw up incoming remote terminal connections. Or it might not.

If defining the name later doesn't help, maybe it would be sufficient to define it in the LOGIN.COMs of those users who will be using the package. The LOGIN.COM could look at the translation of the SYS\$COMMAND logical name and, if it contained the string "RT", could refuse to define the RT logical.

Finally, please send the cretins who wrote the package a sternly worded letter. *ALL* non-DEC defined logical names should be of the form facility_name; the imbedded underscore will ensure no collisions with DEC-defined logicals.

Grr. VMS supplies perfectly good mechanisms for keeping user-defined stuff out of the way of DEC-defined stuff. There's just no excuse for this sort of thing.

Jamie Hanrahan
Simpact Associates
c/o PO Box 261687
San Diego, CA 92126
619-565-1865

```
=====
Note 532.2      Warning - BSO/DECnet problem      2 of 2
"Larry Kilgallen" 17 lines 20-SEP-1986 00:14
-< Unfortunately there are many sloppy vendors >-
-----
```

This sort of lazy attitude about fitting into the VMS environment is unfortunately all too common. Software AG has managed to hit the jackpot by telling users to set up a queue name with a dollar sign in it (reserved to DEC) which actually conflicted with a queue name set up for a DEC layered product. Access Technology recently sent out a new release of their spreadsheet with a long list of logical names containing dollar signs which had NOT been there in previous successful releases.

My own reading of the situation is that the vendors most likely to make such mistakes are those who sell "portable" software for a variety of machines, and how well it fits the VMS environment is just a small concern (if at all) of one person in their organization.

Larry Kilgallen
Box 81, MIT Station
Cambridge, MA 02139-0901

*
*

Registration for Non-DEC VMS Products

DIGITAL VMS Product Registrar
SQM/System Quality Group

The VAX/VMS Registrar service has been put in place beginning with VMS V4.4 to enhance the level of confidence in the uniqueness of software products residing on one system. As documented in the "Developer's Guide to VMSINSTALL" this service is being provided to prevent conflicts with customer developed application and existing VMS products. Digital will attempt to insure that your registration data will remain unique on the VMS Operating system, however due to possible noncompliance with the registration service, we cannot guarantee that your data will be unique.

The purpose of registering a product is to prevent conflicts with other software products and allow all VAX/VMS software products and customer developed applications to co-exist in the same environment. Many registration items MUST be unique for each individual product. For example, facility numbers used with the message utility, and logical name prefixes all must be unique. Other registration items promote good coding practices and standards as well as help to identify components of a particular product.

To register your product, simply fill out the form at the back of this issue and send it to the address provided. If the Registrar has any questions or concerns, or you have chosen an item which is already registered to another product, you will be contacted by the Registrar. When your product information has been registered, you will receive a confirmation notice. The confirmation notice will list the items registered for your product. If any of the information is incorrect, please notify the Registrar as soon as possible. If you have any questions at all about the registration process, or need assistance, you MAY contact the Registrar at the address provided.

Definition of Registration Items

A. Facility Name

A Facility Name is an alphanumeric string containing from 3-27 characters. This string is used as a prefix to uniquely identify your product and its components. All Customer software should use their facility name followed by an underscore () to identify the software as Customer supplied. The following is a list of items which should use the facility name as a prefix. You may find additional uses for your facility name. Please indicate if a VMS error message status code number is to be assigned. These are used to signal errors with the VMS Message Utility. See the VMS Utilities volume and the VAX/VMS Message Utility Reference manual for more information.

Global Symbols - Global symbols should be named in the format: facilityname_symboldescription.

Entry Points - Entry points should be named in the format: facilityname_procedurename.

Rights Database Identifiers - If you are using identifiers entered into the Rights Database they should be named in the format, facilityname_rightsidentifier.

Data Structure Definitions - Data structure definitions should be named in the following format, facilityname_string, i.e. DSC_KCLASS, DSC_KDTPET.

File Names - File names should be created using the facility name as a prefix to the file specification. It is recommended that an underscore () follow the facility name prefix, for example, ADA_STAT.EXE.

B. Logical Names Prefix(es)

The logical name prefix(es) that will be used must be registered. This should be the same as the product's facility but may differ. The format that should be used is, facilityname_string, for example, CDD_DICTIONARY. Please note that it is very important to use this naming convention when assigning logical names to permanent mailboxes.

C. System-wide process names

A system resident process is one that is always in the system. It will usually be in hibernation or in a wait state. For system resident processes in a layered product, unique registered names are required to ensure that another digital product does not run a process with the same name as yours. The system-wide process name should be prefixed by the facility name.

D. System-wide mailbox names

Permanent mailboxes are those mailboxes which are entered into LNM_PERMANENTMAILBOX which is usually set to LNM_SYSTEM (the system logical name table). Other processes can then access the mailbox simply by knowing its name and using the \$ASSIGN directive to assign their channel to it. If a process was to perform a create mailbox (\$CREMBX) directive, using a name it thought was unique but already existed, it would get a channel to the existing mailbox and no error indication. This could obviously cause some serious problems if a user thought he was talking to his mailbox, and was really sending messages to another product's mailboxes.

E. Shareable images

If your product ships a Shareable image, you should register the image name with the Registrar. You should use your facility name as a prefix when naming the image.

F. Contact name

The name of one person who will be the Registration Contact for this product. This will be the person whom the VMS Product Registrar will contact for any additional information.

Registration items are assigned on a first come, first serve basis. You must select a new item if someone else is currently using the one that you wanted.

The information in this document is subject to change without notice and should not be construed as a commitment by Digital Equipment Corporation. Digital Equipment Corporation assumes no responsibility for any errors that may appear in this document.

Tape Management System Needs

By

Dave Schmidt
Management Science Associates
5100 Centre Avenue
Pittsburgh, Pa. 15232
(412) 683-9533

Draft White Paper submitted to:

The VAX SIG Commercial Working Group

September 9, 1986

Background and philosophical overview:

The following topical points are in no particular order of priority or preference. I feel that all of the following points need to be considered in evaluating any tape management system software that Management Science Associates would consider. MSA has a library of 38,000+ tapes with the number of tape mounts approaching 1000 a day.

MSA's use of magnetic tape is to cost effectively store and archive large amounts of sequentially accessed data. MSA processes much of its market research data on tape as well as having a need to effectively back up 22 Gigabytes of disk storage from a cluster of 4 11/780's and 1 8650 and 4 stand alone Micro VAX II's. An additional tape processing load is imposed by the high volume of printing spooled to tape for printing on our Xerox 9700 printer, 1.4 to 1.7 million pages a month.

Our tape farm consists of 10 IBM plug compatible 125 ips start/stop 1600/6250 bpi mainframe class tape transports, two Systems Industry 125 ips start/stop 800/1600/6250 bpi mini class tape transports and one CDC 92185 1600/6250 streamer for Micro Vax II backup processing. The IBM class drives are tri ported to the 8650 and two of the 780's, the SI drives are switch selectable between the remaining two 780's and the CDC drive is switch selectable between three of the Micro Vax II's. The

fourth MV II must use a TK50 drive for backup and printing but that is a problem of a different tape.

Though obvious I feel that it should be stated that any site considering a tape management system should have full time operators and a full time tape manager.

One view that I personally have is that the management issue to address is not explicitly a tape or disk management problem. The problem that I feel needs to be addressed is one of data management, independent of device residency. Consequently I have tended to include some issues that historically have been associated with disk backup. The areas of concern over tape mounting procedures addressed in the other part of the draft white papers will not be discussed here.

Additionally I am not considering any motivational issues over use or justifying a tape management system or enforcing its procedures once in use. These issues will vary widely from shop to shop and will sometimes be based on economic motivation and sometimes on political motivation. If DEC considers a tape management system as a product then it will be appropriate discuss these issues in terms of cost/benefit and marketability.

DEFINITIONS

Enumerated issues relating to tape and tape usage management:

1. Data sets. Users will access a data set that is either a disk file or a tape volume set. The tape management system will access disk files in a sequential fashion when migrating disk data sets to tape and back.

Normally a data set on tape has a one for one correspondence with a tape volume set. The exception to this is a tape containment volume set which contains more than one data set. These data sets are logically and physically independent of each other.

2. Volume sets. A volume set consists of one or more tape reels and normally contains one data set. The tape management system must track and identify all component reels of a volume set for any of the supported labeling conventions. It is acceptable to require consistent labeling conventions for all reels in a volume set and all volume sets in a tape shadow set. The tape management system must allow a logical one for one replacement of a physical reel in a volume set by one or two physical reels when that reel needs replacement due to damage or deterioration.
3. Tape volume shadow sets. A facility to allow one or more copies of critical volume sets needs to exist. This facility should follow the disk shadow set structure fairly closely but be adjusted for tape reels of differing lengths. A tape volume shadow set may have more than two component volume sets comprising the set. One volume set would be considered the master volume set and the remaining volume sets would be alternate volume sets.
4. Containment volume set. This is a volume set consisting of one or more reels of tape containing multiple independent data sets. The tape management system must track all data sets independently from tracking tape reels in a containment volume set.

Except for building a containment volume set by appending data sets to the end, users have only read access to the data sets within a containment volume set. The tape management system has responsibility for managing containment volume set compression.

GENERAL

5. Labeling conventions. All major tape labeling conventions and variants including DEC, IBM, ANSI, unlabeled (with and without a tape mark at BOT) must be supported.

6. Retention cycle. Some form of retention cycle support is necessary to avoid data sets sitting in the tape management system forever. However the removal of expired data sets must not occur until a positive acknowledgment is received from the owner or authorized user. This is necessary due to personnel changes over time on a project or changing requirements on the length of time it is necessary to hold data. Consequently a newly assigned user manager may be unaware of file expirations set by a prior user manager.
7. Location tracking. Some very large sites may have multiple tape storage rooms. It is necessary to know what storage room the volume set is stored in as well as what the cost parameters are. Some storage may be an offsite fail safe site with longer time lags (possibly days!) to obtain the tape and incur financial charges in order to retrieve.
8. Import/Export controls. Record keeping of tapes and volume sets that are no longer on site is needed. Some items that need to be recorded are:
 - A. Shipment authorized by.
 - B. Sent to.
 - C. Expected to be returned by.
 - D. Volume set or tape reel identification.
 - E. Who to notify on the tape reel failing to return.

Some sites (ours included) need to deal with tapes that are exogenous to whatever system we are using and preserve the original creators identification of the tape. This requires that a tape management system function perfectly with only a visual identification entry or an alias entry that points to a visual tape identification.

An additional requirement is some form of inventory and tracking of tapes in one or more transitory tape areas. A particular problem with exogenous tapes is recording shipping and return information for follow up tracking months or years after the event. An additional problem with exogenous tapes arises over drive alignment problems and readability which also needs to be tracked.

9. Indexing. This is one of the more complex needs to describe since it is largely affected by the individual users modes of operation and accountability. Global indexing must be provided for the tape manager. Group and individual indexing must be provided for the user(s). Some items that need to be tracked:
 - A. Tape reels in use by reel and size in feet or meters.
 - B. Volume sets in use and component reels.
 - C. Starting data block number for each tape reel in a multi reel volume set.
 - D. Data set in use and volume set components or containment set.
 - E. Expired data sets pending disposition.
 - F. Expired data sets resolved by tape manager action and disposition.
 - G. Data set location on disk or in fast, slow or offsite tape archival.
 - H. Which data sets in a compressed containment volume set have expired and can be removed during the next compression cycle for the volume set.
10. Automation. Allow hooks so that the tape management software can be used to manage some form of automated tape storage retrieval system. This should include Volume set retrieval, mounting and verification.

11. Additional general comments.
 - A. Allow for multiple simultaneous tape managers accessing the tape management system database.
 - B. Allow for multiple user access and searches of the tape management system database.
 - C. Allow some number of user definable fields to be added to the database information kept by the tape management system and filled in by either exits to user provided code or system services.
 - D. Allow for batch reporting and offline reference lists of tapes and data sets.
 - E. DO NOT assume that the visual id will be the same as the tape label or that the visual id is one of the several numbers on the tape.
 - F. DO NOT require that a tape go through a long manual entry process to be used. It must be possible to mount and use a tape that is totally outside of the tape manager. Assume an operator intelligent enough to put the appropriate tape reel on a specified drive when directed! BUT LOG this event with appropriate notation.
 - G. DO NOT require that a labeled tape volume identification be its tape management reel number! Tape labels are defined by the users, tape reel ids normally come from the librarian system. Tapes must be mountable by either criteria.

ACCOUNTING

12. Accounting hooks. In order to support user billing and charge back many hooks to identify events need to be provided. For openers consider:

- A. Initial mount count.
- B. Retrieval from onsite storage cost.
- C. Retrieval from offsite archive cost.
- D. Continuation reel mount cost differing from initial mount cost.
- E. Storage costs for the volume set per reel per time period.
- F. Transaction costs for reading and writing a data block to tape.
- G. Vary the above by shift of the day, day of the week, week of month and month of the fiscal or calendar year.
- H. Vary the above on a user by user basis!
- I. Budget limit monitoring.
- J. Dismount or end of job logging to the users log file of errors and transaction counts as well as generating accounting entries. This must be kept disaggregate by either data set or device.

SCHEDULING

13. Job scheduling. Must be a three stage process! The current que structure is inadequate for tape management operations. Stage one is a set up stage where the jobs sit in an organizational que that allows the operators to:
- A. Retrieve and identify all input volume sets.

- B. Validate and allocate all reels of tape to make up the output volume sets.
- C. Allocate all scratch tapes needed as temporary work volume sets.
- D. Perform any ancillary support tasks like cleaning tapes, repairing or replacing tapes and verifying visual identifications of required tapes.
- E. Verify authorization to access or write the involved volume sets.

Once the set up stage is complete the job is released by the operator to the execute scheduler as stage two. (Note jobs that require no operator set up automatically release to the execute scheduler.)

The stage two scheduler needs to consider the following:

- A. Number of tape transports and type required to initiate the job. (This is the maximum number needed at any point in the job stream.)
- B. Execution time estimate.
- C. Memory requirements.
- D. Disk space requirements by device to execute.
- E. Recording of stage one events in the user log file.
- F. Recording of conflicts that cause the job to stall such as required drive in use, required volume set or reel of volume set in use, or required cpu to run on not available in the user log file.
- G. Recording oops events in the user log file such as operator shut down of required queues, devices, device conflicts or any other unusual event that affects the job status, such as instituting a failing tape reel recovery.

- H. Date and time dependencies for execution.
- I. Operator holds placed on the job. All holds and releases placed on the job by either the scheduling system or the operator need to be recorded in the log file.
- J. User defined completion dependencies. Support must be provided for the user to assign job identifiers for scheduling control to batch jobs and list multiple completion dependencies that release the job when any one of the dependencies is satisfied. For example release after jobs BUILDZ and CLEANZ end normally or after FORCETWO ends.

The scope of job identifier checking will also need to be extended to other members of the group. Users must be able to add user defined completion dependencies at any time after submission of the job.

Stage three processing is a job rundown and break down of mounted tape volume sets. During this phase log files are routed and printed (including copies to multiple users) and generation of instructions on where to return the tapes to for the operators and clerks. Additionally the tape management system would update the usage log for the tape volume sets used and schedule any cleaning or duplication of critical volume sets based on tape wear. Any tapes that are to be returned to an exogenous temporary storage location would be identified and instructions generated to accomplish this.

RELIABILITY/INTEGRITY

- 14. Cleaning cycle. Tape needs to be exercised and cleaned when stored for long periods of time. When one is required to keep data for years it is imperative that a cleaning and read validation program be kept up. The following are minimal requirements:

- A. Generation of a cyclic cleaning list.
- B. Status of cleaning, I.E. when done and by whom.
- C. Read validation pass list. (Read validation pass is optional)
- D. Status of validation, I.E. when done and by whom.
- E. Recording in the archival database.
- F. Positive control, I.E. requires a person to sign off rather than assuming it will be done.
- G. Tape reel failure recovery. Provide some criteria to indicate that the data on a particular tape reel is becoming unsafe and should be recovered as well as recovery procedures. This is not simply a global setting but must be tailorable by individual users and data sets. This mechanism will need to be automatically initiated by the appropriate file system when enough disk or tape reel errors have been detected. The recovery procedure should be able to replace a bad tape reel in a volume set or create an entirely new replacement volume set as determined by the user. The recovery procedure will also need to update the definition of a tape shadow set if the volume set being regenerated is a member of a tape shadow set.

- 15. Failure tracking. This is a major reliability item that must provide information concerning volume set or device failures (including soft parity errors) to:

- A. The submittor.
- B. The appropriate operator.
- C. The tape manager.
- D. The appropriate operations and user supervisors.

- E. The maintenance support group (including NON DEC field service organizations).
 - F. The user end of run log file summary.
 - G. Failing tape reels or devices that require failing volume recovery to be initiated for the volume set or tape reel.
16. Event Logging. All significant events in the tape management system MUST be logged to the affected user, operator, supervisors and tape manager. Some events to log are:
- A. Initial reel of a volume set mount.
 - B. Continuation reel of a volume set mount.
 - C. Significant parity errors.
 - D. Leaving a known bad block on a reel in a volume set that has XOR block recovery processing enabled.
 - E. Billable transactions such as Read QIO's and Write QIO's.
 - F. Switching to an alternate member volume set of a shadow set including why and at what point in the original volume set.

The event logs need to contain a date/time stamp, error count for the drive and tape reel, tape reel identification, the volume status, the data block number the error occurred in for the reel relative to both the beginning of the data set and the tape reel, the operator identification (name or unique number) and operator response.

17. OOPS recovery. Some events are relatively catastrophic and need some form of specific recovery and intervention by the user, operator or tape manager. These events include at least the following:

- A. A tape reel allocated to more than one volume set. (not to be confused with a tape reel in a containment volume set that contains several data sets)
 - B. A volume set or tape reel that was improperly mounted and verified by the operator and destroyed by improperly being written on.
 - C. A tape reel that was eaten or damaged by the drive.
 - D. A tape reel or volume set lost in transit.
 - E. A catastrophic read failure on a non shadowed volume set. Possibly on the second failure after a reposition restart with the tape reel on a different tape transport.
18. Data integrity issues. In addition to the normal treatment of bad tape reels etc. Support should be provided for redundant storage of critical data in one or more TAPE VOLUME SHADOW SETS as well as allowing user specification of XOR recovery block generation into each volume set. (XOR generation is not to be limited to volume sets that are members of a shadow set) The shadow set members would be linked to the master volume set and inherit any modifications to expiration criteria when created. The shadow sets would have independent management of expiration criteria after creation.

The concept of tape shadow set would work in much the same fashion as a disk shadow set but would not require a one for one correspondence for each reel. All volume sets that comprise a tape shadow set should be required to have the same physical blocking and XOR block structure.

On fail over to the use of an alternate member of the shadow set the fail over would be complete at the failing block. The alternate volume set would be positioned to the restart data block on the appropriate tape reel and the job would continue. Note that once the data block being read from the alternate volume set member of the shadow set became greater than the beginning block number for the next reel of the

original volume, set the original volume set could be used as a recovery volume set for the alternate shadow set member currently in use. NOTE this capability would also need to apply to a containment volume set with multiple data sets.

SECURITY

19. Security support must be provided for:
 - A. Physical tracking (sign out/sign in of tapes) of volume sets.
 - B. Access validation similar to the resource pooling description.
 - C. Modification validation similar to resource pooling description.
 - D. Access and modification controls to a finer granularity than the whole file. Possibly to a two dimensional matrix defined by data field and record identifiers.
 - E. Declassification of data set procedures and authorization to declassify.

ARCHIVAL NEEDS

20. Archival capability. Some form of data set archival is needed in a tape management system. Note that for this discussion the source data set can reside on EITHER a TAPE DATA SET or a DISK FILE. If the source data set resides in a disk file some record of the migration and resultant state must be placed into the disk file system until the user deletes the entry or recovers the

data set to disk. NOTE that deletion of the disk entry does not automatically delete the archived data set from the tape management system, A separate deletion is required to delete the archived data set. However a disk file delete may indicate that a file should be moved from the rapid recovery archival area to a slower long term archival area.

Under some set of predefined conditions a data set will be defined as inactive and will begin a migration process to less active storage. (This is normally restricted to disk files but I feel that inclusion of tape volume sets into the record keeping facility will significantly reduce the management problems of finding data.) The triggering process will need to consider how long the files have been inactive, free space shortages on disk drives, needs that the system manager may have to temporarily reallocate the disk space or just that the user wants the disk files moved off of disk to archival storage.

When a data set has been archived, any user attempt to access the data set will result in an automatic reload of the data set from archival storage to either the original disk file or a new tape volume set and release of the short term archival entry. Additionally some explicit facility is needed to allow extraction of a contained data set from a containment volume set to either a tape volume set or a disk file in some directory. Additionally support is needed to delete data sets from containment volume sets and modify their attributes.

Once an archival process is triggered the following events can happen to the data set:

- A. Place it in the waste can (to avoid upsetting Apple who owns the trash can ICON!) and delete it. Placing a tape volume set in the waste can releases the component tape reels to the users available tape reel resource pool or to the groups pool or to the tape managers pool.

- B. Assign a new expiration criteria and abort the migration.
- C. Migrate the data set into fast recovery archival and assign an expiration criteria that will move the data set from fast recovery archival to either slow recovery archival storage or the waste can. This movement of data to the waste can after removal from tape or disk may be used to provide a short term recovery window for all files and a longer term recovery for valuable files. This methodology would be useful in having the tape management system perform much of the record keeping for a backup system.
- D. Migrate the data set into slow recovery archival.

Note files do not automatically enter the waste can, a positive user response is required or failing a response within a set time limit the tape manager can override. If the tape manager overrides, a user notification must be given that the files were deleted by the tape manager.

During the archival process data sets that are migrating off of disk or tape to archival storage should be stored in containment volume sets that are members of a shadow set.

All archival containment volume sets should be periodically purged of deleted data sets and recompressed automatically by the tape management system. This process will also need to update the record keeping entries in the tape management system database.

For archival purposes it is frequently desirable to reduce the physical number of reels that data is stored on. Some facility to concatenate many discrete data sets each occupying a single volume set into a containment volume set consisting of several reels and containing several data sets is needed. The value of this facility is obvious in archiving the 75 individual partial reels that represent one week of data input for one of MSA's processing groups. This compressed file consists of 10 reels of data at 6250 bpi and 32000 byte blocks. This support facility must not require placing

the data sets to be concatenated on disk as an intermediate step. The concatenation facility must be able to accept mixed disk and tape inputs while preserving all attributes of the individual data sets.

RESOURCE POOLING

- 21. Resource pooling. Since tapes and drives represent a finite resource, most users require infinite resources and inevitable conflict arises. A way to solve this is to provide some form of quota management for Disk space, Disk drives (for mountable packs), Tape reels and Tape drives. The quota management system should provide for quotas assigned by:
 - A. Anarchy (no restrictions).
 - B. Group (paralleling disk UIC based affiliations).
 - C. Owner (paralleling disk UIC based affiliations).
 - D. World (paralleling disk UIC based affiliations).
 - E. System (paralleling disk UIC based affiliations).
 - F. Tape ACL (definition of some arbitrary collection of users and or attributes).
- 22. A BAD tape pool. Frequently tape reels that go bad are bad only at the load point and are repairable. The value of this concept may be insignificant with the advent of cartridge tapes like the TK50. Cartridge reels are generally not repairable due to the lack of spare parts, though some creative users may try to scavenge parts from multiple cartridges to repair one cartridge. Also the wear pattern on a TK50 cartridge is not as intensely focused on one physical location. Some items to consider in providing a bad tape pool are:

- A. Retirement of the tape reel id.
 - B. Replacement of the tape reel with another of the same id.
 - C. Repair of the tape reel.
 - D. Release of the tape reel from the bad tape pool to the available tape pool of the tapes owner or to the tape managers pool.
 - E. Replacement of a bad tape reel in a write volume set and re scheduling the job for execution.
 - F. Replacement of a bad tape reel in a write volume set by copying the data written successfully to the current bad tape reel onto one or more replacement tape reels and retrying the failed write operation. This requires an extra tape transport to perform as well as modification of the volume set component tape reel definition. This is not affected by the existence of a tape volume shadow set.
23. Free tape pool. The tape management system needs to provide support for tracking and reassignment of available tape reels by:
- A. Individual user.
 - B. Group relationship by both project and uic.
 - C. The tape manager.
 - D. Scratch tape pool
 - E. Any other definitive relationship that can be defined by a process similar to ACL's.
 - F. Allow authorized "owners" to reassign tape reels into some other group both permanently or temporarily.
24. Scratch tape pool. The tape management system should provide a shareable resource pool of tape reels that can be freely used by any one for the duration of a job step. Additionally some "grace period" of time defined by the tape manager should exist, where the user could hold onto the tape reels temporarily. This is useful to allow retention of data across jobs or a recovery point to restart processing at in the event of error or machine failure. This concept of scratch tape management would be very useful if implemented in a similar fashion for disk volumes, a very good example of this use would be sort work files.

VAX System SIG Committee List

As of August 25, 1986

Osman K. Ahmad - Large Systems Integration Working Group
Association of American Railroads
Technical Center, Research and Test Department
3140 South Federal Street
Chicago, IL 60616

Joe Angelico - Assistant Symposium Coordinator
US Coast Guard CCGD8(DT)
Hale Boggs Federal Building
500 Camp Street, New Orleans, LA 70130

Elizabeth Bailey - Volunteer Coordinator
222 CEB
Tennessee Valley Authority
Muscle Shoals, AL 35660

June Baker - Advisor
Computer Sciences Corporation
6565 Arlington Boulevard
Falls Church, VA 22046

Joe L. Bingham - Librarian
Mantech International
2320 Mill Road
Alexandria, VA 22314

Bob Boyd - Commercial Working Group
GE Microelectronics Center
MS 2P-04
Post Office Box 13409
Research Triangle Park, NC 27709

C. Douglas Brown - Security
Sandia Labs
Division 2644
P.O. Box 5800
Albuquerque, NM 87185

Jim Caddick - VAXcluster
General Datacom
Strait Turnpike
Middlebury, CT 06762-1299

Jack Cundiff - Symposium Coordinator
Horry-Georgetown
Post Office Box 1966
Conway, SC 29526

Tom Danforth - Handout Editor
Woods Hole Oceanographic Institute
Woods Hole, MA 02543

Barbara Dow-Pleines - MicroVAX Working Group
Magic One
1971 Mount Pleasant Road
San Jose, CA 95148
(408) 238-0861

Jim Downward - Migration and Host Development, VAXintosh Working Group
KMS Fusion Incorporated
3941 Research Park Drive
Ann Arbor MI 48106

Jane Furze - Campground
3830 West Cochise
Phoenix, AZ 85064

Dennis Frayne - Real Time/Process Control Working Group
McDonnell Douglas
5301 Bolsa Avenue
Huntington Beach, CA 92646

Carl E. Friedberg - Internals Working Group
In House Systems
165 William Street
New York, NY 10038

Don Golden - Communications Committee Representative
c/o Shell Oil Company
Westhollow Research Center
Post Office Box 1380, Room D2132
Houston, TX 77001

B. Hancock - Network Working Group
Dimension Data Systems, Incorporated
2510 Limestone Lane
Garland, TX 75040
(214) 495-7353

Jeffrey S. Jalbert - Historian
J C C
Post Office Box 381
Granville, OH 43023
614-587-0157

Ray Kaplan - MicroVAX Working Group
Pivotal, Incorporated
Post Office Box 32647
Tucson, AZ 85715-32647
(602) 886-5563

Lawrence J. Kilgallen - Newsletter Editor
Box 81, MIT Station
Cambridge, MA 02139-0901

Margaret Knox - Chair
Computation Center
University of Texas
Austin, Texas 78712

Art McClinton - Advisor
MITRE
1820 Dolley Madison Boulevard
McLean, VA 22102

Ross W. Miller - Vice Chair and Working Group Coordinator
Online Data Processing, Inc.
N 637 Hamilton
Spokane, WA 99202

Mark D. Oakley - System Improvement Request
Battelle Columbus Labs
Room 11-6-008
505 King Avenue
Columbus, OH 43201-2669

Eugene Pal - Multiprocessor Working Group
US Army
CAORA (ATOR-CAT-C)
Fort Leavenworth, KA

Susan Rehse - System Management Working Group
Lockheed Missiles
3251 Hanover Street
Palo Alto, CA 94301-1187

Bob Robbins - Advisor
Array Computer Consultants
5364 Woodvale Drive
Sarasota, FL 33582

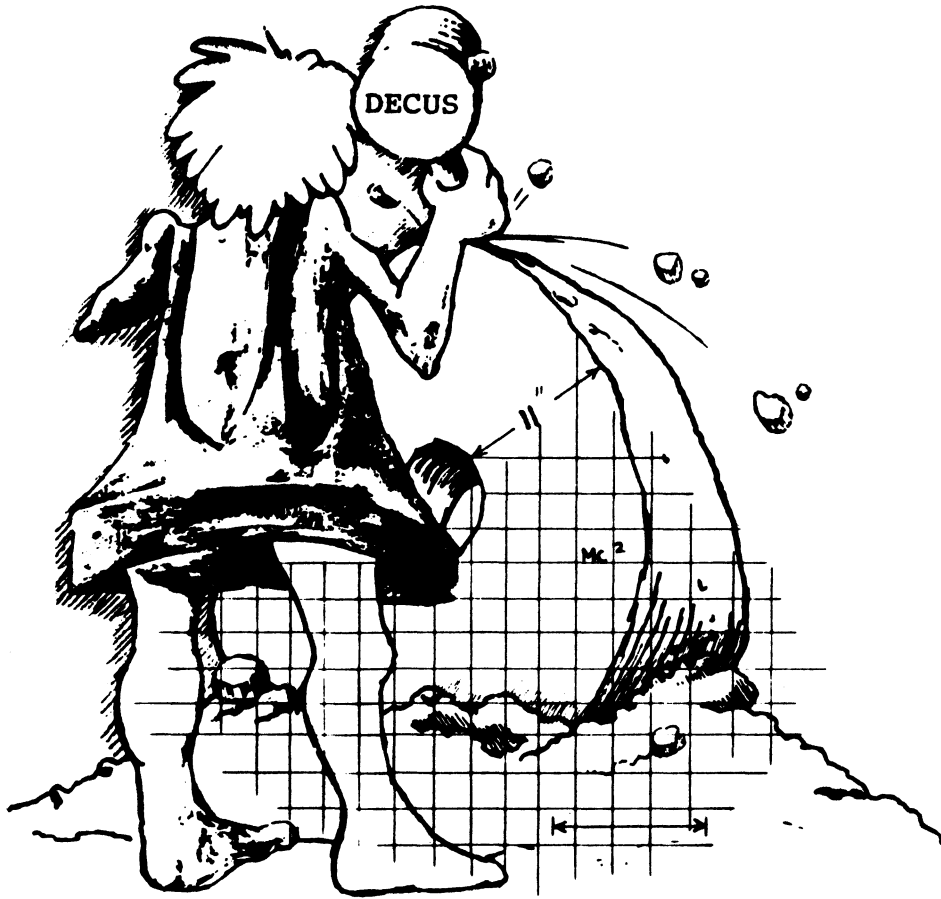
Larry Robertson - Real Time/Process Control Working Group
Bear Computer Systems Inc.
5651 Case Avenue
North Hollywood, CA

David Schmidt - LUG Coordinator, Hardware Working Group
Management Sciences Associates
5100 Centre Avenue
Pittsburgh, PA 15232

Al Siegel - Advisor
Battelle Memorial Institute
505 King Avenue
Columbus, OH 43201-2693

D. Slater - Artificial Intelligence Working Group
Institute for Defense Analysis
1801 North Beavregard Street
Alexandria, VA 22314

LIBRARY



DECUS PROGRAM LIBRARY

NEW LIBRARY PROGRAMS AVAILABLE FOR THE PROFESSIONAL-300 SERIES OF COMPUTERS

DECUS NO: PRO-158 **Title:** Bonner Labs RUNOFF - Pro-350/380 **Version:** BL 8.1, March 1986

Author: John Clement, Rice University

Submitted by: Jack Wenrick, BF Goodrich R&D, Brecksville, OH **Operating System:** P/OS V2.0A **Source Language:** MACRO-11 **Memory Required:** 24,192 Words **Keywords:** RUNOFF, Text Formatting

Abstract: This is a PRO version of Bonner Labs RUNOFF, the best version of RUNOFF I have used. For a more complete description of RUNOFF see DECUS Program No. 11-703.

Notes: Sources are not available with this program. However, DECUS Program No. 11-703 contains complete sources.

Sources not included.

Media (Service Charge Code): Two RX50 Diskettes (JB) **Format:** FILES.11

August 25, 1986

DECUS NO: PRO-159 **Title:** Space Invasion for Pro-350/380 **Version:** 1986

Submitted by: John M. Crowell, Crowell, Ltd., Los Alamos, NM **Operating System:** PRO RT-11 V5.3 **Source Language:** MACRO-11, FORTRAN 77 **Memory Required:** 20KB **Hardware Required:** Pro-350, Pro-380, PDP-11 or LSI-11 with floating point instruction set and VT100 or VT220 terminal. **Keywords:** Games

Abstract: Space Invasion for the Pro-3xx is a complete rewrite of the original VT52-based game. It is a real-time interactive game simulating the popular arcade game. Written in FORTRAN 77, it takes advantage of the native hardware on the Professional-300 series of computers. It can also be run on PDP-11 systems with the floating-point instruction set and a VT100 or VT220 terminal (preferably at 9600 baud or greater). The program runs entirely too fast on the 11/73 and 11/83, so a foreground program DELAY.REL is also included to make the computer twiddle its thumbs.

Notes: RT-11 V5.1 or later is required.

Media (Service Charge Code): One RX50 Diskette (JA) **Format:** RT-11

August 25, 1986

NEW LIBRARY PROGRAMS AVAILABLE FOR THE VAX/VMS FAMILY OF COMPUTERS

DECUS NO: VAX-180 **Title:** Parallel Library V2 **Version:** V2, May 1986

Submitted by: Digital Equipment Corporation **Operating System:** VAX/VMS V4.3, 4.4 **Source Language:** MACRO-32, VAX-11 FORTRAN **Memory Required:** 5KB **Keywords:** Tools - Software Development

Abstract: The Parallel Library routines assist in writing a parallel application by implementing many of the functions commonly required for parallelism. These functions include establishing shared data and executable code regions, creating and deleting subprocesses and implementing barrier synchronizations and critical sections. Included in the kit is a sample parallel program whose comments describe many of the standard parallelism concepts and suggested VAX/VMS solutions.

Notes: Requires a V4.X version of VAX/VMS

Media (Service Charge Code): 600' Magnetic Tape (MA) **Format:** VMS/BACKUP

August 25, 1986

DECUS NO: VAX-182 **Title:** SNOOP **Version:** V4.1, April 1986

Submitted by: R. D. Brownrigg, Applied Mathematics Division, DSIR, Wellington, New Zealand **Operating System:** VAX/VMS V4.0, 4.1, 4.2 **Source Language:** VAX-11 BASIC **Memory Required:** 300KB **Hardware Required:** VT100 compatible terminal **Keywords:** Terminal Management

Abstract: SNOOP will interactively display to a VT52 or VT100 terminal the state of processes on a VMS system, updating the display at regular intervals. Items displayed initially are the username, terminal name, image name, CPU time, and process state, with the option of dynamically adding one more item from a list which corresponds essentially to the information available from the \$GETJPI system call.

Processes displayed can be system processes only, user processes only, or both types, up to 43 being able to be displayed simultaneously (or 67 on a VT100 terminal). Privilege also has a bearing on which processes are available to be displayed.

Restrictions: Requires group or world privilege

Media (Service Charge Code): 600' Magnetic Tape (MA) **Format:** VAX/ANSI

August 25, 1986

DECUS NO: VAX-183 **Title:** JUICER ODS-2 Disk Compressor **Version:** V01-011, March 1986

Submitted by: Michael N. LeVine, Naval Weapons Center, China Lake, CA **Source Language:** MACRO-32 **Memory Required:** VAX/VMS V3.7, 4.X **Keywords:** Utilities - Disk - VMS

Abstract: The JUICER package of programs and command files is provided to the system manager to allow him to monitor VAX/VMS ODS-2 disks for disk and file fragmentation and to do such compression as might be needed. The package is made up of four parts:

- . JUICER to do on line disk compression.
- . FRAG to monitor disk fragmentation.
- . FILE to monitor and optionally compress fragmented files.
- . DIR to make a map of disk directory structure and its file/block usage

JUICER is an in-place disk compression utility for VAX/VMS ODS-2 disks suffering from excessive fragmentation. This program, within limitations, attempts to move portions of files from the high end of the disk to any unused areas (fragments) at the low end, freeing up larger contiguous free areas at the high end.

FRAG is run on a disk to see how badly the target disk free space is fragmented, giving a histogram of fragmented areas by size, and a calculated measure of the disk free space fragmentation.

FILE scans all the file headers on the target disk and outputs two list files, one containing a list of the 100 files having the most retrieval pointers in use, and the second being a matrix of file size versus number of pointers in use. The command file CONTIG is used which reads one of the list files produced by FILE and running interactively with the user, converts the listed files from fragmented to contiguous.

The command file DIR scans a target disk and creates an output file DIRECTORY.MAP containing a graphical output showing the on disk directory structure, with a notation for each directory showing the number of files and blocks contained therein.

Restrictions: No Volume Sets

Media (Service Charge Code): 600' Magnetic Tape (MA) **Format:** VAX/ANSI

August 25, 1986

NEW LIBRARY PROGRAMS AVAILABLE FOR THE VAX/VMS FAMILY OF COMPUTERS

DECUS NO: VAX-184 **Title:** DRAWTREE **Version:** V1, December 1985

Submitted by: Robert Donnenberg, Lear Siegler Avionic Systems Corp., Florham Park, NJ **Operating System:** VAX-VMS V4.1 **Source Language:** FORTRAN 77 **Hardware Required:** At least one VT100 compatible terminal per site. **Keywords:** File Management

Abstract: This submittal includes a new version of the DRAW-TREE utility. This utility produces a tree structure oriented display of the directory structure beneath a given directory spec. The display is produced using VT100 special graphics characters. This utility is essentially the same as that previously released by DECUS, but is MUCH faster and has several added features. Also included is the program CVTREE, which converts the VT100 special graphics characters in a DRAWTREE output file to printable text. Documentation for both programs, as well as sample output are also included.

Notes: Program requires VMS V4.1 as it uses many VMS specific system calls.

Sources not included

Media (Service Charge Code): 600' Magnetic Tape (MA) **Format:** VMS/BACKUP

September 15, 1986

DECUS NO: VAX-186 **Title:** The MLR MACRO Language **Version:** Spring 1986

Submitted by: Rodrick A. Eldridge, Iowa State University, Ames, IA **Operating System:** VAX/VMS V4.2 **Source Language:** MACRO-32 **Keywords:** MACRO, Structured Languages/ Programming

Abstract: The MLR MACRO Language is a set of macros which implement structured programming in MACRO-32.

These include:

- . IF-THEN-ELSE IF-ELSE
- . CASE
- . LOOP-END
- . WHILE
- . REPEAT
- . AND OTHERS

Documentation is included on tape. Also included on the tape are two terminal I/O routines and a quadword math routine, all written in MLR to serve as examples.

Media (Service Charge Code): 600' Magnetic Tape (MA) **Format:** VAX/ANSI

August 25, 1986

DECUS NO: VAX-187 **Title:** RSTSOPEN **Version:** V3.002, May 1986

Submitted by: Victor Lindsey, VL Systems, Inc., Irvine, CA **Operating System:** VAX/VMS V4.1 **Source Language:** MACRO-32 **Memory Required:** 7700 bytes **Keywords:** Tools - Applications Development, BASIC

Abstract: RSTSOPEN is a series of MACRO-32 subroutines used to augment the OPEN statement of any VMS BASIC program through the use of its USEROPEN clause. With it, a user or programmer can append various qualifiers directly onto the filename for processing by RSTSOPEN prior to doing the OPEN itself. Originally modeled after the way qualifiers

are used under the PDP-11 operating system RSTS/E. RSTS-OPEN provides the programmer with easy access to a wide variety of features available to RMS under VMS, as well as providing a partial emulation of qualifiers found only on the RSTS/E environment.

Items like creation date, multi-buffer count (data caching), protection code, and ownership are easily handled by appending a qualifier, such as /GLOBAL_BUFFER=5 (used to establish 5 global buffers on an OPEN). Furthermore, an extensive amount of information is returned concerning the file just OPENed, thus making up for the lack of a SYS(CHR(12%)) call (return info on last file OPENed) that is found only on RSTS/E. Powerful error handling and message reporting permits easy diagnosis of obscure errors such as "%RMS-E-ENQ ENQ system service request failed".

Included with the distribution is an extensive help file suitable for inclusion in the standard HELP facility of VMS, examples of its use in a BASIC program, and examples of its inclusion in shareable libraries called by BASIC programs.

Restrictions: /VERSION_LIMIT known not to work properly. Everything else is fine. Program requires VMS V4.1 or later.

Media (Service Charge Code): 600' Magnetic Tape (MA)
Format VMS/BACKUP

September 15, 1986

DECUS NO: VAX-190 Title: TDE: Table Driven Editor
Version: V2.00, March 1986

Author: Ian Stewart, Municipal Electricity Dept., Wellington, New Zealand

Submitted by: Richard Naylor, Municipal Electricity Dept., Wellington, New Zealand
Operating System: VAX/VMS V4.1
Source Language: MACRO-32, VAX-11 BASIC
Software Required: EXE and OBJ files included, so VAX BASIC compiler not required.
Hardware Required: Only runs on VT52 and VT100 upwards compatible terminals, (i.e. works on VT200 terminals).
Keywords: Editors

Abstract: TDE is a table driven editor for editing sequential, relative and indexed-sequential files. It allows users to examine, insert, modify and delete records on a field by field basis.

TDE is a full-screen editor which is usable on any VT52, VT100 or VT200 upwards compatible terminal. It can be used for editing any file which has fixed-length records and fields which are fixed in position, size and data type. It provides some degree of data validation and an optional audit trail, making it highly suitable as a data-entry tool. It can be used across DECnet to edit files stored on remote nodes.

TDE supports all the standard VAX data types: signed and unsigned byte, word and longword integers, signed quadword integers, single and double precision floating point fields (as well as G-Float and H-Float), packed decimal fields, fixed length string fields.

Also supported are VMS quadword format absolute time fields (as per \$ASCTIM), 1 byte logical fields, all common numeric string data types (e.g. left separate and left overpunched sign, right separate and right overpunched sign, unsigned and zoned

sign numeric string fields), EBCDIC fields, and 2 byte data fields.

Packaged with TDE are two other table driven utilities, TDA and TDR. These utilities use the same format table file as TDE. TDA is a table driven audit report generator for creating audit reports from the log and audit files generated by TDE. This allows you to see which users have changed which records, at what time and from which terminal. TDR is a table driven report generator for creating simple columnar reports. Column totals can be calculated for some numeric fields.

Full RUNOFF-source documentation is included, as well as an INSTALL command procedure to automatically install the three utilities and their associated files.

Release Notes are distributed with each order.

Media (Service Charge Code): 600' Magnetic Tape (MA)
Format VMS/BACKUP

September 15, 1986

DECUS NO: VAX-191 Title: MTU Version: April 1986

Submitted by: E. Langner, Hahn-Meitner Institut
Operating System: VAX/VMS V4.2
Source Language: PASCAL
Memory Required: 16 KB
Software Required: PASCAL for new compilation
Hardware Required: One tape drive
Keywords: Utilities - Tape

Abstract: MTU is a program for accessing magnetic tapes in a physical mode. It's able to compare, copy, read, write and dump tapes or part of tapes without interpreting the data.

It's possible to copy tapes with only one tape device if there is enough disk space to buffer the content of the tape into a file. Another feature is to copy a tape to a set of tapes (e.g. if they have a different density).

The user interface is based on DCL command format.

Release Notes are distributed with each tape.

Notes: This program requires VAX/VMS V4.X. To execute the program the privilege PHY_IO is necessary.

Media (Service Charge Code): 600' Magnetic Tape (MA)
Format VMS/BACKUP

September 15, 1986

NEW LIBRARY PROGRAMS AVAILABLE FOR THE PDP-11 COMPUTER FAMILY

DECUS NO: 11-839 Title: Symposium Tape from the RT-11
SIG, Spring 1986, Dallas Version: Spring 1986

Submitted by: R. W. Barnard, Sandia National Labs, Albuquerque, NM
Operating System: RT-11 V5 or later
Source Language: MACRO-11, FORTRAN 77, FORTRAN IV, C
Memory Required: Various, specified in submission
Software Required: Will be specified, if required.
Hardware Required: Special requirements will be specified in the submissions.
Keywords: FORTRAN, Symposia Tapes - RT-11

Abstract: The symposium swap tape from the RT-11 SIG contains twenty packages in subdevice format. The tape includes an annotated directory TAPDIR.TXT, and instructions for RT-11 and RSTS users on recovering files from subdevices. The file TAPDIR.TXT includes a summary, cross-reference and index section. The tape contains the following submissions:

- . The portions of the March 1986 update of the DECUS C programming language applicable to RT-11.
- . Update of the TSXLIB system service calls for FORTRAN programmers who use TSX+ (*).
- . Utilities for easily connecting to subdevices.
- . Program to translate OTS error numbers to text.
- . Improved version of UCL+ user command linkage (version 7.54A).
- . A FORTRAN 77/RT OTS update kit, routines for writing virtual arrays to disk, diagnostic overlay handler.
- . An object file disassembler for use on FORTRAN IV and FORTRAN 77 modules.
- . An emulator for EIS, FIS and FPU microcode.
- . Utilities for compressing load maps, patching directories, searching for strings in a file. A VT100 "pocket calculator".
- . Program for "filtering" unsolicited smart-modem comments from a terminal emulator program.
- . System to add executable modules to BASIC programs.
- . An "experimenter-friendly" experiment development library.
- . Ambrose Bierce's "Devil's Dictionary".

No guarantees are made as to the completeness, usability or quality of the programs on the tape, and the material has not been checked or reviewed.

(*TSX+ is a trademark of S & H Computer Systems, Inc.

Restrictions: If any, they will be specified in the individual submissions.

Media (Service Charge Code): 2400' Magnetic Tape (PS)
Format RT-11

August 25, 1986

LIBRARY PROGRAMS AVAILABLE FOR THE PDP-11 COMPUTER FAMILY

DECUS NO: 11-838 Title: SMARTMAILER for RSTS/E
Binary Version Version: V1.1, July 1986

Submitted by: Digital Equipment Corporation
Operating System: RSTS/E
Source Language: BASIC-PLUS2
Keywords: Business Applications, Mail

Abstract: The SMARTMAILER for RSTS/E software is an application used to create and maintain mailing lists of names and addresses, and generate address labels.

Mailing List Contents

- . Mailing lists contain packed addresses with up to 327 characters, each composed of a name, three address lines, city/town, state/country, zip code, two separate identifiers, a counter, a comment and up to six user-defined categories.
- . Category information is user-defined and can be different for each mailing list.
- . Each mailing list can have associated sublists.
- . Each mailing list can be set up differently.

Major Functions

- . Update - mailing lists can be created and maintained by adding, removing, and changing individual address entries.
- . Display - any or all addresses, category definitions, or sublist definitions can be displayed on a video screen.
- . Listings - full addresses, category definitions, and sublist definitions can be printed (or written to a disk file).
- . Labels - any mailing list or sublist can be printed on a variety of labels.
- . List Processing Interface - a standard list document file can be generated for use with Digital Equipment Corporation word processing systems to produce personalized letters.

Features

- . User Interface - all user interaction is menu or form driven.
- . Label Printing - various parameters for label printing can be defined to meet specific needs.
- . Category Information - up to six categories of related information can be stored for each mailing list.
- . Sublists - addresses can be selected from mailing lists by defining requirements on specific address fields.
- . Sorting - all lists may be sorted by any address field (except comments) before being printed as listings or labels.
- . Presort - SMARTMAILER for RSTS/E can presort U.S. addresses to take advantage of U.S. Postal rules (in effect in July 1979), which allow a reduced postage rate on First Class Mail.

Sources not included.

Media (Service Charge Code): Two RX02 Diskettes (LB)
Format: RT-11, 600' Magnetic Tape (MA) Format: DOS-11

September 15, 1986

DECUS NO: 11-839 Title: SMARTMAILER for RSTS/E
Version: V1.1, July 1986

Submitted by: Digital Equipment Corporation
Operating System: RSTS/E
Source Language: BASIC-PLUS2
Keywords: Business Applications, Mail

Abstract: The SMARTMAILER for RSTS/E software is an application used to create and maintain mailing lists of names and addresses, and generate address labels.

Mailing List Contents:

- . Mailing lists contain packed addresses with up to 327 characters, each composed of a name, three address lines, city/town, state/country, zip code, two separate identifiers, a counter, a comment and up to six user-defined categories.
- . Category information is user-defined and can be different for each mailing list.
- . Each mailing list can have associated sublists.
- . Each mailing list can be set up differently.

Major Functions:

- . Update - mailing lists can be created and maintained by adding, removing, and changing individual address entries.
- . Display - any or all addresses, category definitions, or sublist definitions can be displayed on a video screen.
- . Listings - full addresses, category definitions, and sublist definitions can be printed (or written to a disk file).

- . Labels - any mailing list or sublist can be printed on a variety of labels
- . List Processing Interface - a standard list document file can be generated for use with Digital Equipment Corporation word processing systems to produce personalized letters

Features:

- . User Interface - all user interaction is menu or form driven.
- . Label Printing - various parameters for label printing can be defined to meet specific needs.
- . Category Information - up to six categories of related information can be stored for each mailing list.
- . Sublists - addresses can be selected from mailing lists by defining requirements on specific address fields.
- . Sorting - all lists may be sorted by any address field (except comments) before being printed as listing or labels.
- . Presort - SMARTMAILER for RSTS/E can presort U.S. Postal addresses to take advantage of U.S. Postal rules (in effect in July 1979), which allow a reduced postage rate on First Class Mail.

Media (Service Charge Code): Three RX02 Diskettes (LC) Format: RT-11, 600' Magnetic Tape (MA) Format: DOS-11 August 25, 1986

DECUS NO: 11-847 **Title:** RTMULTI and Addons **Version:** V2.2, July 1986

Author: Fermilab Computing Dept.

Submitted by: Fermi National Accelerator Laboratory, Batavia, IL **Operating System:** RT-11 V4.0 or greater **Source Language:** MACRO-11, FORTRAN IV **Software Required:** FORTRAN compiler **Hardware Required:** Jorway 411 Branch Driver, Tektronix 4010, DR11-C useful, DR11-W useful **Keywords:** Physics Applications

Abstract: For over ten years, Fermi National Accelerator Laboratory in Batavia Illinois has developed and used the software package RTMULTI for use in high speed CAMAC data acquisition for high energy physics experiments. This submission includes many of the most useful developments to RTMULTI as well as the latest version of RTMULTI itself.

RTMULTI, originally created at Caltech and extensively developed by Fermilab is a CAMAC based data acquisition and monitoring system using the Jorway 411 branch driver. Histogramming and analysis of the acquired data can be formatted interactively to provide graphics output to Tektronix 4010 type displays. Over 150 experiments and collaborations have used MULTI all over the world.

Restrictions: Requires overlaying

Media (Service Charge Code): 2400' Magnetic Tape (PA) Format: RT-11 August 25, 1986

DECUS NO: 11-848 **Title:** PRM-11 PASCAL/RXS **Version:** March 1986

Submitted by: Norbert Herbold, Spanner-Pollus GmbH, D-6700 Ludwigshafen, West Germany **Operating System:** RSX-11M V4.1 **Source Language:** PASCAL/RXS, MACRO-11

Memory Required: 1KW plus RMS-11 + user code **Software Required:** PDP-11 Record Management Service, PDP-11 PASCAL/RXS V1.0. **Keywords:** PASCAL

Abstract: PRM-11 is a set of routines written in PDP-11 PASCAL/RXS (with an additional assembly language module) to interface user programs written in PASCAL to RMS-11. This is simply a conversion of previous DECUS program Nos. 11-479 and 11-691 (by Keneth G. Tibesar and Doug Bliss) from PASCAL to PASCAL/RXS V1.0.

The package provides high level interface commands and keywords implemented through externally defined procedures to create and allow access to all RMS file types (sequential, relative and indexed). PRM routines are called by the user, which in turn call the required RMS routines. The PRM routines are linked at task build time with the user code.

Restrictions: Implemented and tested on RSX-11M V4.1. Will also run on VAX/VMS with VAX-11 PASCAL in compatibility mode, except that shared files may not be opened with write access, due to restrictions of the compatibility mode emulation.

Media (Service Charge Code): One RX01 Diskette (KA) Format: FILES-11, 600' Magnetic Tape (MA) Format: FILES-11 MEDIA FORMAT MUST BE SPECIFIED ON ORDER FORM

September 15, 1986

NEW LIBRARY PROGRAMS AVAILABLE FOR THE DECsystem-10 FAMILY OF COMPUTERS

DECUS NO: 10-SP-11 **Title:** Symposium Tape from the TOPS-10 SIG, Fall 1985, Anaheim **Version:** Fall 1985

Submitted by: Jack Stevens, The Gillette Company **Operating System:** TOPS-10 **Source Language:** MACRO-10 **Memory Required:** Various **Keywords:** Symposia Tapes - TOPS-10, Utilities - TOPS-10

Abstract: The TOPS-10 Fall 1985 DECUS Symposium Tape comprises software contributed by users at the Anaheim 1985 DECUS Symposium. It consists of submissions by Pima Community College (tape and other utilities and tools).

Notes: Correction files only to Digital Equipment Corporation sources are included.

Complete sources not included.

Media (Service Charge Code): 600' Magnetic Tape (MS)

September 15, 1986

DECUS NO: 8-938 **Title:** VISTA EDITOR **Version:** April 1986

Author: Stuart DeWar

Submitted by: Wally Kalinowski, Aerospace Corp., Los Angeles, CA **Operating System:** OS/78, OS/8 **Source Language:** PAGE8 **Memory Required:** 12KW **Keywords:** Editors

Abstract: VISTA is a full screen editor which allows for scrolling forward and backward. By means of 'VCM' modules, this editor can be made to work with any CRT. It supports many features including:

- . String/word search
- . Step/iterative replacement
- . Status information
- . Pickup/putdown, etc.

An updated user manual is supplied (hardcopy only) as well as the original manual which is on a disk. Also, included on disk are: HELP.SV, VERSN3.SV, PAGE8.SV, FLIST.SV, BATCH.SV, HELP.SV, ACID.SV and DIRECT.SV, CCL.SV. With the exception of ACID and PAGE8, these programs are enhanced versions of the originals.

Media (Service Charge Code): User's Manual (EB), Four RX01 Diskettes (KD) Format: OS/8

September 15, 1986

REVISIONS TO LIBRARY PROGRAMS

DECUS NO: 11-750 **Title:** TEM: A Terminal Emulator for RSX-11 **Version:** V86.093, June 1986

Submitted by: Thomas R. Wyant III, E. I. du Pont de Nemours, Richmond, VA **Operating System:** RSX-11S V4.0, RSX-11M V4.0, VAX-11 RSX V2.0, VAX/VMS V4.1, RSX-11M-PLUS V2.0 **Source Language:** MACRO-11 **Memory Required:** 32KB **Hardware Required:** Dial-out Modem **Keywords:** Data Communications, Emulators, Utilities - RSX-11

Abstract: TEM provides "dumb" terminal emulation over a full duplex TT: line. It allows the user to "become" a terminal on a remote system, and to do ASCII file transfers between systems. TEM has been used to communicate with RSX, VMS, RSTS and TOPS-20 systems, as well as non-DEC equipment. It requires no software on the remote system (and therefore has no error checking).

In addition to the basic functionality, TEM can automatically issue canned commands to smart modems at the beginning and end of a session. The user can also select from the following features:

- . Local Echo
- . Automatic line feed on carriage return
- . Translation of inbound control characters to ASCII abbreviations
- . Passthru of control/s, control/q, control/o and control/x to the remote system.
- . User selectable attention and end-of-file characters
- . Inbound and outbound character mapping
- . Specifiable record delay and prompt character for file transfer.
- . Parity generation and checking
- . Support for dialout modems as remote devices

TEM requires at least RSX-11M-PLUS V2.0, VAX-11 RSX V2.0, RSX-11M V4.0 or RSX-11S V4.0. If running under RSX-11M or RSX-11S, it requires the full-duplex TT: driver, get/set multiple characteristics, and unsolicited input AST's. Correct

access of named directories and files numbered in decimal requires the FEAT\$ directive. The GIN\$ directive is used to prevent nonprivileged users from using TEM to read files that are none of their business (e.g. LB: O.O. RSX11.SYS). An attempt has been made to conditionalize TEM for RSX-11M V3.2, but I have no way to check it. TEM can be initiated from and communicate with any reasonable serial device, but there may be restrictions if not being used on a TT-type device.

Media (Service Charge Code): Two RX01 Diskettes (KB) Format: RT-11, 600' Magnetic Tape (MA) Format: DOS-11

September 15, 1986

DECUS NO: CPM-269 **Title:** CP/M-86 KERMIT **Version:** V2.9, July 1986

Author: Bill Catchings, Columbia University

Operating System: CP/M-86 **Source Language:** ASM86 **Keywords:** KERMIT

Abstract: KERMIT is a public domain communications program available for a wide variety of machines, including the Digital Equipment Corporation Rainbow 100, Professional, PDP-11 (most operating systems) and VAX computers, plus other manufacturers' computers.

Using KERMIT, you can transfer files between two machines with error recovery, log a terminal session to a file, or just to terminal emulation.

Notes: CPM-86 V2.0 and higher is required.

Media (Service Charge Code): One RX50 Diskette (JA) Format: CP/M August 25, 1986

DECUS NO: 11-SP-47 **Title:** PortaCalc (AnalytiCalc): A 3D Spreadsheet/Database System **Version:** V20-05 A, April 1986

Submitted by: Glenn C. Everhart, Ph.D. **Operating System:** P/OS, MS/DOS, VAX/VMS, RSX-11M-PLUS, RSX-11M, RSX-11D, RSTS/E, IAS **Source Language:** MACRO-32, MACRO-11, VAX-11 FORTRAN, FORTRAN 77, FORTRAN IV-PLUS, C **Memory Required:** PDP-11: 64KB - VAX/VMS: N/A **Keywords:** Business Applications, Data Base Management, PortaCalc, Spreadsheet

Abstract: PortaCalc is a powerful three dimensional spreadsheet/database and analysis system with easy user extensibility designed to outperform most any commercial package available, running on PDP-11 systems able to support the F4P compiler, or VAX systems, needing the VAX FORTRAN compiler to compile. Several terminals are supported, including the VT100 series, VT52, Datamedia Colorscan 10 and Elite 1500, Televideo 925, and ANSI color terminals.

The program is designed for power, and to be easily portable to other systems supporting FORTRAN, with peculiarities used documented, and its manual is designed to be turned into a system HELP file so that it can be read online.

A data management system interface is built in, permitting spreadsheets to access a potentially unlimited number of files and records or parts of records in those files for user defined

functions, numbers, formulas, text, or whatnot. In fact, it has many of the attributes of a language.

A powerful text integration function permits integration of word processing files with reports, permitting use of AnalytiCalc (PortaCalc) to integrate sections of reports which are edited with any editor. It also simplifies inserting text from external files flexibly over null cells of the spreadsheet.

The current release adds an "Enter-mostly" mode of operation that provides a command structure similar to commercial spreadsheets on micros (numbers, formulas, and text just get typed in, commands prefixed with /) as an option, plus various additional capabilities.

You have the ability to access HUGE spreadsheet dimensions; up to 32,000 rows and 32,000 columns can be addressed on the VAX, and up to 10,000 by 10,000 on the PDP-11.

This version of AnalytiCalc (PortaCalc) now contains a two page memory paging algorithm for the PDP-11 version. It also has a version presented here for the first time containing a Datatrieve-32 interface permitting exchange of data between DTR databases and AnalytiCalc spreadsheets in both directions.

Complete source code for all versions is provided. It is assumed the F4P and F77 compiler is available for PDP-11, or the Digital Equipment Corporation VAX FORTRAN compiler for VAX. Please Note: Object libraries are provided for VAX systems not owning FORTRAN, and task images for RSX systems without F77.

Release Notes are distributed with each order.

Notes: VAX/VMS users see DECUS No. V-SP-24.

Changes and Improvements: Fixed bugs in matrix math in one version; some minor speedups. Improved "quick-reference card" file.

Media (Service Charge Code): User's Manual (EO), 2400' Magnetic Tape (PC) **Format:** RMSBCK

August 25, 1986

DECUS NO: 11-674 **Title:** FILTRA: A File Transfer Program **Version:** December 1985

Submitted by: Frank Bosso, Presco Inc., Woodbridge, CT **Operating System:** RSX-11M-PLUS V2.1, RSX-11M V4.1 **Source Language:** MACRO-11, FORTRAN IV-PLUS, FORTRAN IV **Memory Required:** 20KW **Keywords:** Data Communications

Abstract: FILTRA is a file transfer program written for a host computer (PDP-11). It enables the host computer to transfer files to and from a micro computer. As FILTRA includes error checking, the micro computer must have a compatible program such as MODEM. Compatible programs for micro computers such as the VT180 and the Rainbow-100 are available commercially, and others are in the public domain.

While FILTRA is designed as a host program, it could also be used as a local file transfer program. In this manner files could be transferred to another PDP-11 or to a VAX. To use FILTRA as a local program, either minor programming changes would have to be made, or hardware (switches) would have to be added to a system.

FILTRA is written in FORTRAN IV-PLUS for a PDP-11 with a RSX-11M operating system. The program makes use of several system subroutines, and it is therefore limited to IAS/RSX-11M type operating systems. Files are stored on the host computer as formatted data files.

No restrictions are made on the type of data to be transferred. It may be either 7 bit ASCII or 8 bit data. The actual transfer uses 8 bit values. Binary (8 bit) files are stored as 128 byte records. ASCII files are reformatted so that each line corresponds to a record.

Media (Service Charge Code): One RX02 Diskette (LA) **Format:** FILES-11, 600' Magnetic Tape (MA) **Format:** FILES-11

August 25, 1986

DECUS NO: 11-795 **Title:** GRAPHKIT: Graphics Routines for the HP-7221C Plotter **Version:** V2, May 1986

Submitted by: R. E. Beverly III Ph.D., R. E. Beverly III and Associates, Columbus, OH **Operating System:** RSX-11M V4.1 **Source Language:** FORTRAN 77 **Memory Required:** Largest program requires 26KW **Software Required:** Hewlett-Packard PLOT/21 software library **Hardware Required:** Hewlett-Packard 7221C Plotter **Keywords:** Graphics, Scientific Applications

Abstract: GRAPHKIT is a collection of software tools designed to supplement Hewlett-Packard's PLOT/21 library by providing routines to easily plot linear, semilogarithmic and logarithmic graphs in standard scientific/engineering formats of publication quality. An additional routine is provided which permits rapid layout and production of viewgraphs and transparencies.

The user is given full control over the x- and y-axis minima and maxima, the generation of axis labels and major and minor tick marks and curve legends. Multiple curves can be drawn on a single plot. Each curve can consist of data symbols only, data symbols connected by straight lines, or lines connecting the data points with no symbols. The user selects the pen number and symbol type (if any) for each curve.

Changes and Improvements: Version 2 supports embedded superscripts and subscripts in the axes titles and graph legends. Additional cosmetic changes and user-adjustable parameters have been implemented.

Media (Service Charge Code): One RX01 Diskette (KA) **Format:** FILES-11, 600' Magnetic Tape (MA) **Format:** FILES-11

August 25, 1986

DECUS NO: 11-812 **Title:** CU - A Program for Converting Units **Version:** April 1986

Author: Ted Dustman, V.A. Hospital, Salt Lake City, UT

Submitted by: Robert Dustman, V.A. Hospital, Salt Lake City, UT **Operating System:** TSX+ V5, RT-11 V4 **Source Language:** DECUS C, C **Memory Required:** 12KW **Software Required:** DECUS C, with Floating Point if modifications are made **Hardware Required:** FPU Floating Point Unit **Keywords:** Conversions

Abstract: CU (Convert Units) converts a value specified in one set of units to a corresponding value in another dimensionally compatible set of units. For instance, a length specified in meters can be converted to feet or a volume specified in gallons can be converted to pints. However, the program is not limited to simple conversions such as these; one can easily perform the following conversions:

Aces	to Feet \wedge 2 (square feet)
Newtons	to Slug Feet/Second \wedge 2
Gallons/Day	to Inches \wedge 3/Year
Watts	to Meters \wedge 2 Slug/Week \wedge 3

The last example is absurd but demonstrates the flexibility of the program. Prefixes (e.g. milli, mega, micro etc.) may be used to scale a unit.

Currently, the program recognizes 97 unit names, including prefix names, but additional names can be added to the program's list of units, (a maximum of 200 units can exist). Note that the program will NOT perform temperature conversions. Also, units that are nonlinear, such as the decibel, cannot be converted using this program.

The program was written in DECUS C and can only be run on those processors that are equipped with FPU floating point hardware (11/23, 11/45, 11/70 etc.). The program will not run on processors with FIS hardware (11/40, 11/03 etc.). This is due to the implementation of floating point by the DECUS C compiler.

Changes and Improvements: New features, bug fixes

Restrictions: Will not perform temperature conversions or conversions on Nonlinear Units.

Media (Service Charge Code): One RX01 Diskette (KA) **Format:** RT-11, 600' Magnetic Tape (MA) **Format:** RT-11

August 25, 1986

DECUS NO: V-SP-24 **Title:** PortaCalc (AnalytiCalc): A 3D Spreadsheet/Database System in VMS/BACKUP **Version:** V20-05A, April 1986

Submitted by: Glenn C. Everhart, Ph.D. **Operating System:** P/OS, MS/DOS, VAX/VMS, RSX-11M-PLUS, RSX-11M, RSX-11D, RSTS/E, IAS **Source Language:** MACRO-32, MACRO-11, VAX-11 FORTRAN, FORTRAN 77, FORTRAN IV-PLUS, C **Memory Required:** PDP-11: 64KB - VAX: N/A **Keywords:** Business Applications, Data Base Management, PortaCalc, Spreadsheet

Abstract: PortaCalc is a powerful 3-dimensional spreadsheet/database and analysis system with easy user extensibility designed to outperform most any commercial package available, running on PDP-11 systems able to support the F4P compiler, or VAX systems, needing the VAX FORTRAN compiler to compile. Several terminals are supported, including the VT100 series, VT52, Datamedia Colorscan 10, and Elite 1500, Televideo 925, and ANSI color terminals.

The program is designed for power and to be easily portable to other systems supporting FORTRAN, with peculiarities used documented, and its manual is designed to be turned into a system HELP file so that it can be read online.

A data management system is built in, permitting spreadsheets to access a potentially unlimited number of files and records or parts of records in those files for user defined functions, numbers, formulas, text or whatnot. In fact, it has many of the attributes of a language.

A powerful text integration function permits integration of word processing files with reports, permitting use of AnalytiCalc (PortaCalc) to integrate sections of reports which are edited with any editor. It also simplifies inserting text from external files flexibly over null cells of the spreadsheet.

The current release adds an "Enter-mostly" mode of operation that provides a command structure similar to commercial spreadsheets on micros (numbers, formulas, and text just get typed in, commands prefixed with /) as an option, plus various additional capabilities.

You have the ability to access HUGE spreadsheet dimensions; up to 32,000 rows and 32,000 columns can be addressed on the VAX, and up to 10,000 by 10,000 on the PDP-11.

This version of AnalytiCalc (PortaCalc) now contains a two page memory paging algorithm for the PDP-11 version. It also has a version presented here for the first time containing a Datatrieve-32 interface permitting exchange of data between DTR databases and AnalytiCalc spreadsheets in both directions.

Complete source code for all versions is provided. It is assumed the F4P or F77 compiler is available for PDP-11, or the Digital Equipment Corporation VAX FORTRAN compiler for VAX. Please note: object libraries are provided for VAX systems not owning FORTRAN, and task images for RSX systems without F77.

Release Notes are distributed with each order.

Notes: PDP-11 users see DECUS No. 11-SP-47.

Changes and Improvements: Fixed bugs in matrix math in one version; some minor speedups. Improved "quick-reference card" file.

Media (Service Charge Code): User's Manual (EO), 2400' Magnetic Tape (PC) **Format:** VMS/BACKUP

August 25, 1986

DECUS NO: VAX-146 **Title:** WATCHDOG **Version:** V4.1, August 1986

Submitted by: George Walrod III **Operating System:** VAX/VMS V3, X-4X **Source Language:** FORTRAN 77, VAX MACRO **Memory Required:** less than 1K **Keywords:** System Management - VMS, Utilities - VMS

Abstract: WATCHDOG is a program which monitors an interactive process for inactivity. A process is logged out after a defined interval. An inactive process is indicated by no change in CPU time and no buffered I/O count within a defined interval. Messages will be sent to the inactive process at a defined interval until the maximum inactive time limit is reached. A final message is sent to the user and an optional message is sent to the central operator making note that a user has been stopped. Another option includes ignoring a group of users. Many options exist and are documented. You should enjoy the comments made by the developer.

Changes and Improvements: Version 4.x enhancements. supports DBMS-32 and ALL-IN-1.

Media (Service Charge Code): 600' Magnetic Tape (MA)
Format: VMS/BACKUP, or order VAX-LIB-4

September 15, 1986

DECUS PROGRAM LIBRARY CHANGES

DECUS NO: PRO-146, **Title:** PRO/Smart Mailer Binary
Version as listed in the catalog no longer available

DECUS NO: RB-105, **Title:** SEDT: EDT/WPS Screen Editor
for MS/DOS as listed in the catalog is no longer available

DECUS NO: 10-364, **Title:** CRYPT, PSWCHK, PODTYP,
MONRPT/RESP, **Version:** V2(2), December 1984 will be included in DECUS NO: 10-LIB-12.

DECUS NO: VAX-LIB-1, **Title:** The VAX Library Tape 1,
Version: 1986/1987, reads "Restrictions: SPICE2, (VAX-6)".
This should be "Restrictions: SPICE3, (VAX-6)".

DECUS NO: VAX-6, **Title:** SPICE3 A6, **Version:** 3 A6, February 1986, please add the following note in the "Changes and Improvements" section that a Digital Equipment Corporation C Compiler is needed.

HOW TO SUBMIT TO A SPECIFIC SECTION OF THE NEWSLETTER

The following is a listing of the Newsletter Editors with their addresses and phone numbers. All submissions to the newsletter should be submitted directly to the appropriate Editor.

ARTIFICIAL INTELLIGENCE

Terry Shannon
160 State Street
Boston, MA 02109
(617)367-7190

BUSINESS APPLICATION

Thomas Byrne
L. Karp & Sons
1301 Estes
Elk Grove, IL 60007
(312)593-5705

COMMERCIAL LANGUAGES

Ted Bear
RAMTEK
2211 Lawson Lane
Santa Clara, CA 95950
(408)988-2211

DAARC

Ellen Reilly
William H. Rorer
500 Virginia Drive
Ft. Washington, PA 19034
(215)628-6547

DATA MANAGEMENT SYSTEMS

Russ Poisson
Seed Software Corp.
2121 Eisenhower Avenue
Alexandria, VA 22314
(703)783-4944

DATATRIEVE/4GL

Donald E. Stern, Jr. c/o
Warner Lambert Company
10 Webster Road
Milford, CT 06460
(203)783-0238

EDUSIG

Fred Bell
Taft College
29 Emmons Park Drive
P.O. Box 1437
Taft, CA 93268
(805)763-4282

GRAPHICS APPLICATION

Michael Anton
P.O. Box 591293
Houston, TX 77259-1293
(713)928-4838

HMS

William Walker
Monsanto Research Corp.
P.O. Box 32 A-152
Miamisburg, OH 45342
(513)865-3557

IAS

Frank Borger
Physics Division
Michael Reese Hospital
Lake Shore Drive at 31st St.
Chicago, IL 60616
(312)791-2515

LANGUAGES & TOOLS

Alan Folsom Jr.
Fischer & Porter Company
E. County Line Road
Warminster, PA 18974
(215)674-7154

LARGE SYSTEMS

Clyde T. Poole
The University of Texas at Austin
Department of Computer Science
Taylor Hall 2.124
Austin, TX 78712-1188
(512)471-9551

MUMPS

Janet Berryman
2405 N. Bush
Santa Ana, CA 92706
(714)953-1025

NETWORKS

Vicki Hancock
2510 Limestone Lane
Garland, TX 75040
(214)495-7353

OFFICE AUTOMATION

Therese LeBlanc
275 London
Wheeling IL 60090
(312)459-1784

PERSONAL COMPUTER

Kenneth LeFebvre
Sytek, Inc.
19 Church Street
P.O. Box 128
Berea, OH 44017-0128

RSTS

Charles Mustain
Stark County Local School System
Dept. of Education Service Ctr.
7800 Columbus Road NE
Louisville, OH 44641
(216)875-1431 x279

RSX

Bruce Mitchell
Machine Intelligence & Industry Magic
P.O. Box 816
Byron, MN 55920
(507)775-6268

RT

Bill Leroy
The Software House, Inc.
2964 Peachtre RDNW #320
P.O. Box 52661
Atlanta, GA 30355
(404)231-1484

SITE MANAGEMENT & TRAINING

Gregory Brooks
Washington University
Behavior Research Lab.
1420 Gratton St.
St. Louis, MO 63104
(314)241-7600 x257

UNISIG

James Livingston
Measurex Corp.
1 Results Way
Cupertino, CA 95014
(408)255-1500 x4468

VAX SYSTEMS

Larry Kilgallen
c/o DECUS Office
219 Boston Post Road (BP02)
Marlboro, MA 01752-1850

SUBMITTING ARTICLES TO THE HMS SIG NEWSLETTER

The purpose of the HMS SIG Newsletter is to serve as a forum to share information related to DEC hardware with the members of the SIG. As such, the existence of the newsletter is entirely dependent on your contributions. If you have an HHK item, a better or safer way to do something, product news, a tutorial article of general interest, etc., we are interested in publishing it in the newsletter. It is intended that the HMS SIG Newsletter be published at least four times a year.

There are several ways to submit material for the newsletter:

- o The Hardware Submission Form in the back of the newsletter can be used for brief items (there is not enough room if you have a lot to say).
- o You can send me camera-ready hard-copy (this saves me a lot of typing).
- o I will accept submissions on floppys. I can handle RX50's or 8" diskettes (either density, single or double sided). I prefer RT-11 format, if possible, but I can probably handle RSX or VMS stuff somehow. I will return your diskette(s), of course.
- o Those of you that have access to DCS can send things to username WALKER. I check DCS daily.
- o I am also on CompuServe as "Bill Walker 71066,24".

In any event, if you have anything to submit, send it! If it is a mess, but I can read it, I will get it in the newsletter somehow. Finally, if you have any question about submitting material, call me. My telephone number is listed below.

Contributions can be sent to:

HMS Editor		William K. Walker
DECUS	OR	Monsanto Research Corp.
BP02	==	P.O. Box 32 A-152
249 Northboro Road		Miamisburg, OH 45342
Marlboro, MA 01752		(513) 865-3557

If you need to get something to me quickly, send it to my work address.



DECUS SUBSCRIPTION SERVICE
SIGs NEWSLETTERS
U.S. CHAPTER MEMBERS ONLY

As a member of DECUS U.S. Chapter, you are entitled to contribute and subscribe to the DECUS monthly publication, **SIGs Newsletters**. You also have the opportunity to subscribe to the Symposia Proceedings which are a compilation of the reports from various speakers at the U.S. National DECUS Symposia.

- **No Purchase Orders will be accepted.**
- The order form below must be used as an invoice.
- All checks must be made payable to DECUS.
- All orders **MUST** be paid in full.
- No refunds will be made.
- The address provided below will be used for all DECUS mailings; i.e. Membership, Subscription Service and Symposia.
- SIGs Newsletters Price is for a one-year subscription beginning the month following receipt of payment.

Name _____ DECUS Member No. _____
 Company _____
 Address _____

City _____ State _____ Zip _____
 Phone _____

Subscription Service Offering	Qty.	Unit Price	Total
SIGs Newsletters	_____	\$35.00	_____
Fall '85 Proceedings (FA5)	_____	15.00	_____
Spring '86 Proceedings (SP6)	_____	15.00	_____
Fall '86 Proceedings (FA6)	_____	15.00	_____
Spring '87 Proceedings (SP7)	_____	15.00	_____
TOTAL COST OF SUBSCRIPTION			\$ _____

MASTERCARD VISA DINERS CLUB/CARTE BLANCHE®

_____ Exp. Date _____

I understand that there will be no refunds even if I decide to cancel my subscription.

Signature: _____

FOR DIGITAL EMPLOYEES ONLY

Badge No. _____ CC: _____
 CC Mgr. Name _____
 CC Mgr. Signature _____

FOR DECUS OFFICE ONLY

Check No. _____
 Bank No. _____
 Amount \$ _____

Subscription Service, DECUS(BP02), 219 Boston Post Road, Marlboro, MA 01752-1850, (617) 480-3418.

TYPE OF BUSINESS (ENVIRONMENT)/COMPUTER APPLICATIONS

Please check that which best describes your business/application

- 21 ACCOUNTANCY
- 7 BANK
- 64 BUSINESS/COMMERCIAL
- 74 BUSINESS/INFORMATION SYSTEMS
- 57 CHEMISTRY
- 54 CLINICAL LABORATORY
- 63 COMPUTATION
- 11 CONSUMER ELECTRONICS
- 18 CONSULTANT
- 72 DATA ACQUISITION
- 52 DATA COMMUNICATIONS
- 13 DATA PROCESSING SERVICES
- 71 DATA REDUCTION
- 17 DIGITAL EMPLOYEE-ENGINEERING
- 15 DIGITAL EMPLOYEE-MARKETING
- 16 DIGITAL EMPLOYEE-SERVICE GROUP
- 60 EDUCATIONAL ADMINISTRATION
- 1 EDUCATION/PRIMARY
- 2 EDUCATION/SECONDARY
- 61 EDUCATION-TECHNOLOGY
- 3 EDUCATION/UNIVERSITY
- 67 ENGINEERING
- 65 FINANCE/ACCOUNTING
- 77 GOVERNMENT
- 75 GRAPHICS
- 4 HOSPITAL
- 62 INDUSTRIAL
- 55 LABORATORY/SCIENTIFIC
- 14 LIBRARY
- 58 LIFE SCIENCES
- 70 MANUFACTURING
- 79 MARKETING
- 59 MEDICAL RESEARCH
- 6 MILITARY INSTALLATION
- 73 NUMERICAL CONTROL
- 68 OEM-COMMERCIAL
- 78 OEM-TECHNICAL
- 56 PHYSICAL SCIENCES
- 20 RESEARCH/DEVELOPMENT
- 10 RETAIL
- 76 SOFTWARE DEVELOPMENT
- 53 TELECOMMUNICATIONS
- 19 TELEPHONE/UTILITIES
- 51 TIMESHARING
- 80 TRAINING/INSTRUCTION
- 66 TYPESETTING/PUBLICATION

SPECIAL INTEREST GROUP (SIGs) ENROLLMENT

I wish to participate in the following DECUS U.S. Chapter Special Interest Groups.

- 33 APL SIG
- 2 COMMERCIAL LANGUAGES
- 6 DATA MGMT.SYS.
- 5 DATATRIEVE
- 7 BUSINESS APPL
- 8 EDUSIG
- 10 GRAPHICS APPL
- 11 HARDWARE AND MICRO
- 35 IAS
- 31 DAARC(LABS)
- 27 LARGE SYSTEMS
- 16 LANG. AND TOOLS
- 14 MUMPS
- 15 NETWORKS
- 34 OFFICE AUTOMATION
- 36 PERSONAL COMPUTER
- 18 RSTS/E
- 17 RSX
- 19 RT-11
- 32 SITE MGMT.& TRNG
- 21 UNISIG
- 26 VAX SYSTEMS

JOB TITLE/POSITION - Please check:

- 1 CORPORATE STAFF
- 2 DIVISION OR DEPARTMENT STAFF
- 3 SYSTEMS ANALYSIS
- 4 APPLICATIONS PROGRAMMING
- 5 SYSTEMS ANALYSIS/PROGRAMMING
- 6 OPERATING SYSTEM PROGRAMMING
- 7 DATABASE ADMINISTRATION
- 8 DATA COMMUNICATIONS/TELECOMMUNICATIONS
- 9 COMPUTER OPERATIONS
- 10 PRODUCTION CONTROL
- 101 CORPORATE DIRECTOR OF DP/MIS
- 102 ADMINISTRATIVE ASSISTANT
- 103 TECHNICAL ASSISTANT
- 104 SERVICES COORDINATOR
- 105 MANAGER
- 106 ANALYST
- 107 PROGRAMMER
- 108 DATABASE MANAGER
- 109 DATABASE ADMINISTRATOR
- 110 MANAGER OF DP OPERATIONS

CITIZEN OF UNITED STATES OF AMERICA? Yes No Country: _____

Signature: _____ Date: _____

Forward To:

DECUS U.S. CHAPTER, MEMBERSHIP PROCESSING GROUP
219 BOSTON POST ROAD
MARLBORO, MA 01752, USA
PHONE: (617) 480-3418

Ask the WOMBAT WIZARD
Submission Form

To submit a problem to the WIZARD, please fill out the form below
and send it to:

WW Editor, Philip A. Naecker
Consulting Software Engineer
3011 North Mount Curve Avenue
Altadena, CA 91001
USA

Name: _____ DECUS Membership No. _____

Affiliation: _____

Address: _____

Telephone Number: _____

Statement of Problem: _____

Please following the following guidelines when submitting support
material:

1. If you are trying to demonstrate a method or a concept,
please simplify the procedures, records, and other information
to the shortest form possible.
2. Annotate your attachments. Simple comments or hand-written
notes ("Everything worked until I added this statement.") go a
long way toward identifying the problem.
3. Keep an exact copy of what you send. And number the pages
on both copies. But send everything that is related to your
question, even remotely.
4. If you would like a direct response or would like your
materials returned, please don't forget to include a stamped,
self-addressed envelope large enough to hold the materials you
send.

DATATRIEVE/4GL SIG
Product Improvement Request Submission Form

Submittor:
Address:

DECUS Membership Number:
Firm:

Phone:

Product or Products:

How to write a PIR

A PIR should be directed at a specific product or group of products. Be sure to give the full name of the product(s) and version numbers if applicable. Describe the functionality you would like to see in as complete terms as possible. Don't assume that the PIR editors or software developers know how it is done in some other software product - state specifically how you want the software to function. Provide justification of your request and give an example of its use. If you can, suggest a possible implementation of your request.

Abstract: (Please limit to one or two short sentences.)

Description and Examples: (Use additional pages as necessary.)

[Put my name and address on reverse side, thus:]

PIR Editor, Philip A. Naecker
Consulting Software Engineer
3011 North Mount Curve Avenue
Altadena, CA 91001
USA

IAS WHIMS

WHAT: (Describe your WHIM) (Please print or type)

WHY: (Describe the reason for the WHIM)

HOW: (Make any suggestions for a possible implementation)

Name: _____

Company: _____

Address: _____

Phone: _____

Please mail to:

Kathleen M. Anderson
EATON Information Management
Systems Division
2017 Cunningham Drive
Suite 208
Hampton, Virginia 23666

Phone: (804) 326-1941

DATAGRAM

DATAGRAMs are short messages, comments, requests, or answers that are published in NETwords. Please fill in the sections below and send the DATAGRAM to:

Vickie Hancock
NETWords Editor
2510 Limestone Ln.
Garland, Tx. 75040

Title: _____

Message: _____

Your Name: _____

Address: _____

Telephone: _____

If this is a reply to a previous DATAGRAM, what #? _____

Signature: _____ Date: _____

**Place
Stamp
Here**

**Vickie Hancock
NETWords Editor
2510 Limestone Ln.
Garland, Tx. 75040**

Fold Here

INPUT/OUTPUT Submission Form

A SIG Information Interchange

Please reprint in the next issue of the Pageswapper

If this is a reply to a previous I/O, which number? _____

Caption: _____

Message: _____

Contact:

Name _____

Address _____

Telephone _____

Signature _____ Date _____

Mail this form to: Larry Kilgallen, PAGESWAPPER Editor
Box 81, MIT Station, Cambridge, MA 02139-0901, USA

For information about on-line submission, dial (in the United States): (617) 262-6830 and log in with the username PAGESWAPPER.

PAGESWAPPER - November 1986 > Volume 8 Number 4
INPUT/OUTPUT Submission Form

Tear out or photocopy reverse to submit an I/O item

Larry Kilgallen, PAGESWAPPER Editor
Box 81, MIT Station
Cambridge, MA 02139-0901
USA

System Improvement Request Submission Form

Page 1 of _____

Submittor:

Firm:

Address:

Phone:

How to write an SIR:

Describe the capability you would like to see available on VAX systems. Be as specific as possible. Please don't assume we know how it's done on the XYZ system. Justify why the capability would be useful and give an example of its use. If you wish, suggest a possible implementation of your request.

Abstract (Please limit to four lines):

Description and examples (use additional pages if required)

PAGESWAPPER - November 1986 - Volume 8 Number 4
System Improvement Request Submission Form

Tear out or photocopy reverse to submit an SIR

Mark D. Oakley
Battelle Columbus Division
Room 11-6-008
505 King Avenue
Columbus, Ohio 43201-2369
USA

Non-DEC VMS Product Registration Form

Note: If a category is not applicable, please write in "N/A".
Each description can only be up to 70 characters.

Product name

.....

A. Facility name or software acronym

.....

Short description

.....

VMS Error Message status code, Yes or No?

.....

B. Logical name prefix(es)

.....

C. System wide process name(s)

.....

D. System wide mailbox name(s)

.....

E. Shareable images

.....

F. Company name

.....

Contact name

.....

Address

.....

.....

Phone number

.....

Mail to: Digital Equipment Corporation
110 Spitbrook Road
Nashua, NH 03062-2698

VAX Tape Drive Questionnaire

VAX SIG Hardware Working Group
Tape Drive Questionnaire

Over the past few years a number of Symposium attendees have expressed concern over DEC's tape drives. During the past few months the working group has put together a questionnaire addressing HIGH PERFORMANCE tape drives, to determine just how important the issue of good quality high performance tape transports drives is to the SIG membership.

Rather than limiting the input on this critical area to those who attend a specific symposia we are publishing it in the PAGESWAPPER to obtain the widest possible response. We appreciate your efforts to fill this out and return it to:

Dave Schmidt
Hardware Working Group Chair
Management Science Associates
5100 Centre Avenue
Pittsburgh, Pa. 15232

The deadline for responding to the questionnaire will be three weeks after the PAGESWAPPER is delivered, based on my receipt of the issue. Once the responses have been received the material will be used to present a position paper to DEC and publish it in the PAGESWAPPER. Your cooperation is greatly appreciated.

Please define the size of your computing facility.

1. Name and address. (OPTIONAL)

2. Installed VAX models (please indicate the number of each type):

___ MicroVAX I/II ___ VAX-11/7XX ___ VAX 8XXX

3. Are any of your CPU's in a VAXcluster? ___ Yes ___ NO

4. Does your site utilize an IBM system? ___ Yes ___ NO

5. Does your site use tapes from an IBM site? ___ Yes
___ NO

6. Total disk storage for all of your CPU's?

7. How many tape drives are installed at your site?

8. Are you using reel-to-reel tape drives with your VAX systems?

_____ YES _____ No

If yes, please list all makes and models: _____

9. How many tapes are in your tape library? _____

10. On average, how many tape mounts per normal working day are performed at your site? _____

What are your current tape applications? Do you expect them to change over the next 5 to 10 years? How?

PAGESWAPPER - November 1986 - Volume 8 Number 4
 VAX Tape Drive Questionnaire

Are you aware of or do you use IBM 3480 cartridge tape drives?

_____ Aware of _____ Use _____ Neither aware of nor use.

If you are aware of or use IBM 3480 cartridge tape drives, and currently use reel-to-reel tape drives. Do you think a 3480-type of DEC cartridge drive would be better than, about the same as or worse than your current reel to reel drives for each of the items below?

	Better than Reel-to-reel	About the same as reel-to-reel	Worse Than reel-to-reel
Performance	_____	_____	_____
Cost of ownership	_____	_____	_____
Data Integrity	_____	_____	_____
Reliability	_____	_____	_____
Meets Current Application Needs	_____	_____	_____
Meets Future Application Needs	_____	_____	_____
Ease of USE	_____	_____	_____
Other (please list)			

Two years from now if you could have either a 3480-type DEC drive or your current reel-to-reel tape drive (BUT NOT BOTH), which would you choose?

3480-type DEC drive Reel-to-reel Can't Decide

Several years ago DEC offered an IBM plug compatible tape controller. If such a product was available again would your organization purchase it?

Would purchase Would be interested No interest

If a product was available to allow DEC and IBM systems to share tape drive(s), How much would your organization benefit from it?

Substantial benefit Some benefit No benefit

If your organization would benefit from such a product, who would be your vendor of choice? _____

Thank you for completing this survey.



DECUS SUBSCRIPTION SERVICE
DIGITAL EQUIPMENT COMPUTER SOCIETY
219 BOSTON POST ROAD, (BP02)
MARLBORO, MA 01752-1850

Bulk Rate
U.S. Postage
PAID
Permit No. 18
Leominster, MA
01453

STATUS CHANGE

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

Change of Address
 Please Delete My Membership Record
(I Do Not Wish To Remain A Member)

DECUS Membership No: _____

Name: _____

Company: _____

Address: _____

State/Country: _____

Zip/Postal Code: _____

Mail to: DECUS - Attn: Subscription Service
219 Boston Post Road, BP02
Marlboro, Massachusetts 01752-1850
USA

_____ label
_____ is not
_____ int old
Include
_____ allation,
_____ iversity,